

Prozessorientiertes Requirements Management - und der Roundtrip kann bleiben, wo der Pfeffer wächst

Sebastian Graf
PROMATIS software GmbH
Pforzheimer Strasse 160, 76275 Ettlingen

Schlüsselworte:

Geschäftsprozess-Management, BPM, Modellierung, Simulation, Referenzmodelle, Workflow, BPEL, Monitoring, Roundtrip, Risk Management

Einleitung

Im Kontext der Implementierung immer komplexer werdender IT Systeme ist eine fundierte, qualitativ hochwertige und vollständige Anforderungsanalyse ein absolutes Muss. Niemand wird diese Aussage ernsthaft bestreiten, und doch ist die Anforderungsanalyse und insbesondere die prozessorientierte Analyse in vielen Projekten ein ungeliebtes Stiefkind. Fragt man nach den Gründen, so begegnet man immer den gleichen bekannten Verdächtigen: keine Zeit, kein Budget, kein Werkzeug, keine ausgebildeten Ressourcen und die immer wieder gestellte Frage, wo denn der Nutzen einer prozessorientierten Analyse für das Projekt liegt. Verfolgt man Projekte über die diversen Phasen, so stellt man fest, dass sich Fehler aus der Analyse in das Design und in die Implementierung fortpflanzen, was zu kostspieligen Rework-Szenarien führt. Um dieser Problematik Herr zu werden, hat man den vorhandenen Entwicklungswerkzeugen neuerdings kurzerhand einen Roundtrip verordnet, also die technologische Möglichkeit geschaffen, Projekt-Artefakte von der Entwicklungs- bzw. Designphase wieder zurück auf die analytische Ebene zu generieren. Dahinter verbirgt sich die Philosophie, dass Analysefehler, die es bis in die Implementierung geschafft haben, dort von den Softwareentwicklern korrigiert werden und die Resultate der Korrektur dann per Reverse Engineering auf die Analyseebene zurücktransportiert werden können. Zugegeben, solche Konzepte können in begrenztem Umfang durchaus hilfreich sein, aber es führt nichts an der Erkenntnis vorbei, dass der Königsweg zum Projekterfolg eben doch über eine hochqualitative Analyse führt. Der vorliegende Beitrag beschreibt die wesentlichen Aspekte, auf die bei einer prozessorientierten Anforderungsanalyse zu achten ist, um eben nicht auf einen vermeintlich rettenden Roundtrip angewiesen zu sein.

Warum IT-Projekte scheitern

Es gibt sie tatsächlich – IT-Projekte, die scheitern. Und glaubt man der renommierten Standish Group (<http://www.standishgroup.com>), dann handelt es sich dabei keineswegs um eine Randerscheinung: Laut dem CHAOS Report der Standish Group sind in den Jahren 2004 – 2008 durchschnittlich 20% der IT-Projekte gescheitert, 48% sind nur knapp dem Scheitern entgangen, und nur 32% wurden erfolgreich abgeschlossen. Für 2010 prognostiziert der Report sogar eine Zunahme der gescheiterten Projekte auf insgesamt 32% (vgl. Abbildung 1).

