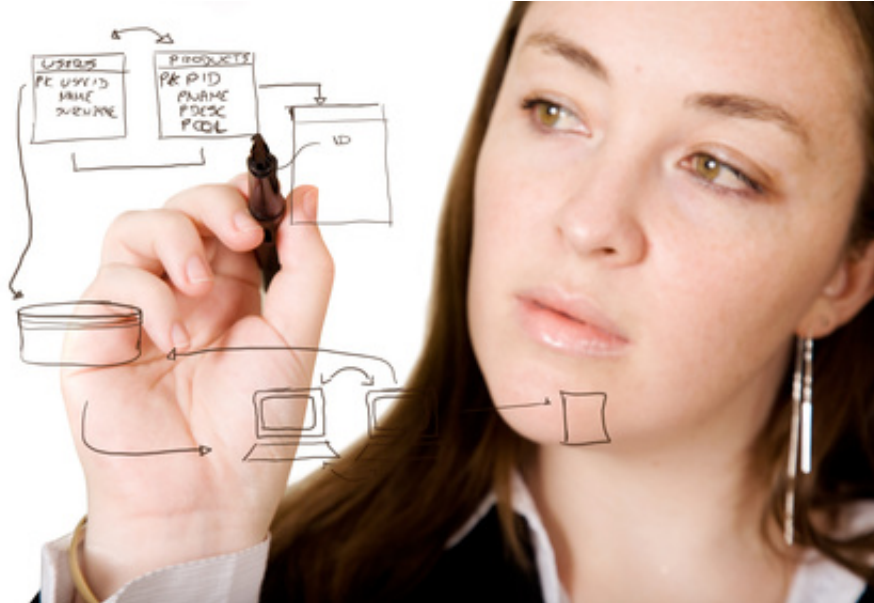


SAP Applikationsentwicklung

Ein beginnendes SAP-Projekt erkennt man daran, dass die Zahl der Schlipsträger in dunklen Anzügen von einem Tag auf den anderen explodiert — diese böse anmutende Beschreibung ist nicht von mir, aber so treffend, dass ich sie hier zu zitieren wage. Und ein kleines bisschen Selbstironie sollte man schon aufbringen können....

SAP-Projekte sind eines der besten Beispiele für verzahnte Arbeit vieler Beteiligten mit vielen möglichen Fehlerquellen. Berater ohne Wissen um die technischen Hintergründe eines SAP-Systems oder Kenntnisse in benachbarten Modulen sind ebenso fehl am Platze wie Entwickler ohne Geschäftsprozess-Know-How. Kein Projektmitarbeiter kann „sein Ding“ allein machen ohne sich exakt mit den Kolleginnen und Kollegen



abzustimmen.

In einem mir bekannten Beratungsunternehmen gilt die Regel: Unsere Berater können alle programmieren — darum brauchen wir auch keine Programmierer. Dass ein solcher Anspruch nicht nur sehr ambitioniert ist, sondern gar als praxisfremd bezeichnet werden muss, zeigte sich nicht zuletzt daran, dass eine ganze Batterie an Entwicklern vorgehalten und in Projekten eingesetzt wird. Zweifellos ist es sinnvoll, wenn ein Berater auch programmieren kann (ebenso sinnvoll wie ein Entwickler, der sich im Customizing auskennt), aber ersetzen kann der eine den anderen nicht. In SAP muss man sich schlicht und ergreifend aufgrund der Komplexität der Themen spezialisieren, will man sein Tätigkeitsfeld wirklich tief beherrschen.

ABAP-Entwicklung

Die ABAP-Entwicklung ist so alt wie SAP R/3 selbst; und das nicht nur, weil SAP in weiten Teilen in ABAP entwickelt wurde, sondern weil schon kleinere Anforderungen z. B. an das Reporting in ABAP umgesetzt werden müssen.

Entwickler in Module einzuteilen, ist der erste Fehler, der oftmals gemacht wird. Abgesehen von einigen Spezialfällen (wie das Modul HR oder IS-U) ist es dem Entwickler meistens egal, in welchem Modul er arbeitet. Und da viele Entwicklungen von Haus aus modulübergreifend stattfinden, müssen einige Entwickler nachdenken, wollen sie die Frage beantworten „für welches Modul ist das nun?“.

```
LL METHOD cl_gui_cfw=>
  l_row_no-row_id GT 0.
  READ TABLE gt_bseg_ove
    ASSIGNING <bseg_ov
p_req_buzei = <bseg_ov
DIF.
DRM.
```

Zwischen ABAP-Entwicklern gibt es teilweise große Unterschiede: Viele Programmierer mit sehr langer Berufserfahrung kommen noch aus der COBOL-Ecke, andere eher aus dem Java-Bereich und haben intensive Erfahrung mit objektorientierter Entwicklung, mit Design Patterns, Methoden und Klassen gemacht.

