

Offshore Outsourcing

Kontrolle behalten!

Die Potenziale beim Offshore Outsourcing von IT-Leistungen sind hoch und werden es auf Jahre hinaus bleiben. Mit zunehmender Reife des Offshore-Marktes haben sich mittlerweile Mechanismen und Instrumente herausgebildet die Offshore-Risiken beherrschbar machen.



Von herausragender Bedeutung für den Offshoring-Erfolg ist neben der Partnerwahl, die Vereinbarung sinnvoller SLAs und eines flexiblen Preismodell, welches die oft gegenläufigen Interessen beider Partner unter einen Hut bringt.

Produktivitätsunterschiede zwischen Onshore- und Offshore-Modellen sind dramatisch. Die Produktivität von Onshore-Modelle entspricht in etwa dem Branchendurchschnitt (siehe Bild 1, etwa 500 € pro Function Point). Offshore-Modelle liegen dagegen meist deutlich besser im Bereich der Best-In-Class-Produktivität (z.B. 180 € pro Function Point). Doch wie kann die Produktivität bei Offshore-Verlagerungen auch vertraglich abgesichert werden? Der Schlüssel dazu ist unter anderem ein geeignetes Preismodell.

Instrument 1: Preismodell (Kontrolle über Kosten)

Wichtiges Erfolgskriterium für jede Outsourcing-Transaktion ist ein vertraglich abgesicherter Business-Case. Im „klassischen“ Outsourcing wird meist ausgehend vom Ist-Aufwand ein klarer Zielkostenkanal als „Deckel“ für die Kostenentwicklung künftiger Jahre vereinbart, um Kosteneinsparungen und Produktivität des Providers vertraglich abzusichern.

Beim Offshore-Outsourcing ist gerade das besonders schwierig, weil vor allem Anwendungsentwicklungsleistungen in Offshore-Länder verlagert werden. Diese Leistungen bedingen ein hohes Maß an Flexibilität und lassen sich kaum über eine Laufzeit von mehreren Jahren per Festpreis „deckeln“, denn der benötigte Leistungsumfang ist nicht

statisch, sondern ergibt sich kurzfristig aus neuen bzw. veränderten Geschäftsanforderungen, die beispielsweise neu zu programmierende Funktionalitäten erforderlich machen. Bei vielen Offshore-Transaktionen werden daher Anwendungsentwicklungsleistungen auf Basis „Time & Material“ immer noch nach Aufwand vergütet. Die Anwendungswartung wird zumeist auf Festpreisbasis oder ebenfalls nach Aufwand vergütet.

Hauptproblem dieser Festpreis- oder „Time&Material“-Modelle ist, dass sich die Vergütung nicht an der Produktivität des Providers orientiert. Daher kann der Auftraggeber die Produktivität seines Providers nur unzureichend steuern, geschweige denn vertraglich vereinbaren. In diesen Modellen orientiert sich die Vergütung nicht an dem Leistungsoutput. Der Provider bekommt dieselbe Vergütung, ob er



Produktivität der Anwendungsentwicklung

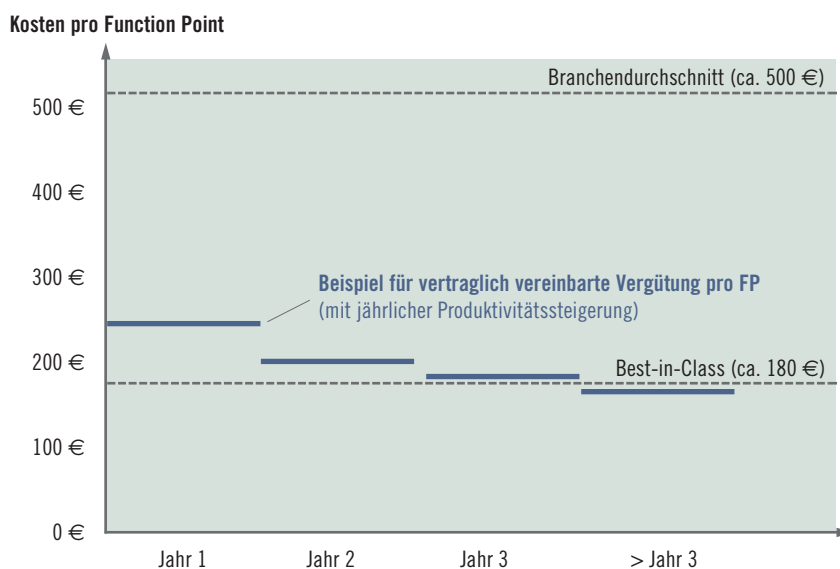


Bild 1: Preismodell mit garantierter Produktivität.

nun gut oder schlecht arbeitet. Die hohen Erwartungen an Offshore-Projekte in Bezug auf Produktivitätssteigerungen und Einsparungen werden dann nur teilweise erfüllt. Mögliche Einsparpotenziale werden bei beiden Modellen oft nicht voll ausgeschöpft.