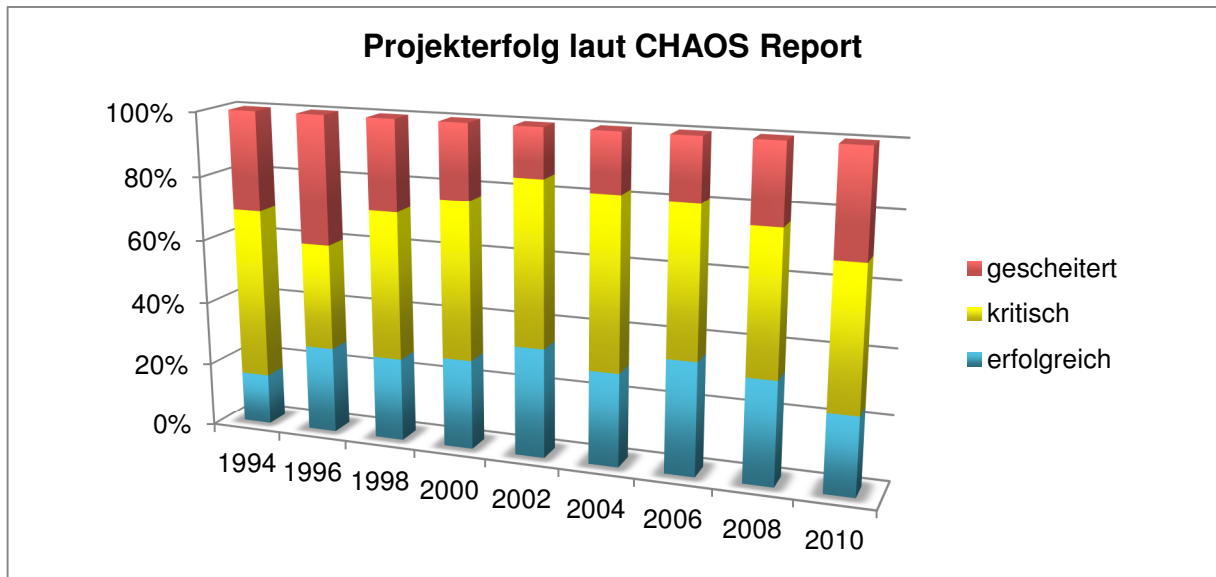


Abb. 1: CHAOS Report 1994 - 2010, Standish Group

Die im Report genannten Gründe für das Scheitern sprechen ebenfalls eine deutliche Sprache: In 13% der Fälle wurde eine unvollständige Anforderungsanalyse als der Hauptverursacher identifiziert, in 12% wurden Business User nicht in ausreichendem Maße miteinbezogen, in 10% wurden unrealistische Annahmen gemacht, und in 9% wurden sich ständig ändernde Anforderungen gar nicht erkannt. Addiert man diese Zahlen, dann stellt man fest, dass beim Scheitern eines Projekts Mängel in der Anforderungsanalyse zu 44% beteiligt sind. Mangelndes technisches Know-How hingegen wurde nur in 4% der Fälle als Verursacher identifiziert (vgl. Abbildung 2).

