

21. Nov. 2018 · 4 Min. Lesezeit

Der SAP-Standard liefert keine Prozesse

Aktualisiert: 5. Aug. 2019

Es ist viel die Rede von „Prozessen“ im SAP-Umfeld. Es gibt Geschäftsprozesse, „Business Process Monitoring“, Softwareentstehungsprozesse, Softwarewartungsprozesse, Transportprozesse, Migrationsprozesse und viele mehr. Kaum jemand macht sich noch Gedanken darüber, was dieser Begriff für Software und ihre Verwendung bedeutet. Einer der prominentesten Geschäftsprozesse im SAP ERP ist der Wareneingangsprozess. Dieser besteht aus den folgenden Schritten:

- Eine Bestellung wird ausgelöst und als Beleg im SAP angelegt
- Die Bestellung wird an den Lieferanten versendet
- Die Ware wird geliefert, erfasst und ein Wareneingangsbeleg wird im SAP ERP angelegt

Dieses Beispiel zeigt alle wesentlichen Eigenschaften eines Geschäftsprozesses:

- Der Prozess hat einen zeitlichen Verlauf
- Der Prozess hat eine Richtung, d.h. es lässt sich ein Fortschritt messen
- Der Prozess hat zu jedem Zeitpunkt einen definierten Zustand
- Der Prozess hat verschiedene Bearbeiter; diese können seriell oder parallel tätig sein
- Der Prozess führt ein Protokoll mit sich
- Vergangene Prozesszustände müssen ggf. wiederherstellbar sein
- Es können SAP-Belege einbezogen werden oder erstellt werden (Bestellung, Materialbeleg, evtl. weitere)
- Es sind physische Dokumente (Formulare) einbezogen (z.B. Lieferschein, Bestellung)

Es gibt eine große Zahl von weiteren, ähnlichen Geschäftsprozessen, die im Allgemeinen über das SAP ERP abgewickelt werden.

Als Beispiele seien genannt:

- Rechnungseingang
- Reklamationseingang
- Vertriebsprozess (Kundenanfrage->Angebot->Auftrag)
- Inventarisierung

Der SAP-Standard liefert keine Prozesse

Als Geschäftsprozesse sind die genannten Prozesse jedoch nur auf organisatorischer Ebene im SAP ERP vorhanden. Der SAP-Standard liefert lediglich die Belege, zwischen denen die Prozesse ablaufen. Beim Wareneingang sind es z.B. der Bestellbeleg und der Materialbeleg, die im Laufe des Prozesses erstellt werden. Die Abwicklung des Prozesses ist dabei allein dem Benutzer überlassen.

Termine, Fortschritt, Zustand, involvierte Bearbeiter, Dokumente müssen vom Benutzer verwaltet werden. Es gibt keine zentrale Stelle, an der der Benutzer eine Übersicht der laufenden Prozesse gewinnen kann. Monitoring der Prozesse ist zeitaufwendig, umständlich und unzuverlässig, da eine Vielzahl von Datenquellen (Belegen) zur Auswertung herangezogen werden müssen. Bei der Prozessabwicklung kommt es zu häufigen Medienbrüchen: Es wird kommuniziert über Email, Telefon, Kurznachrichtendienste. Diese Kommunikation bleibt undokumentiert. Im Nachhinein ist es schwer, Entscheidungen nachzuvollziehen.

Der SAP Standard lässt hier offenbar – absichtlich oder unabsichtlich – eine Lücke erstaunlichen Ausmaßes.

Die Anforderungen an ein Prozessabwicklungswerkzeug

Will man ein Softwarewerkzeug schaffen, um die Geschäftsprozessabwicklung zu unterstützen, so ist es zielführend von speziellen Anwendungsfällen zu abstrahieren. D.h. man betrachtet z.B. die Anwendungsfälle „Rechnungseingang“ und „Wareneingang“ und versucht die gemeinsamen Eigenschaften und Anforderungen zu isolieren.

Businessobjekt Prozess

Es hat sich als sinnvoll erwiesen, den Prozess selbst als Businessobjekt zu betrachten.

Prozesslaufzeit

Prozesse können manuell vorangetrieben werden, d.h. vom Benutzer in einer Dialogtransaktion gesteuert, sie können aber auch automatisch fortschreiten.

Bei den gängigen Geschäftsprozessen kommen häufig beide Formen des „Vorantreibens“ vor. Daraus folgt, dass zum automatischen Antreiben der Prozesse eine Art „Prozesslaufzeit“ benötigt wird.

