

Überführung papierbasierter Antragsprozesse in die elektronische Abwicklung bei einem Finanzdienstleister

Learns aus dem Projektmanagement in SAP-Großprojekten



Im klassischen Projektmanagement können einige weiche Faktoren über Erfolg oder Misserfolg entscheiden.

Wie behält man bei einem SAP-Projekt mit einem Gesamtbudget im mittleren einstelligen Millionenbereich, einem reinen Entwicklungsvolumen von einigen Tausend Tagen und rund 40 beteiligten Personen in einem Zeitraum von eineinhalb Jahren Budget, Qualität und Zeit im Griff? Noch dazu, wenn verschiedene Dienstleister und Auftraggeber in einem Projekt vereint sind und darüber hinaus noch teils unbekannte Technologien eingesetzt werden. Es sind im klassischen Projektmanagement einige weiche Faktoren, die dann über Erfolg oder Misserfolg entscheiden können.

Von Björn Kibbel*

Das hier geschilderte Großprojekt bestand darin, bei einem Finanzdienstleister papierbasierte Antragsprozesse in eine elektronische

Abwicklung zu überführen. Dazu wurde ein auf SAP-Systemen basierendes Online-Portal mit Zugriff auf die Backend-Systeme entwickelt. Die Antragsdaten sollten mit der SAP-Formulartechnologie „SAP Interactive Forms by Adobe“ und

dem SAP-CDP-Produkt (CDP = Custom Development Project) „SAP Online Antragsmanagement“ (OAM) in das Backend eingespielt werden. Nach Bewilligung der Anträge sollte es möglich sein, Geschäftsvorfälle wie Antragsänderungen oder Mittelabrufe vom Kunden zu prozessieren. Um die mit dem Antragsverfahren verbundenen Genehmigungsprozesse zu steuern, wurde das „SAP Business Rules Framework“ (BRF) gewählt.

*Björn Kibbel (45) ist seit 1999 bei der innobis AG tätig. Als Manager Development und Integration Services analysiert er bankfachliche Geschäftsprozesse und die damit verbundenen Anforderungen an die IT-/SAP-Landschaft und übernimmt die Architekturberatung bei Banken. Der studierte Diplom-Physiker ist ein erfahrener Projektleiter.

Der Entwicklung war eine mehrmonatige Fachkonzeptphase vorausgegangen, in der parallel bereits die architektonischen Grundlagen und Datenverarbeitungskonzepte entwickelt wurden. Aufgrund der sehr hohen Sicherheitsanforderungen wurde eine mehrschichtige Architektur gewählt und mit Firewalls und Intrusion-Detection-Systemen entsprechend abgesichert. Das Online-Antragsmanagement basierte auf „SAP Net-Weaver Java“, das Backend wurde auf Grundlage von „SAP EHP 6.0“ mit „CML“ (Loans Management; Darlehensverwaltung) entwickelt. Ein Application Layer auf ABAP-Basis (ABAP = Advanced Business Application Programming) diente als „Vermittler“.

Flexibler Schnitt der Arbeitspakete

Die Entwicklung des Online-Portals wurde in fünf Arbeitspakete unterteilt und der gleichen Anzahl an Teams übergeben. Eine große Gefahr beim Schnitt der Arbeitspakete besteht darin, sie vollständig voneinander abgegrenzt anzulegen. Dies geschieht dann besonders leicht, wenn man sie ausschließlich nach technischen Gesichtspunkten – beispielsweise nach Systemkomponenten – schneidet. Erfolgreicher ist die Planung von mindestens ein oder mehreren Teams, die einen fachlichen Prozess umsetzen. Diese Teams sind auf die Basiskomponenten der anderen Teams angewiesen. Entstehen bei der Entwicklung der Komponenten Lücken in der fachlichen Abdeckung, werden diese von den Prozessteams aufgedeckt. Da die Prozessteams in den ersten Iterationen noch keine Systembasis haben, auf der sie aufsetzen können, werden sie zu Beginn mit wenigen Ressourcen ausgestattet. Mitarbeiter aus den Komponententeams wechseln dann nach zwei bis drei Iterationen in die Prozessteams, sodass diese wachsen und die Komponententeams wieder kleiner werden. Das Know-how rund um die Komponenten wird so in die Prozessteams mitgebracht. Im oben genannten Praxisbeispiel wurden je ein Prozessteam für die Antragsanlage und die Abbildung der Geschäftsvorfälle gebildet.