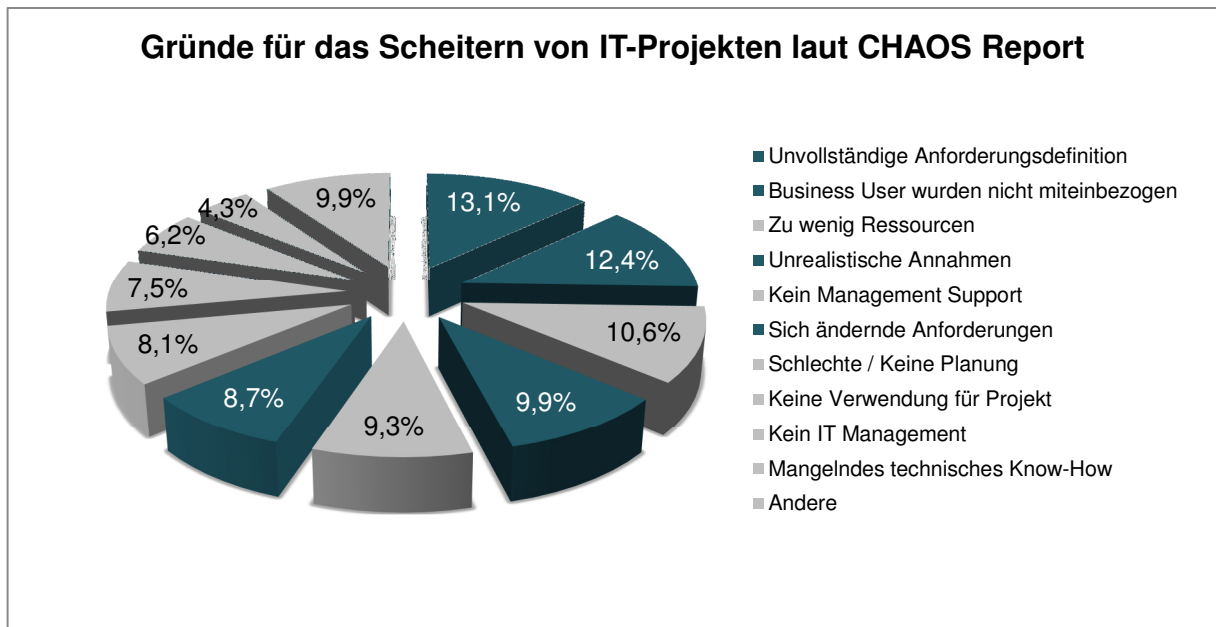


Abb. 2: Details CHAOS Report, Standish Group

Die Schlussfolgerung aus diesen geradezu niederschmetternden Zahlen kann nur darin bestehen, dass der Schlüssel zum Projekterfolg in einer qualitativ hochwertigen Anforderungsanalyse zu suchen ist. Aus diesem Grunde hat das Thema BPM (Business Process Management) zu Recht eine noch vor wenigen Jahren nicht für möglich gehaltene Renaissance erlebt.

Warum der Roundtrip kein Mittel gegen das Scheitern ist

Viele Software-Hersteller, so auch Oracle, haben BPM-Werkzeuge mittlerweile fest in ihrem Produktportfolio verankert und propagieren BPM als einen fundamentalen und integralen Bestandteil der Software-Entwicklung. Es wird sogar suggeriert, dass diese BPM-Werkzeuge mit bidirektionalen Schnittstellen ausgerüstet sind, die den Austausch von Analyseartefakten mit den entsprechenden Entwicklungswerkzeugen in beide Richtungen erlauben. Im Oracle Umfeld stehen hier beispielsweise die BPA Suite, die BPM Suite oder Horus® für die Analyse sowie die SOA Suite für die Entwicklung zur Verfügung, und viele Anwender sind auch durchaus gewillt, diese Produkte in ihren Projekten zum Einsatz zu bringen.

Was jedoch häufig vergessen wird, ist die einfache Tatsache, dass eine Anforderungsanalyse nicht nur geeignete Werkzeuge, sondern auch eine dazu passende Vorgehensweise sowie gut ausgebildete Analysten erfordert, die aufgrund ihrer Tool-Kenntnisse und Erfahrung in der Lage sind, sicherzustellen, dass in der Analysephase auch tatsächlich hochqualitative Ergebnisse entstehen. Nicht selten stellt man daher in der Implementierungsphase eines Projekts fest, dass die auf Basis der Ergebnisse der Anforderungsanalyse generierten Entwicklungsartefakte, wie z.B. BPEL-Prozesse, so nicht umsetzbar sind; und im schlimmsten Fall erfüllen diese Prozesse sogar nicht einmal die fachlichen Anforderungen. Um dieser im Übrigen weitverbreiteten Problematik Herr zu werden, haben die einschlägigen Tool-Hersteller in ihre Werkzeuge einen Roundtrip eingebaut, mit dem nachträgliche Korrekturen an den Entwicklungsartefakten auf die Analyseebene zurückgespielt werden können. Somit soll es möglich sein, Fehler aus der Analyse relativ einfach und mit geringem Aufwand in der Entwicklungsphase zu korrigieren und die Anpassungen anschließend wieder auf die Analyseebene zu transportieren. Dass es sich dabei aber um einen fatalen Irrweg handelt, der keineswegs für Sicherheit im Projekt sorgt, wird schnell klar, wenn man dieses Vorgehen im Detail betrachtet.

