

Flexible Geschäftsprozesse – auch mit individueller Software?

Flexible und kostengünstige Geschäftsprozesse, die sich an ändernde wirtschaftliche Bedingungen anpassen, sind ein wichtiger Wettbewerbsvorteil. Wie kann individuelle Software, die Geschäftsprozesse vorgibt und normalerweise zementiert, den Anforderungen nach Flexibilität gerecht werden?



Prof. Jürgen Spielberger
MSc ETH / MA HSG
Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
InIT Institut für angewandte Informationstechnologie
Schwerpunkt Enterprise Information Integration

Geschäftsprozesse können auf vielerlei Art dokumentiert werden. In den letzten Jahren hat die „Business Process Model and Notation“ (BPMN) viele Anhänger gefunden. Sie stellt mit grafischen Symbolen den Arbeitsablauf (Workflow) und das nähere Umfeld der Prozesse dar (z.B. Personen, Daten, Ereignisse). Diese Diagramme dienen als Spezifikation zur Programmierung der Software. Aus unternehmerischer Sicht wäre es ideal, wenn diese Diagramme direkt als Software verwendet werden könnten. Wenn diese Diagramme schon der „Softwarecode“ selbst sind. In diesem Fall könnten die in der Software abgebildeten Prozesse durch Anpassung der Prozessdiagramme flexibel an die aktuellen Bedingungen angepasst werden, so dass unsere im Titel gestellte Forderung nach Flexibilität erfüllt wäre - leider ist das die grosse Ausnahme. Wieso ist diese simple Idee nicht schon längst umgesetzt worden?

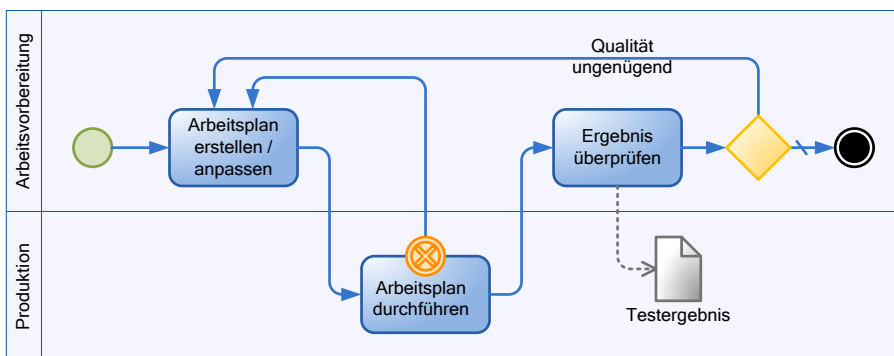


Abbildung 1: Einfaches Beispiel eines BPMN-Diagrammes

Softwareentwicklung heute

Auffallend in der Softwareentwicklung ist, und dies gilt nicht nur für individuelle Software, dass die Komplexität in den letzten 20 Jahren massiv zugenommen hat. Dies hängt mit der vielfältigen Infrastruktur (Vernetzung, Plattformvielfalt, Cloud), den anhaltend raschen Veränderungen, aber auch dem Konkurrenzdenken grosser IT-Firmen zusammen. In diesem Umfeld ist es schwierig, Software so zu entwickeln, dass diese sich leicht an veränderte Prozesslandschaften anpassen kann. Derart entwickelte Software ist deshalb in aller Regel starr und muss bei gewünschten Änderungen im Prozessablauf im Kern angepasst werden. Das ist aufwändig und teuer. Der Prozess wird darum erst dann angepasst, wenn die Kosten des mangelhaften Workflows, respektive Prozesses höher sind als die Kosten der Anpassung.

