Warum AFW

Kein Wettbewerbsvorsprung durch SAP Standard

Mittlerweile hat auch das letzte Unternehmen SAP erreicht. Jedem steht somit die Software mit V8 und 500 PS zur Verfügung. Gute Berater können nun auf dem einen oder anderen Teilstück einer Kurve auf der Nordschleife etwas mehr Abtrieb erzeugen, um ein wenig von der angebotenen Power auf die Straße zu bringen.

Da den Mitbewerbern die gleichen Mittel und Möglichkeiten zur Verfügung stehen, kann hier kein Wettbewerbsvorsprung entstehen. „Hm, ja, … alles schön: XK*, FK*, MK*, aber wir bräuchten dringender noch das und das und das und das …".

Niemand nimmt diesen Augenblick wirklich als „entscheidend" wahr. Erst jetzt wird am Wettbewerbsvorsprung gearbeitet. Die individuellen Besonderheiten der Marktführerschaft werden durch individuelle Softwareunterstützung erst richtig effizient. Eine der wichtigsten Ressourcen eines Unternehmens wird mal „so nebenbei" erstellt, verwaltet, gepflegt. Dieses Know-how finden Sie in keiner APP im App-Store. Dieses entscheidende Firmen-Know-how kann und muss besser, effektiver, effizienter genutzt, erstellt, gepflegt und verwaltet werden. Die Zusatzentwicklung im SAP ist nicht nur ein wesentlicher Erfolgsfaktor, sie ist auch ein Kostenfaktor, dessen wir uns ab 18 € pro Monat für Sie annehmen wollen.

Die beste Lösung zum Erstellen eigener SAPGui Anwendungen

Die Vorteile eines Programmrahmens für die SAP ABAP Eigenentwicklung, Application FrameWork (AFW), sind bekannt. Wir haben die Vorteile zum Nachlesen hier, zusammengefasst. Unser AFW löst nicht Ihre Business-Probleme, es bietet vielmehr die strukturierte Basis für deren Lösung. Auf dem AFW konnten wir ohne großen Aufwand z.B. unseren BUM NotfallUser entwickeln.
Die Eigenentwicklung im SAP wurde bisher nicht als wesentlicher Erfolgsfaktor erkannt, was sie definitiv aber ist. Da Eigenentwicklung im SAP alle Bereiche und alle Phasen fast aller SAP Projekte betrifft, wäre ein einziger Artikel zu diesem Thema zu umfangreich. Im ersten Teil geht es einfach darum, nichts Dummes zu tun.

Teil 1: Fehlervermeidungsstrategie

Was müsste in einer Fehlervermeidungsstrategie berücksichtigt werden?

Es beginnt mit der Erkenntnis: Fehler sowie Fehleinschätzungen passieren jedem auf allen Unternehmensebenen. Damit Fehler erkannt und zukünftig vermieden werden können, müssen alle Beteiligten ohne Furcht vor Sanktionen Unsicherheiten offenlegen dürfen, um in den Fehlervermeidungsprozess eintreten zu können.

Fehler bei der Zieldefinition - wie nicht bekannte aufgelöste Zielkonflikte - führen schnell am Ziel vorbei. Dazu mehr in den folgenden Teilen dieser Serie.

Gutes Zeitmanagement ist ein bedeutender Faktor um permanenten Druck zu vermeiden. Softwareentwicklungen haben eine gewisse Eigendynamik:

st services

Erfahrung ist durch nichts zu ersetzen

Jeder Entwicklungsprozess braucht Sorgfalt und Übersicht, damit die Veränderungen zielgenau funktionieren und in ihrer ganzen Tragweite erfasst und getestet werden können. Gleichzeitig bleiben die finanziellen Ressourcen im Fokus: Mehr Entwicklung braucht mehr Zeit, kostet also zusätzlich Geld. In der Folge entstehen Kompromisse, die nicht selten den Erfolg des Projektes gefährden. Die Kompetenz der Entscheider - in der Praxis die, der Geldgeber - ist an diesem Punkt von immenser Bedeutung. Ohne Erfahrung, ohne Erfahrungswerte und ohne ein erfahrenes Team, muss die mangelnde Erfahrung durch Zeit und/ oder Geld erkauft werden.

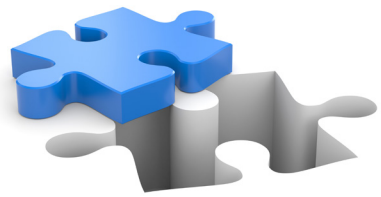
Bei der Entwicklung unseres AFW sind die wesentlichen Parameter der Entwicklung unmittelbar an den Erfordernissen der Praxis orientiert. Unser BUM (BenutzerUser Management) wurde und wird im „laufenden Betrieb“ durch unser AFW, plus Businesslogik, ersetzt. Die unterschiedlichsten Kundenanforderungen können nun pro Kunde gekapselt werden.