Wohlüberlegte Dauer der Iterationen

Die Entwicklungsphase wurde wie im klassischen Projektmanagement nach Prince2 in Iterationen unterteilt. Bei der Wahl der Iterationsdauer sind mehrere Argumente (s. Tabelle auf der nächsten

Seite) gegeneinander abzuwägen. Aufgrund des sehr engen Gesamtzeitplans und des gedeckelten Budgets wurden etwa dreiwöchige Iterationszeiträume geplant, um die Entwicklungsergebnisse möglichst oft kontrollieren zu können. Es zeigte sich, dass gerade in den ersten Iterationen der Overhead für Iterationsplanung, Testfallerstellung, Vor-test etc. die Teams sehr stark belastete. Damit einhergehend waren die vorzeigbaren und testbaren Ergebnisse so wenig aussagekräftig, dass eine wirksame Kontrolle nicht im erhofften Maß stattfinden konnte. Die ersten beiden Iterationen sollten daher in Projekten dieser Größe etwas länger gestaltet werden. Andererseits kann eine Fehlentwicklung gerade in den ersten Iterationen zu großen Problemen im späteren Verlauf der Entwicklung führen. Um dieses Problem im Griff zu behalten, muss der Entwicklungsleiter und Architekt in engem Kontakt zu den Teams stehen und sich regelmäßig einen Überblick über die laufenden Entwicklungen verschaffen. Diese Aufgabe ist bei der Projektgrößenordnung eine Fulltime-Aufgabe und muss in der Aufwandsplanung von vorneherein mitberücksichtigt werden.

Testfalldefinition zu Beginn der Iteration

Zu Beginn einer Iteration definierten die Arbeitspaketleiter die Testfälle, die am Ende der Iteration durchführbar sein sollten. Aufbauend auf diesen führten sie die Iterationsplanung durch und benannten Einschränkungen der Testfälle wie beispielsweise genutzte, aber erst vorläufig ausgeprägte Schnittstellen (auch von anderen Arbeitspaketen). Dieses Verfahren hat sich bewährt, da die Arbeitspaketleiter sich bereits am Anfang einer Iteration tief gehende Gedanken machen müssen, wie ein Stück Software umzusetzen ist. Ebenfalls bewährt hat es sich, dass alle Arbeitspaketleiter die Arbeitspläne aller Teams gemeinsam bewerten. So können sie zusätzliche Anforderungen der Teams untereinander abstimmen.

Regelmäßiges Zusammenstellen der Softwareauslieferungen

In diesem Projekt wurde auf einem zentralen System entwickelt und auf unterschiedliche Schichten des späteren Gesamtsystems ausgeliefert. Dabei ist das regelmäßige Zusammenstellen der Auslieferungen am Ende einer Iteration sehr zu empfehlen, offenbart es doch Auslieferungsprobleme, zum Beispiel

bei unberücksichtigten Abhängigkeiten. Diese müssen bereinigt werden. Erfolgt die Korrektur dagegen erst kurz vor dem Testbeginn, können erhebliche Aufwände entstehen, die in keiner Budget- oder Zeitplanung vorgesehen waren.

Gesamtumfang im Blick behalten

Bei der vorgestellten iterativen Sichtweise auf die Entwicklung und dem Druck, der auf dem Entwicklungsteam am Ende einer Iteration lastet, neigen Arbeitspaketleiter tendenziell dazu, für die nächste Iteration weniger Inhalte einzuplanen, um diesem Druck auszuweichen. Dabei kann es leicht passieren, dass der Gesamtumfang der Aufgaben, die das Team bewältigen muss, aus dem Blickfeld gerät. Um dem entgegenzuwirken, hat es sich bewährt, parallel zur Iterationsplanung immer auch eine Gesamtplanung über alle Arbeitspakete zu erstellen. Diese erfolgt in Form einer sehr großen Mind-Map, in der pro Arbeitspaket und Iteration der geplante Umsetzungsumfang in Form von Features – wenn möglich mit Bezug zum Fachkonzept – eingetragen wird. Dies dient dann auch den Teams, um untereinander abzustimmen, wann welches Feature in welcher Ausprägung benötigt und entwickelt wird.