Monitoring versus Cockpit

Prozesszustand und -Fortschritt müssen überwachbar sein. Dazu wird eine Monitoringtransaktion benötigt, die alle wesentlichen Kennzahlen ausweist, sowie eine Sicht auf die Detaildaten des Prozesses bereitstellt. Erlaubt die Monitoringtransaktion außerdem die Steuerung des Prozesses, d.h. das Vorantreiben des Prozesses und das Ändern der Prozessdaten, so sprechen wir von Prozesscockpit.

Aus Softwaresicht kann Monitor und Cockpit dieselbe Transaktion sein, die unterschiedlich berechtigt wird.

Softwarearchitektur eines Prozessabwicklungswerkzeugs

Wir gehen aus von einer Entwicklung in ABAP OO, da sich aus der ABAP Laufzeit entscheidende Vorteile ergeben.

Monitoring und Cockpit

Das Monitoring liefert dem Anwender alle nötigen Informationen über Fortschritt, Anzahl und Zustand der Prozesse. Vom Basismonitor aus sollte das Protokoll jedes einzelnen Prozesses einsehbar sein. Außerdem sollte der Basismonitor den Neustart und das Debuggen eines einzelnen Prozesses erlauben und erfüllt insofern bereits einige, wenige Cockpit-Funktionen. Das Monitoring ist vollständig von Laufzeit und Handlerklasse entkoppelt. Es kann in jeder beliebigen UI-Technik implementiert werden (UI5, Mobile, SAPGUI, Windows).

Anwendungscockpit

Das Anwendungscockpit lehnt sich inhaltlich an den Monitor an, bietet darüber hinaus aber Sichten auf Anwendungsdaten und stellt auch anwendungsspezifische Funktionen bereit. Das UI stellt eine Erweiterung des Basismonitors dar. Die Umsetzung eines Cockpits lässt sich in sehr kurzer Zeit bewerkstelligen.

Handler-Klasse

Grundlage aller Anwendungsprozesse ist ein abstrakter Prozess ohne Anwendungsbezug. Dieser wird als einfache ABAP-OO Klasse (Handlerklasse) implementiert mit einigen, wenigen Eigenschaften:

- Die Klasse liefert ihre eigene Persistenz, d.h. liest sich von bzw. schreibt sich in die Datenbank
- Die Klasse liefert einen Sperrmechanismus, so dass immer nur ein Änderer zugelassen wird
- Die Klasse schreibt ihr eigenes Protokoll
- Die Klasse hat eine dezidierte Callback-Methode, die für „dunkle“ Abarbeitung von der Prozesslaufzeit aufgerufen wird

- Die Klasse ist vererbbar

Von dieser abstrakten Prozessklasse leiten alle anwendungsspezifischen Klassen ab.

Late Binding

Die Prozesslaufzeit sucht zur Ausführungszeit in einer Registrierungstabelle nach der Handlerklasse, die zum jeweiligen Prozess gehört (z.B. Handlerklasse zum Wareneingangsprozess). Die Handlerklasse wird zur Ausführungszeit instanziiert und ihre Callback-Methode wird von der Laufzeit aufgerufen.

Dieses Konzept der späten Bindung (Late Binding) lehnt sich an das Prinzip des Component Object Models (COM) an, das von Microsoft bereits 1992 etabliert wurde und bis heute in der neuen Windows 10 Runtime (WinRT) zum Einsatz kommt.

Die Lösung: das Universal Process Tool

Für Unternehmen kann es sehr lohnenswert sein, die „Prozesslücke“ auf diese Weise zu schließen:

- Es wird ein zentraler Einstiegspunkt geschaffen aus Prozesssicht, was häufig auch der Abteilungssicht entspricht
- Prozessdurchlaufzeiten werden verkürzt und messbar
- Fehler werden vollständig protokolliert
- Prozesse werden transparent: Reporting lässt sich als einfache Erweiterung des Monitorings umsetzen
- Die Projektumsetzungszeiten sind optimiert durch den hohen Grad an Wiederverwendung einmal vorhandener Softwarekomponenten
- Die Software passt sich dem Geschäftsprozess an und nicht umgekehrt
- Das Konzept ist zukunftssicher, da es sich mit minimalem Aufwand an neue UI-Technologien anpasst.
- Das Prinzip lässt sich auch auf technische Prozesse wie Migrationen oder asynchrone, parallelisierte Massenverbuchungen anwenden

Möchten Sie mehr zu unserer Lösung erfahren?

[DE](#)