Die im Folgenden beschriebenen Probleme lassen recht schnell erkennen, dass das oben beschriebene Roundtrip-Verfahren keineswegs geeignet ist, Probleme, die auf eine unzureichende Analyse zurückzuführen sind, nachträglich zu korrigieren:

1. Das Roundtrip-Verfahren geht von der Annahme aus, dass ein Softwareentwickler im Entwicklungsprozess erkennt, dass die aus den Analyseartefakten abgeleiteten Development-Blueprints fehlerhaft sind. Diese Annahme ist äußerst kritisch, da Softwareentwickler nur sehr selten und auch nur in kleineren Projekten überhaupt über das notwendige fachliche Know-How verfügen, um diese Bewertung durchführen zu können. Typischerweise werden diese Fehler von der Entwicklungsmannschaft eben genau nicht erkannt und treten erst in der Testphase zutage, wenn die Anwender das System zum ersten Mal live erleben.
2. Das Verfahren geht davon aus, dass der Entwickler auf der Implementierungsebene die notwendigen Korrekturen vornimmt. Wie aber soll das gelingen? Schließlich würde das bedeuten, dass der Entwickler, beispielsweise mit BPEL-Prozessen ausgerüstet, gemeinsam mit den Fachanwendern eine Korrektur an diesen Prozessen durchführt. Dieses Vorgehen hört sich zwar recht praktisch an, man sollte dabei aber entgegen aller landläufiger Meinungen berücksichtigen, dass es sich bei BPEL-Prozessen (nach Auffassung des Autors gilt Ähnliches auch für BPMN-Prozesse) um sehr technische Artefakte handelt, die sich in keiner Weise für einen zielführenden Dialog mit einem Fachanwender oder gar dem Management eignen.

3. Schließlich geht das Verfahren von der Annahme aus, dass sich die unterschiedlichen Modelle aus der Analyse- und der Implementierungsphase nahezu verlustfrei ineinander überführen lassen. Und selbstverständlich soll diese Transformation auch noch bidirektional möglich sein. Dabei handelt es sich um den in regelmäßigen Abständen wiederkehrenden Traum der IT-Gemeinde, zwei formale Beschreibungsverfahren, welche mit unterschiedlicher Syntax und Semantik ausgestattet sind, generativ ineinander abzubilden. Die Erfahrung zeigt hier aber ganz klar, dass bei solchen Transformationen stets mit Verlusten zu rechnen ist, und je unterschiedlicher die beiden Verfahren sind, desto größer sind auch die zu erwartenden Verluste. Betrachtet man beispielsweise die Konvertierung von Ereignisgesteuerten Prozessketten (EPK), wie sie u.a. in der BPA Suite zum Einsatz kommen, in BPEL-Prozesse, dann kann hier von einer einfachen und vollständigen Konvertierung nicht die Rede sein.

Letztlich sind somit für Projekte die beiden folgenden Szenarien denkbar:

1. In der Implementierungsphase treten nur noch wenige Unstimmigkeiten zutage (was auf eine qualitativ hochwertige Analyse zurückzuführen ist). In diesem Fall ist es sicher kein kritischer Umstand, wenn die Werkzeuge der Analyse- und der Implementierungsphase über keinen Roundtrip verfügen, und die wenigen notwendigen Anpassungen auf beiden Ebenen manuell nachgepflegt werden müssen.
2. In der Implementierung wird tatsächlich festgestellt, dass große Teile der generierten Blueprints fachlich nicht in Ordnung sind (was wiederum auf dramatische Versäumnisse in der Analyse zurückzuführen ist). In diesem Fall steckt das Projekt in größten Schwierigkeiten und ist ein Kandidat, der im kommenden Standish Report möglicherweise in der Rubrik „gescheitertes Projekt“ wiederzufinden ist. Ein Roundtrip wird daran aber nichts ändern.

BPM-Werkzeuge sinnvoll eingesetzt

Die Schlüsselfrage ist also nicht, ob es Sinn macht, BPM-Methoden und -Werkzeuge in Software-Projekten zum Einsatz zu bringen oder nicht, vielmehr ist die entscheidende Frage, wie BPM-Methoden sinnvoll angewendet werden können, um die angesprochenen Projektrisiken zu minimieren.