Methoden zur Entwicklung prozessorientierter, individueller Software

Unter den vielen Versuchen diese Probleme zu lösen sind auch Ansätze zu finden, welche die Prozesse und den Workflow thematisieren. Hier lassen sich drei grundsätzliche Lösungsrichtungen finden:

1. Dokument-Workflow-Systeme
2. Workflow-Management-Systeme
3. Modellgetriebene Softwareentwicklung

Da sind zunächst die *Dokument-Workflow-Systeme* (nicht zu verwechseln mit Dokumentmanagement-Systemen). Diese eignen sich zur Modellierung von dokumentenzentrierten Prozessabläufen. Dokumente werden hierbei definiert, durchlaufen Schritt für Schritt die vorgesehenen Stationen und werden vom entsprechenden Empfänger mit Informationen ergänzt. Am Schluss des Umlaufs steht das komplett ausgefüllte Dokument. Dokument-Workflow-Systeme sind in diesem eingeschränkten Einsatzfeld als individuelle Software durchaus sinnvoll einzusetzen und sind auch bezüglich Änderungen der Prozesslandschaft vergleichsweise anpassungsfähig. Sie lösen aber nur einen Teil des Problems. In den folgenden Ausführungen werden wir auf diese deshalb nicht weiter eingehen.

Im zweiten Ansatz finden sich *Workflow-Management-Systeme*, welche den Workflow als solchen steuern, die Ausführung des eigentlichen Prozesses (abgesehen von einfachen Schritten) aber an externe, eigens hierfür entwickelte Programme delegieren. Diese Systeme ‚orchestrieren‘ die Ausführung der Prozesse, indem Sie die passenden externen Programme aufrufen (z.B. Webservice via BPEL). Diese Entkopplung des Workflow-Management-Systems von den Programmen bleibt nicht ohne Folgen. Zwar ist es theoretisch möglich, den Workflow beliebig zu ändern. Dafür muss aber durch den Entwickler sichergestellt werden, dass der geänderte Workflow durch die externen Programme noch korrekt ausgeführt wird. Dies ist ohne exakte Kenntnis der Programme nicht möglich und erfordert häufig Anpassungen derselben. Umgekehrt sind Änderungen am Programm schwierig, da sichergestellt werden muss, dass dieses an allen Stellen des Workflow-Management-Systems weiterhin korrekt funktioniert (Round-Trip-Engineering). Dies führt in der Praxis dazu, dass Änderungen nur schwer möglich sind, dass Wandel in der Prozesslandschaft gemieden wird!

In der *modellgetriebenen Softwareentwicklung* (Model Driven Architecture, MDA) werden die Modelle (z.B. das BPMN-Diagramm) direkt zur Erzeugung des ausführbaren Systems verwendet. Eine Entkopplung wie bei den Workflow-Management-Systemen wird vermieden, indem die verschiedenen Aspekte einer Applikation (Daten, Geschäftslogik, User Interface, etc.) direkt miteinander verbunden werden. Um nun schnell und effizient auf Änderungen in der Prozesskette reagieren zu können, muss ein entsprechendes, integriertes Software-Entwicklungssystem die Definition des Prozessmodells ermöglichen. Zur Entwicklung von beliebig komplexer Business-Software sollten zusätzlich logische und technische Aspekte abgebildet werden können (z.B. Organigramm und Rechte, Deployment).

Es geht! Was muss beachtet werden?

An der Zürcher Hochschule für angewandte Wissenschaften (ZHAW) wurde in zwei durch den Bund geförderten Projekten (KTI, Förderagentur für Innovation des Bundes), in Zusammenarbeit mit der ETH Zürich und der Firma Posity, eine modellgetriebene, grafische Entwicklungsumgebung (Posity Design Studio™) konzipiert und implementiert. Damit entwickelte, individuelle Cloud-Applikationen sind bereits bei mehreren Kunden im Einsatz. Die folgenden Ausführungen konzentrieren sich auf das Prozessmodell (Abbildung 2) und vernachlässigen die anderen Diagrammart innerhalb des Posity Design Studios (z.B. Datenmodell, Moduldiagramm, etc.).