Aber auch Erstere tun gut daran, wenigstens ansatzweise mit Methoden und Klassen arbeiten zu können, sonst könnte die Entwicklung sie überholen. Leider muss man feststellen, dass OO als eine Art „Modewelle“ für Dies und Das herhalten muss, ohne dass gefragt wird, ob für den geplanten Einsatzzweck einer Applikation überhaupt sinnvoll ist, auf Objektorientierung zu setzen. Man darf nicht vergessen, dass der Code auch gewartet werden muss; und zwar im Zweifel auch von Entwicklern, die noch nicht sehr viel Kontakt zur Objektorientierten Programmierung hatten.

SAPscript und SmartForms / SAP-Forms

Ein oft unterschätztes Thema ist das der Formularentwicklung, insbesondere weil Kunden (und leider oftmals auch Beratern) oft kein Bewusstsein um die Komplexität dieses Projektabschnittes haben. Trotz aller Bemühungen, die Entwicklung zu vereinfachen ist ein Formular in SAP eben doch viel mehr als ein Serienbrief in einer Textverarbeitung. Es besteht aus händisch erstelltem Coding, das weit über die sichtbaren Inhalte (z. B. einer Bestellung) hinausgeht und in dem viele Sonderfälle müssen bedacht werden. Unglücklicherweise ist die SAPscript-Entwicklungsumgebung nicht gerade ein herausragendes Beispiel für Arbeitsergonomie.

```
...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
/* .....
/* Kopfdaten
/* .....
/E HEADER_TEXT
AS
IN &TTXIT-TDTEXT&
/: INCLUDE &T166K-TXNAM& OBJECT &T166K-TDOBJECT& ID &T166K-TDID& LANGUAGE >
/E HEADER_INFO_ACKNOW
IN Für die folgenden Positionen erwarten wir Ihre Auftragsbestätigung
/E TERMS
CN &'Lieferung an 'PEKKO-ABLAD&
CN &'Lieferung für 'PEKKO-WEMPF&
CN &'Lieferbed.: 'EKKO-INCO1& &EKKO-INCO2&
CN &'Zahlungsbed.: 'PEKKO-ZBTXT&,,&'Währung 'EKKO-WAERS&
/E TERMS_OF_PAYMENT
IN &RM06P-PHTXT&
/E HEADER_TARGET_VALUE
IN &'Zielwert 'EKKO-KTWRT& &EKKO-WAERS&
/E REASON_FOR_REJECTION
IN &'Absagegrund 'EKKO-ABSGR& &T165M-ABSGR_TXT&
/E CHANGE_REMARKS
IN *** &T166T-CHTXT& ***
/* .....
/* Positionsdaten
/* .....
/* -- Bestellung .....
/E ITEM_HEADER_F
```

Wie viele Diskussionen haben wir mit Kunden oder Beratern geführt, die ein paar Striche in ein Formular machten und sich wunderten, dass meine Aufwandsschätzung in Regionen führten, an die beim „Striche machen“ nicht im Traum gedacht wurden. Ein gemeinsamer Blick das Formular MEDRUCK führte dann oft zu einem Aha-Erlebnis und dem überfälligen Bewusstsein darüber, dass man eben nicht mit der Maus ein paar Linien verschiebt, sondern viel Tipparbeit damit hat, die einfach ihre Zeit kostet – unabhängig davon, welchen Schwierigkeitsgrad die Änderung hat.

Nicht umsonst wird der Formularbau noch von Entwicklern durchgeführt und immer noch nicht von „Key-Usern“; auch in Zeiten von SmartForms / SAP-Forms nicht und ich prognostiziere dass das auch nie der Fall sein wird. Alte Hasen wissen sehr wohl, dass schon SAPscript als Key-User-Werkzeug verkauft, aber von selbigen praktisch nie eingesetzt wurde. Und man kann es nicht oft genug sagen: Zu Beginn der Entwicklung muss eine komplett fertige Programmieranforderung zur Verfügung stehen. Eine Arbeitstechnik, die zu mehreren Hin- und Her-Änderungen führt (und das habe ich wahrlich sehr oft erlebt) lässt sonst die Kosten eines Formulars im wahrsten Sinne des Wortes explodieren.

Das Formulardesign (und der Begriff ist hier keineswegs nur optisch zu verstehen) ist ein bei vielen Entwicklern fast verhasstes Thema. Auf der anderen Seite haben sie Werkzeuge an der Hand, mit denen sie hochflexibel fast jeden noch so ausgefallenen Sonderwunsch umsetzen können (wenngleich das manchmal mit hohen

Aufwänden/Kosten verbunden ist). Und wenn man sich in die Grundprinzipien des Formularprozessors hineindenken kann, macht Formularentwicklung auch Spaß!

Dieser Text ist von der Homepage <http://www.heuristika.de/>