Die Verringerung des reinen Source-Codes reduziert die mögliche Fehleranfälligkeit. Die Pflege wird vereinfacht und beschleunigt die Prozesse.
Unsere BUM Erweiterung, der „NotfallUser“, hat sich fast von selbst integriert. Dabei ist der Programmierstil einfach und verständlich. Lesbare und gut nachvollziehbare Sourcen mit klar gefassten Vorgaben erleichtern die zukünftigen Erweiterungen/ Änderungen.

Sind die Entscheidungen an die Kompetenz-Inhaber delegiert, können sinnvolle praxisorientierte Maßnahmen beschlossen werden.

Fragestellungen innerhalb der AFW Entwicklung:

Mit oder ohne Java?
Reines OO oder gemischt oder ABAP normal?
Mit oder ohne Business (FI, CO, MM, SD, usw.)?
Maximale oder minimale Datenmanipulation?
Maximale oder minimale Basisfunktionalität?
Volle Dynpro - Unterstützung oder selektives ALV Grid?
Unsere Entscheidungen sind blau hervorgehoben.



Die Programmierung der Grundlagen (Basisfunktionalität, Datenmanipulation) entsteht Klasse um Klasse und hinterlässt in der Oberfläche keine direkt sichtbaren Spuren.

Es heißt „OO reisst die Performance herunter, bis dann gar nichts mehr geht“. Unser Vorteil war, dass, wenn eine normale Funktion durch eine OO Funktionalität ersetzt wurde, wir sofort ein Feedback zur Performance in unserer Entwicklungs- und Testumgebung hatten. War die OO Funktionalität stabil und performant, war sie es auch bei unseren Kunden. Dazu mehr in den folgenden Teilen dieser Serie.

Eine auf Risikominimierung ausgerichtete Entwicklung verläuft anders als eine, bei der ein Endtermin, ein Budget steht, das nicht überschritten werden darf, Funktionalität, die zu liefern ist, bei einer, wie auch immer definierten Performance.

Das Application Frame-Work basiert auf einer klaren Struktur: Reines SAP Business (Eigenentwicklung, FI, CO, SD, MM, usw.) bleibt außerhalb des AFW. Für die Implementierung der Businessdaten und -tabellen sowie für die reine Business Logik sind „User Exits“ definiert.

Wo befindet sich nun die Grenze?

Es werden Daten verändert, hinzugefügt, gelöscht, überschrieben, kopiert, dupliziert, mit und ohne Vorlagen; dazu werden Änderungsbelege geschrieben und das Sperren an- und ausgeschaltet. Alle notwendigen RFC-Verbindungen und soweit wie möglich „Drag & Drop“ und „Undo“, sind verfügbar. Das AFW überprüft auch die Datenkonsistenz, Key/Foreign Key, Feldlängen/-typen. All das ist AFW. Wird die Businesslogik nicht mehr durch die reine Datenhaltungslogik und -manipulation abgedeckt, kommt Business-Spezialwissen mit individuellen Algorithmen zum Tragen, welches die Anwendung komplettiert.

Zur Zeit finden wir für uns heraus, wie unser AFW mit dem großen vorhandenen BUM Modul korrespondiert. Innerhalb der großen „BUM Logik-Blöcke“ von Rollengenerator, Useradministration, Antragswesen und Audit können wir die integrationsfreien „Stand-Alone“ Logiken als erste ersetzen, sowie die ganz spezielle Kundenlogik für ganz besondere Fälle, z.B. Ticketsystem.

Danach geht die Umstellung der BUM Modul-Logik nur „vertikal“. Sie beinhaltet 3 Stufen:

1. Erweiterte und angepasste Logik in der Datenhaltung/-manipulation, Basis, des AFW.

2. Es wird zwischen der reine BUM Modul Logik und

3. der BUM Modul übergreifenden Logik unterschieden.

Bei der weiteren Umsetzung wird nach unserer Erfahrung immer weniger Arbeit zu 1. und 3. anfallen, und zu 2. wird die individuelle BUM Modul Logik überschaubarer.

Was bedeutet dies für Kunden unseres AFW?

Dass Sie bei 1., dem AFW nichts tun müssen - das machen wir.
Dass Sie bei 2., die individuelle Businesslogik implementieren müssen.
Dass Sie bei 3., der übergreifenden Logik für Ihr Unternehmen, immer weniger entwickeln müssen, da Sie sich nach und nach eine Vorlage dieser Logik zulegen.

Fazit: OO funktioniert hervorragend! Richtig und sinnvoll angewandt erhält man die Vorteile, die man von einer OO Entwicklung erwartet.

Kosten: Unser AFW stellen wir ab € 18 pro Monat zur Verfügung.

Der 2. Teil wird sich im Detail damit befassen, welche Funktionsblöcke für welche Aufgaben zuständig sind und wie die Performance und die Antwortzeiten hoch gehalten werden.