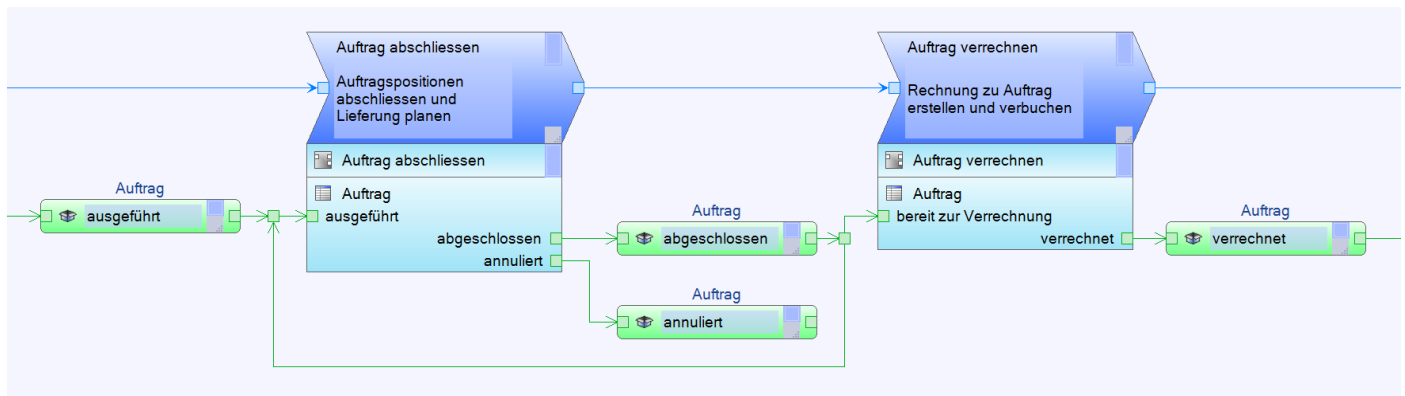


Abbildung 2: Ausschnitt eines Prozessdiagrammes des Posity Design Studios mit Workflow der Prozesse (blau) und Stateflow der Daten (grün)

Wichtig ist, dieses Diagramm ist der Programmcode! Änderungen an diesem Diagramm wirken sich sofort auf das Verhalten der Applikation aus. Die Prozesse (blaue Prozesspfeile) sind mit Modulen verknüpft (hellblaue Rechtecke). In diesen Modulen ist die Geschäftslogik abgebildet (als visuelle Datenflussdiagramme). Die blauen Pfeile definieren den Workflow eines Anwenders. Hat der Anwender z.B. den Prozess ‚Auftrag abschliessen‘ vollendet, wird automatisch der Prozess ‚Auftrag verrechnen‘ gestartet. Als Erweiterung zum BPMN-Standard wird der Stateflow (grüne Pfeile) festgehalten. Diese Erweiterung ist absolut zentral. Der Stateflow definiert, wie der Status (grüne Boxen) der Daten durch die Prozesse verändert werden kann. Gemäss Prozessdiagramm ist es z.B. nicht möglich Aufträge zu verrechnen, die noch nicht abgeschlossen wurden. Soll dies neu möglich sein, kann das durch Erweiterung des Prozessdiagrammes implementiert werden (es muss ein neuer Status eingeführt werden), ohne dass die Geschäftslogik erweitert werden muss.

Wir haben die Erfahrung gemacht, dass folgende Eigenschaften unterstützt werden müssen, um die Geschäftsprozesse möglichst effizient anpassen zu können:

1. Das Prozessmodell muss sauber *getrennt von der Geschäftslogik* definiert werden können, aber vollständig in die Applikation integriert sein (Round-Trip-Engineering verhindern). Zusätzlich müssen im Prozessmodell mindestens Verzweigungen (Gateways) und Ereignisse (Events) definiert werden können, welche den Workflow beeinflussen.
2. Der *Stateflow* der Daten muss zwingend im Prozessmodell festgehalten werden. Nur so ist es möglich, die meisten Änderungen im Prozessablauf im Prozessmodell selbst zu gestalten. Andernfalls resultieren Anpassungen in der Geschäftslogik und die Flexibilität ist stark eingeschränkt.
3. Das Prozessmodell muss auch *Parameter* erlauben (z.B. Zahlungsziel der Rechnung) um wichtige Einflussgrößen sichtbar und änderbar zu machen.
4. Das Prozessmodell muss verschiedene *Ebenen des Prozessmodells* darstellen können (z.B. Prozesslandkarte auf oberster Ebene, Prozesse mit auszuführender Geschäftslogik auf unterster Ebene) und diese Ebenen hierarchisch miteinander verknüpfen. Das ermöglicht eine saubere Strukturierung und Dokumentation der Applikation.

Wir möchten mit folgender positiver Aussage schliessen: Individuelle Business-Software flexibel, günstig und schnell an ein sich änderndes Umfeld anzupassen ist möglich, wenn das Entwicklungssystem im Kern die umfassende Verwaltung von Geschäftsprozessen ermöglicht.