Diese Problem lösen Preismodelle die auf Function Points basieren (Erläuterung der Function Point Methode siehe Bild 2). In diesen Modellen bezahlt der Kunde einen Preis für den Output des Providers. Bezahlt werden implementierte Function Points als Maß für den Umfang beziehungsweise die Größe einer implementierten Funktionalität. Eine solche leistungsgerechte Vergütung stellt die Produktivität des Providers sicher und lässt sich ermitteln mit Hilfe der Kennzahl „Implementierungskosten pro Function Point in Euro.“

Bei einem solchen Preismodell kann die Produktivität des Providers vertraglich abgesichert werden, indem etwa eine Vergütung von „190 € pro Function Point“ für Anwendungsentwicklungsleistungen vereinbart wird. Bei Verträgen mit längeren Laufzeiten werden auf diese Weise Produktivitätssteigerungen zum Beispiel jährlich festgeschrieben (Beispiel siehe Bild 1). Die Wirtschaftlichkeit der Outsourcing-Transaktion lässt sich auf diese Weise optimal absichern.

Aus Kundensicht attraktiv ist auch ein Function-Point-Deckel. Der Provider rechnet dabei zwar mit einem definierten Stundensatz nach Aufwand ab. Dabei wird eine Mindestproduktivität in Form eines Deckels (etwa bei maximal 190 € pro Function Point) vertraglich festgeschrieben. Bei Überschreitungen erhält der Provider keine zusätzliche Vergütung für seine eingesetzten Stunden, bei Unterschreitungen wird die Einsparungen an den Kunden weitergegeben. Der Kunde profitiert in jedem Fall. Gegebenenfalls kann bei Unterschreitungen auch ein Bonus an den Provider gezahlt werden, um diesem einen Anreiz für Einsparungen zu bieten.

Function Point-Modelle sind flexibel, da neue und vorher nicht bekannte Anforderungen gerecht vergütet werden. Das oft mühsame Feilschen um die etwaige Anpassung eines starren Festpreisdeckels entfällt.

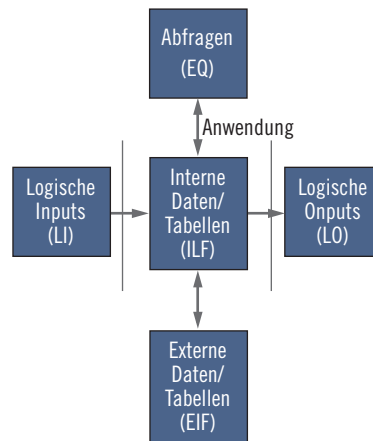
Ein weiterer Vorteil ist die einfache Benchmarkfähigkeit eines solchen

Die Function Point Methode

Die Function Point Methode dient zur Messung von Größe und Umfang eines Anwendungssystems basierend auf den gespeicherten Datenobjekten, seinen Schnittstellen und den Transaktionen/Prozessen zur Verwaltung der Daten. Gezählt werden:

- Daten/Tabellen, die von der Anwendung selbst erstellt und verwaltet werden (sogenannte Internal Logical Files/ILF).
- Daten/Tabellen anderer Anwendungen, auf die die Anwendung über Schnittstellen zugreift (sogenannte External Interface Files(EIF).
- Prozesse/Transaktionen, die Daten hinzufügen, verändern, löschen oder abfragen (dabei werden sogenannte external Inputs/EI, external outputs/EO, und external Queries/EQ unterschieden).

Diese Objekte werden nach einem standardisierten Algorithmus gewichtet und summiert, so daß sich ein eindeutiger Function Point Wert für die Größe des gezählten Anwendungssystems ergibt.