Auswahl des geeigneten Werkzeugs

Der Einsatz eines BPM-Werkzeugs startet in der Regel mit der Auswahl des geeigneten Werkzeugs. Bereits in dieser frühen Phase sind grundlegende Dinge zu beachten, und nicht immer ist eine Ein-Hersteller-Strategie der beste Ratgeber. Vielmehr sollten die folgenden Überlegungen im Mittelpunkt stehen:

- Können über das Werkzeug alle für den weiteren Verlauf des Projekts benötigten Informationen abgebildet werden? Alle auf dem Markt verfügbaren Werkzeuge erlauben die Abbildung von Ablaufmodellen und Organisationsstrukturen. Darüber hinaus kann es aber notwendig sein, die Geschäftsobjekte, die Gegenstand des Prozesses sind, strukturiert beschreiben zu können, um später beispielsweise Datenmodelle oder Serviceschnittstellen ableiten zu können. Weitere Artefakte, die bei der Dokumentation der Anforderungen von großem Interesse sein können, sind z.B.: Geschäftsregeln, Rollen, Ressourcen, Kennzahlen, Risiken, Ziele und Leistungen, um nur einige zu nennen.
- Wie detailliert können die jeweiligen Artefakte beschrieben werden? Hier lohnt sich ein genauer Blick auf das jeweilige Werkzeug. Wird eine Aktivität eines Prozesses z.B. nur durch ein grafisches Objekt repräsentiert, bei dem noch ein beschreibender Text hinterlegt werden

kann, oder ist es möglich, dort weitaus detailliertere Informationen wie Kosten, Zeiten, Ausführungsregeln und dergleichen mehr zu hinterlegen?

- Wie stark vernetzt sind die einzelnen Modelltypen? Hier ist zu bewerten, ob es möglich ist, Artefakte aus einem Modelltyp mit den Artefakten aus einem anderen Modelltyp zu verbinden. Als Beispiel sei hier die Verknüpfung von Objekten aus dem Ablaufmodell mit Objekten aus dem Objektmodell genannt. Die Beschreibung von Objekten im Ablaufmodell erfolgt typischerweise nur sehr oberflächlich. In der Regel wird hier nur ein grafisches Objekt definiert, welches mit einem Kommentar versehen wird, der die Struktur des Objekts in Prosa beschreibt. Soll aus diesem Prozess später aber eine Service-Schnittstelle oder eine Datenbankstruktur abgeleitet werden, dann ist es unerlässlich, dieses Objekt in einem Objektmodell detailliert und wohlstrukturiert zu beschreiben.
- Wie einfach zu bedienen ist das Werkzeug? Der Einsatz eines BPM-Werkzeugs steht und fällt mit der Akzeptanz der Anwender. Weniger ist hier oft mehr. Das umfangreichste Paket ist nutzlos, wenn die Anwender keine Expertise in der Bedienung haben und nicht klar ist, wie mit dem Werkzeug umzugehen ist, um einerseits alle Anforderungen zu erfassen und andererseits sicherzustellen, dass die Ergebnisse der Analysephase auch wirklich in weiteren Projektphasen genutzt werden können.
- Über welche Schnittstellen verfügt das Werkzeug? Hier ist vor Start des Projekts zu klären, welche Artefakte die Entwicklungsmannschaft nach der Analyse erwartet und wie diese aus dem BPM-Werkzeug extrahiert werden können. Die meisten aktuell auf dem Markt verfügbaren BPM-Werkzeuge sind beispielsweise in der Lage, BPEL Blueprints zu generieren. Nur wenige Werkzeuge können aktuell schon BPMN generieren oder sind beispielsweise in der Lage, analytische Objektmodelle in UML-Modelle oder physische Datenbank-Modelle zu überführen.
- Ist konsistentes hierarchisches Modellieren auf einfache Art und Weise möglich? Die meisten BPM-Werkzeuge bieten die Möglichkeit, hierarchisch zu modellieren, um auch bei der Analyse einen Top-Down-Ansatz zu unterstützen. In vielen Fällen werden jedoch untergeordnete Modelle ohne ausreichende logische Verbindung neben die korrespondierenden übergeordneten Modelle gestellt. Dieser Umstand kann später zu Modellinkonsistenzen führen und in Summe zu logisch falschen Modellen mit den entsprechenden Auswirkungen in der Implementierungsphase.