Der Entwicklungsleitung hilft diese Matrix, die Iterationen zu planen und den Projektfortschritt zu kontrollieren. Im vorgestellten Projekt hing eine Mind-Map ausgedruckt auf einem A0-Plakat an der Wand. Fertige Features wurden mit einem grünen Marker gekennzeichnet, sodass der Projektfortschritt wie „ein Baum im Frühling“ sichtbar wurde, was das Team zusätzlich motivierte.

Qualität sichern

Zwischen Entwicklungsende und Fachtest wurde ein umfangreicher Entwicklungstest eingeplant, in dem die Architekten und Arbeitspaketleiter die Gesamtanwendung umfassend testeten. Iterationstest lassen sich aufgrund der verfügbaren Zeit meist nur relativ oberflächlich durchführen. Eine Gefahr bei dieser Vorgehensweise besteht darin, dass Entwickler ein Entwicklungsobjekt nach bestandener Iterationstest als abgeschlossen betrachten und die fehlenden zehn Prozent bis zur endgültigen Fertigstellung aus Zeitmangel auf der Strecke bleiben. Auch werden am Iterationstest in der Regel keine Schwachstellen, wie beispielsweise fest verdrahtete Parameter, aufgedeckt.

Es ist illusorisch anzunehmen, dass Entwickler unter zeitlichem Druck nicht sogenannte „Erstmal“-Implementierungen als „fertig“ kennzeichnen. 50 Prozent solcher Defekte können die Prozesssteams im Laufe ihrer Implementierung aufdecken. Die übrigen 50 Prozent lassen sich durch einen entsprechend ausgedehnten Entwicklungstest identifizieren, der nicht durch die Entwickler selbst durchgeführt wird, sondern vom Entwicklungsleiter oder Architekten. Dieser sollte als wirksamstes Mittel gegen Softwarequalitätsmängel von vorneherein im Projekt eingeplant werden.

Fazit

Das SAP-Großprojekt wurde erfolgreich fertiggestellt. Die Learnings aus diesem Projekt seien hier nochmals kurz zusammengefasst: Schnitt der Arbeitspakete nach technischen und fachlichen Gesichtspunkten und bedarfsorientierter Wechsel der Mitarbeiter zwischen den Teams; gutes Abwägen zwischen kurzen und langen Iterationen; Definition der Testfälle bereits zu Beginn jeder Iteration; Zusammenstellen der unterschiedlichen Auslieferungen am Ende jeder Iteration; Gesamtumfang im Blick

	Kurze Iterationen	Lange Iterationen
Organisatorischer Overhead (Iterationsplanung, Testfalldefinitionen, Testdurchführung etc.)	Relativ hoch, da mehr Iterationen im Zeitraum durchgeführt werden.	Niedriger, da weniger Iterationen im Zeitraum durchgeführt werden.
Iterationsergebnisse	Besonders in den ersten Iterationen ist nur wenig Vorzeig- und Testbares vorhanden.	Besonders in den hinteren Iterationen kann der Testaufwand aufgrund der Menge der erzeugten Features erheblich werden.
Kontrolle der Ergebnisse	Es gibt häufig die Möglichkeit des Eingriffs und der Steuerung.	Es besteht größere Gefahr, dass etwas in die falsche Richtung entwickelt wird.
Disziplin	Die häufigere Replanung wird erzwungen, in die auch neuere Erkenntnisse einfließen können.	Entwicklungsprobleme lassen sich leichter im Arbeitspaket „verstecken“ und treten erst später zutage.

behalten. Klassische iterative Projektmanagementmethoden eignen sich nach wie vor hervorragend, um komplexe und umfangreiche Entwicklungsprojekte mit vorgegebenem Scope umzusetzen. Einige hier beschriebene Learnings bzw. weiche Erfolgsfaktoren tragen dazu bei,

dass manche Projekte erfolgreicher verlaufen als andere und es lohnt sich für Projektleiter, sich diese Faktoren und ihre Auswirkungen am Anfang des Projektes bewusst zu machen und aktiv zu managen.

(ap) @