Die Function Point Methode ist die weitverbreitetste Methode mit der sich die Größe einer Software erfolgreich messen läßt, um Produktivität, Entwicklungszeiten, und -aufwand und Entwicklungsqualität effektiv abschätzen und kontrollieren zu können. Andere Methoden spielen im Markt kaum eine Rolle.

Bekanntlich kann man nur managen, was man auch messen kann. Dies gilt besonders im Bereich des Outsourcing, wo Messmethoden zur Kontrolle von Qualität und Produktivität die Basis für eine erfolgreiche Produktivitätssteuerung sind. Daher spielt die Function Point Methode insbesondere im Bereich des Offshorings eine Schlüsselrolle.

The Internal Function Point User's Group (IFPUG. at www.ifpug.org) ist als Standardisierungs-Gremium verantwortlich für die Erhaltung des Function Point Standards. Durch die Standardisierung der Methode kommt eine Zählung immer zum gleichen Ergebnis, unabhängig davon, welche Person die Zählung durchführt. Die Abweichungen bei unabhängig voneinander durchgeführten Zählungen sind sehr gering (i.d.R. < 5 %).

Bild 2: Die Function Point-Methode.

Preismodells. Marktvergleiche für Anwendungsentwicklung oder Wartung basieren fast immer auf Function Point-Metriken. Ein Function-Point-basiertes Preismodell ermöglicht auch die hohe Vergleichbarkeit von verschiedenen Bietangeboten im Auswahlprozess.

Für Wartung und Anwendungsentwicklung wird jeweils ein separater Preis benötigt, denn natürlich hat eine „gewarteter“ Function Point einen niedrigeren Preis als ein neu entwickelter Function Point. Vertraglich vereinbart werden daher beispielsweise 150 pro „gewartetem“ Function Point und 190 pro neu entwickeltem Function Point.

Herausforderungen der Function Point-Methode

Nachteil von function-point-basierten Abrechnungsmodellen ist der zusätzliche Aufwand für die Zählung der Function-Points. Der Aufwand wird jedoch oft überschätzt und bleibt bei pragmatischer Anwendung der Methode moderat.

Zu Beginn eines Offshorings sind die Function Points einer bestehenden Anwendung ggf. noch nicht bekannt, werden aber benötigt, um die absolute Höhe der Wartungskosten kalkulieren zu können. In diesem Fall können die Functions Points grob über das sogenannte „Backfiring“-Verfahren aus der Anzahl der „Lines of Code“ rückgerechnet werden. Entsprechende Um-

rechnungsfaktoren sind für verschiedene Entwicklungssprachen verfügbar. Jedoch ist das „Backfiring“ vergleichsweise ungenau (10% bis 20% Abweichung). Daher kann ggf. im Rahmen der Transitionsphase der Offshoring-Transaktion eine exakte Zählung stattfinden. Entsprechende vertragliche Klauseln, die die Auswirkung der Zählung auf die Vergütung festlegen, sind dann notwendig.

Wer auf ein Function-Point-basiertes Preismodell verzichten will, kann auch ein „Win-& Risk-Sharing-Modell“ mit dem Provider vereinbaren. Die Verrechnung erfolgt nach Aufwand zu einem definierten Tagesatz. Verbraucht der Provider mehr Tage als budgetiert, so gewährt er dem Kunden eine Rabatt von z.B. 30%, schafft er es unter Budget zu bleiben, so erhält er für jeden eingesparten Tag 30% des Tagesatzes als Bonuszahlung. Das Modell stellt für beide Seiten finanzielle Anreize für Einsparungen sicher und bietet den meisten Kunden deutlich mehr Vorteile als Festpreismodelle, oder rein aufwandsbezogene Modelle.

Überschreitet der Provider das Budget, wird er unprofitabel (wg. der -30% Maluszahlung). Wie beim Festpreismodell wird er folglich versuchen das Budget einzuhalten. Aber der Hauptnachteil des Festpreismodells entfällt, denn beim „Win- & Risk-Sha-

ring-Modell“ wird die Einsparung an den Kunden weitergegeben. Der Provider wird also versuchen das Budget zu unterschreiten da er in diesem Fall die Bonuszahlung erhält. Zudem kennt der Kunde am Ende der Vertragslaufzeit die benötigten Wartungstage genau. Sofern die Wartungstage unter der ursprünglichen Kalkulation liegen, was häufig der Fall ist – kann er am Ende der Laufzeit den Vertrag entsprechend anpassen. Bei Festpreisvereinbarungen ist diese Transparenz oft nicht gegeben.