Der prozessorientierte Ansatz

Nachdem das geeignete Werkzeug identifiziert wurde, ist sicherzustellen, dass das Werkzeug auch nutzbringend eingesetzt wird. Hier ist oftmals ein Umdenken der Tool-Anwender erforderlich. In vielen BPM-Projekten kann man beobachten, dass BPM-Werkzeuge missbraucht werden, um Anforderungen rein funktional und somit mehr oder weniger zusammenhangslos zu dokumentieren. Ist dieser Irrweg einmal eingeschlagen, dann kann ein Scheitern des Projekts bereits in dieser frühen Phase prognostiziert werden. Sehr verlockend ist dieser Weg übrigens bei SOA-Projekten, bei denen bereits in einer frühen Projektphase die Gefahr besteht, in Services und nicht in Prozessen zu denken. Beim Einsatz von BPM-Methoden muss allen Beteiligten klar sein, dass das zentrale Objekt in der Anforderungsanalyse ohne Wenn und Aber der fachliche Prozess ist. Sehr oft hört man hier das Argument, dass man eigentlich gar keinen Prozess habe, sondern nur ein System entwickeln möchte, welches erst dieses und dann jenes tut, und wenn der Fall X auftritt, dann muss man darauf in einer bestimmten Art und Weise reagieren. Ohne es zu wissen, hat der geneigte Kritiker damit aber bereits begonnen, einen Prozess zu beschreiben. Ferner ist zu berücksichtigen, dass Fachanwender, die einen fundamentalen Anteil an der Analysephase haben müssen, weil nur diese das Wissen über fachliche Abläufe haben (was von IT-Spezialisten oft vergessen oder gar ignoriert wird), grundsätzlich bewusst oder unbewusst ausschließlich in Prozessen denken. Möchte man also ein vollständiges Bild der

Anforderungen gewinnen, so sollte man es tunlichst unterlassen, mit Anwendern über Funktionen oder Services zu diskutieren.

Knowledge Bases / Referenzmodelle

Je umfangreicher das Projekt wird, welches durch BPM-Methoden unterstützt werden soll, umso größer wird auch der Aufwand, der in die Prozessanalyse investiert werden muss. Dies gilt z.B. für umfangreiche Enterprise Resource Management-Projekte. Für solche Projekte lohnt es sich, nach sogenannten Knowledge Bases oder Referenzmodellen Ausschau zu halten. Einige BPM-Tool-Hersteller, wie z.B. die Horus software GmbH, haben erkannt, dass ein BPM-Werkzeug alleine noch kein Garant für den erfolgreichen Einsatz dieses Werkzeugs ist. Viel wichtiger ist oftmals die Bereitstellung von entsprechendem Content, der dem Anwender dabei hilft, die ersten Schritte mit dem Werkzeug zu machen, ohne auf „einem leeren Blatt Papier“ beginnen zu müssen.

Ferner verbirgt sich in guten Knowledge Bases das Know-How aus unzähligen Projekten, was ein unschätzbare Vorteil für das eigene Projekt sein kann.

Simulation von Geschäftsprozessen

Das Thema Simulation spielt in der IT-Branche bis heute leider immer noch eine untergeordnete Rolle. Fast scheint es, als ob die Gilde der Software-Ingenieure sich hier im Vergleich zu ihren Kollegen eine Sonderstellung herausnehmen wollte: Für einen Automobilhersteller ist es selbstverständlich, dass ein neues Fahrzeug, bevor es überhaupt gebaut wird, Millionen von Kilometern in einem Simulator zurückgelegt hat. Ein Bauingenieur wird beim Bau einer großen Brücke selbstverständlich einen Simulator zurate ziehen, wenn es um die Frage geht, was denn wohl geschehen würde, wäre die Brücke einem Erdbeben ausgesetzt. Und wir alle begrüßen die Tatsache, dass bei der Lufthansa die Piloten erst einmal in einem Simulator ausgebildet werden, bevor Sie ein reales Flugzeug fliegen dürfen. Warum aber kommt das Mittel Simulation in der IT-Branche nicht zum Einsatz – ausgerechnet in der Branche, die für die ganzen anderen Industriezweige die entsprechenden Simulatoren entwickelt? In den meisten Fällen ist den Fachanwendern und den Softwareentwicklern überhaupt nicht klar, worin die Vorzüge einer Simulation von Geschäftsprozessen bestehen; dabei liegen diese eigentlich klar auf der Hand:

1. Mit einem Simulator lassen sich die logische Korrektheit und die Vollständigkeit eines Prozesses nachweisen. Voraussetzung hier ist, dass das verwendete Modellierungswerkzeug auf einem formalen Beschreibungsverfahren basiert, welches eine solche Prüfung zulässt. Die Sicherstellung der Vollständigkeit von Prozessen ist in der Praxis mithin eines der größten Probleme. Nicht selten verursachen Prozesszweige, die in der Analyse vergessen werden, erhebliche Mehraufwände in der Implementierung. In der Regel denken Fachanwender nämlich zuerst an den Standardprozess. Ausnahmen oder Sonderfälle werden gerne vergessen. Zumeist, weil diese durch die geringe Auftrittswahrscheinlichkeit unterschätzt werden. Ein Simulator kann hier den Modellierer dabei unterstützen, Unvollständigkeiten in einem Prozessmodell bereits in der Analyse aufzudecken.
2. Mit der Simulation lässt sich das Verhalten von SLA (Service Level Agreement)-relevante Gütekriterien des Prozesses prognostizieren. Es können Aussagen über Durchlaufzeiten, Durchsatz, Wartezeiten, kritische Pfade, Ressourcenengpässe und dergleichen mehr gemacht werden – und das bereits, bevor die erste Zeile Code programmiert wurde, und nicht erst, wenn sich das System bereits in Produktion befindet und sämtliche SLA-Parameter verletzt werden. Insbesondere bei SOA-Projekten hat sich gezeigt, dass die SLA-relevanten Anforderungen an eine Anwendung in der Analyse meistens unterschätzt oder gar nicht bewertet werden, mit dem Ergebnis, dass in der Produktionsumgebung festgestellt wird, dass

die implementierte Lösung gar nicht in der Lage ist, die geforderten Durchsatzeigenschaften zu erfüllen. Mit einer Simulation wäre dieser Umstand leicht im Vorfeld zu identifizieren gewesen.

Betrachtet man die auf dem Markt verfügbaren BPM-Werkzeuge, dann stellt man sehr schnell fest, dass nur sehr wenige Hersteller wirklich brauchbare Simulatoren im Angebot haben. Falls für ein BPM-Projekt Simulation zum Einsatz kommen soll, dann sollte dieses Thema in der Produktauswahl eine große Rolle spielen. Bei manchen Werkzeugen wird nämlich fälschlicherweise ein Animator als Simulator angeboten. Der Unterschied besteht hier darin, dass ein Animator zwar in der Lage ist, die Dynamik eines Prozesses durch Animationen darzustellen, dass aber im Unterschied zu einem echten Simulator keine tiefgreifenden Auswertungen der Simulationsläufe möglich sind. Ferner wird von einem guten Simulator erwartet, dass Schwachstellen in den Prozessen automatisch erkannt und entsprechende Hinweise gegeben werden, wie die Prozesse verbessert werden können.

Schließlich ist sicherzustellen, dass der infrage kommende Simulator nicht nur in der Lage ist, „Hello World“-Prozesse zu simulieren. Dies ist insbesondere dann wichtig, wenn nicht nur die logische Korrektheit eines Prozesses nachgewiesen werden soll, sondern wenn es darum geht, über einen langen Simulationszeitraum (in bestimmten Fällen ist es notwendig, Geschäftsprozesse über mehrere Monate hinweg zu simulieren) die relevanten Leistungskriterien eines Geschäftsprozesses zu untersuchen.