Auch der Hauptnachteil des aufwandsbezogenen Modell, die oft ausufernde Kosten, entfallen, denn beim „Win- & Risk-Sharing-Modell“ hat der Provider anders als beim aufwandsbezogenen Modell aufgrund der 30%-igen Maluszahlung einen starken Anreiz zur Budgeteinhaltung. Somit verbindet das „Win- & Risk-Sharing-Modell“ die Sicherheit eines Festpreismodells mit der Flexibilität



Win- & Risk-Sharing Preismodell

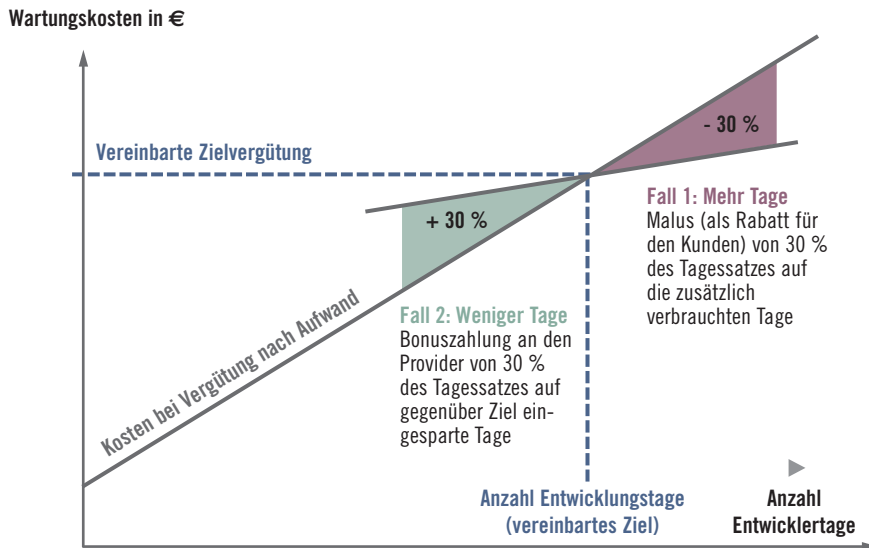


Bild 3: Bei der Win- & Risk-Sharing-Methode besteht für den Dienstleister ein Anreiz, effektiv zu arbeiten.

und den Kostenvorteilen eines aufwandbasierten Preismodells. Zugleich entschärft es die Nachteile beider Modelle. Geringe Entwicklungs- und langfristig sinkende Wartungskosten sind in beiderseitigem Interesse.

Dieses Modell ist besonders für die Wartung geeignet, denn um die Höhe der jährlichen Gesamtvergütung für die Wartung ermitteln zu können, müssen beim Function Point Modell die Function Points der bestehenden Anwendung bekannt sein. Dies ist aber zu Beginn einer Offshoringtransaktion oft nicht der Fall und eine Zählung ist aus Zeitgründen nicht möglich.

Instrument 2: SLAs (Kontrolle über die Qualität)

Neben dem richtigen Preismodell ist auch die vertragliche Vereinbarung von geeigneten SLAs die Basis für eine erfolgreiche Offshoring. Da vor allem

die Fremdvergabe von Anwendungsentwicklung und Wartung am weitesten verbreitet sind, fokussiert dieser Artikel auf Service Level Agreements für diese beiden Leistungen.

SLAs - Anwendungsentwicklung

Die Qualität der Anwendungsentwicklung lässt sich vor allem anhand der Fehler im ausgelieferten Source Code messen. Allerdings funktioniert auch dies nur, wenn man die Größe der ausgelieferten Softwarekomponente als Bezugsgröße kennt. Auch hier hat daher die Nutzung der Function Point Metrik entscheidende Vorteile. Sinkt die Qualität unter X Fehler pro Function Point so kann man im Vertrag Pönale vereinbaren. Die Lieferqualität des Providers kann auf diese Weise effektiv vereinbart und gesteuert werden.

Ebenfalls wichtig sind u.a. die zeitliche Liefertreue und Fehlerbehebungszeiten, die festlegen wie schnell der Lie-

Übersatz der 2. Marginalie

der haben stark gegenläufige kommerzielle Interessen, die nur über vertragliche Regelungen koordiniert werden können.