Kennzahlenbasiertes Monitoring

Ist die BPM-gestützte Implementierung eines Prozesses abgeschlossen, und befindet sich der Prozess in der Produktionsumgebung, dann ist es überaus wichtig, über ein kennzahlenbasiertes Monitoring das reale Verhalten der Prozesse in Echtzeit zu überwachen, um somit die Lücke zwischen Simulation und realer Welt zu schließen. Dabei ist zwischen dem rein technischen und dem fachlichen Monitoring zu unterscheiden. Das technische Monitoring dient in erster Linie dem Prozessbetreiber dazu, die Einhaltung relevanter SLA-Eigenschaften sicherzustellen. Mit BAM (Business Activity Monitoring) und dem Enterprise Manager stellt Oracle hier die geeigneten Werkzeuge zur Verfügung.

Weitaus wichtiger ist aber das fachliche Monitoring. Hier werden fachliche Kennzahlen überwacht, die typischerweise in sogenannten Balanced Scorecards oder Management Performance Cockpits den Fachanwendern zur Verfügung gestellt werden. Diese Kennzahlen müssen bereits in der Analysephase Bestandteil der Untersuchungen sein. Ist dies nicht der Fall, dann wird nach der Implementierung zwar das System in Betrieb genommen, aber die Fachanwender und das Management befinden sich, was die Bewertung der laufenden Prozesse im Sinne des Business Performance Managements angeht, im Blindflug. Kritische Zustände, Engpässe und sich langsam aufbauende Probleme können nicht erkannt werden. Bei der Dokumentation von Kennzahlen in der Analysephase ist darauf zu achten, dass die Herkunft der Messwerte, die den Kennzahlen zugrunde liegen, klar definiert wird, und dass über Kennzahlenmodelle einzelne Kennzahlen über Hierarchiebildung und Aggregation miteinander verknüpft werden können. Anschließend müssen diese Kennzahlenhierarchien in den entsprechenden Monitoring-Werkzeugen abgebildet werden. Aktuell sind leider keine BPM-Werkzeuge verfügbar, die eine automatische Übernahme von Kennzahlenhierarchien in Monitoring-Werkzeuge erlauben, was ein durchaus lohnendes Betätigungsfeld für die Hersteller von BPM-Werkzeugen wäre.

Hat man die Leistungsindikatoren in der Analyse vollständig als Teil der Prozessmodelle erfasst und anschließend in einem Monitoring-Werkzeug implementiert, so besteht die Möglichkeit, die Informationen über das reale Verhalten dieser Kennzahlen in einem Verfeinerungsschritt wieder in die Analysemodelle einfließen zu lassen. In diesem Zusammenhang wird in der BPM-Community gerne von closed-loop-Monitoring gesprochen, wobei auch hier alle Tool-Hersteller noch weit von einer brauchbaren Umsetzung entfernt sind. Dieses Zurückspielen der Informationen über Kennzahlen auf die analytische Ebene hilft eminent bei der Verfeinerung der analytischen Modelle und erlaubt somit

eine noch bessere Anpassung der Prozessmodelle an die Gegebenheiten der tatsächlich gelebten Prozesse. Ferner handelt es sich dabei um eine Grundvoraussetzung, wenn es darum geht, auch in Zukunft auf Veränderungen in der Prozesslandschaft schnell und flexibel reagieren zu können. Und genau dieser durchgängigen und konsistenten Vorgehensweise bei der kontinuierlichen Optimierung von Geschäftsprozessen, im Sinne eines Prozessregelsystems, ist nach Ansicht des Autors, im Gegensatz zu einem rudimentären Roundtrip-Verfahren zwischen Analyse- und Design- bzw. Implementierungswerkzeugen ganz klar der Vorzug zu geben.

Kontaktadresse:

Dipl.-Inform. Sebastian Graf
PROMATIS software GmbH
Pforzheimer Strasse 160
D-76275 Ettlingen

Telefon: +49 (0) 7243-2179-0
Fax: +49 (0) 7243-2179-99
E-Mail sebastian.graf@promatis.de
sebastian.graf@doag.org
Internet: <http://www.promatis.de>
<http://www.horus.biz>