Fehlen geeignete Preismodelle oder sinnvolle SLA-Kennzahlen, so endet die Outsourcing-Transaktion zumeist schnell in der Sackgasse. Entweder profitiert der Provider einseitig oder die Offshoring-Transaktion scheitert gar, mit hohen Kosten für Provider und Kunden.

Anwendungsentwicklung – SLA Beispiele

Service Level	Vereinbarer Wert	Erläuterung																	
Anzahl Fehler pro Function Point	< x	Es wird die Summe über die Fehler aller Fehlerklassen gebildet. Anzahl Fehler im Mittel für alle Software-Auslieferungen (nach Systemtest) eines Monats																	
Pünktliche Auslieferung (Größere Erweiterungen)	x %	Prozentsatz der Lieferungen (Entwicklungsleistungen mit einer Größe > 10 Personentage) die pünktlich																	
Pünktliche Auslieferung (Kleinere Erweiterungen)	x %	<table border="1"> <thead> <tr> <th>Service Level</th> <th>Zeiten¹</th> <th>Spezifikation²</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Fehlerklasse E1</td> <td>Reaktionszeit: x Stunden</td> <td rowspan="3">Fehlerklasse E 1 beschreibt Fehler in Liefereinheiten (der Größe > 10 Function Points) mit folgender Auswirkung: – der Test einer gesamten Liefereinheit oder wesentlicher Teile wird verhindert, oder – die Liefereinheit ist instabil, so dass es immer wieder zu Programmabstürzen kommt, bis der Fehler behoben ist</td> </tr> <tr> <td>Wiederherstellungszeit: x Stunden</td> </tr> <tr> <td>Lösungszeit: x Werktage</td> </tr> <tr> <td rowspan="2">Fehlerklasse E2</td> <td>Reaktionszeit: x Stunden (avg. p.m.) und max. x Stunden</td> <td rowspan="2">Fehlerklasse E 2 beschreibt Fehler mit der folgenden Auswirkung: – der Test von Liefereinheiten (der Größe > 10 Function Points) ist stark eingeschränkt, oder – der Fehler verhindert den Test einer gesamten Liefereinheit (bei Größen < 10 Function Points und kleinere Erweiterungen), oder – der Fehler kann nur mit signifikantem Aufwand oder signifikantem Zeitverzug kompensiert werden.</td> </tr> <tr> <td>Lösungszeit: x Stunden (avg. p.m.) und max. x Werktage</td> </tr> <tr> <td>Fehlerklasse E3</td> <td>Reaktionszeit: x Stunden (avg. p.m.) und max. x Werktage</td> <td rowspan="2">Fehlerklasse E 3 umfasst alle sonstigen Fehler. Beispiele sind: – der Fehler dieser Klasse beeinträchtigt den Test, ist jedoch mit vertretbarem Aufwand kompensierbar – kosmetische Fehler, die nicht die Funktionsweise beeinträchtigen, sondern sich lediglich auf die optischer Erscheinung auswirken.</td> </tr> <tr> <td></td> <td>Lösungszeit: x Werktage (avg. p.m.) und max. x Werktage</td> </tr> </tbody> </table>	Service Level	Zeiten ¹	Spezifikation ²	Fehlerklasse E1	Reaktionszeit: x Stunden	Fehlerklasse E 1 beschreibt Fehler in Liefereinheiten (der Größe > 10 Function Points) mit folgender Auswirkung: – der Test einer gesamten Liefereinheit oder wesentlicher Teile wird verhindert, oder – die Liefereinheit ist instabil, so dass es immer wieder zu Programmabstürzen kommt, bis der Fehler behoben ist	Wiederherstellungszeit: x Stunden	Lösungszeit: x Werktage	Fehlerklasse E2	Reaktionszeit: x Stunden (avg. p.m.) und max. x Stunden	Fehlerklasse E 2 beschreibt Fehler mit der folgenden Auswirkung: – der Test von Liefereinheiten (der Größe > 10 Function Points) ist stark eingeschränkt, oder – der Fehler verhindert den Test einer gesamten Liefereinheit (bei Größen < 10 Function Points und kleinere Erweiterungen), oder – der Fehler kann nur mit signifikantem Aufwand oder signifikantem Zeitverzug kompensiert werden.	Lösungszeit: x Stunden (avg. p.m.) und max. x Werktage	Fehlerklasse E3	Reaktionszeit: x Stunden (avg. p.m.) und max. x Werktage	Fehlerklasse E 3 umfasst alle sonstigen Fehler. Beispiele sind: – der Fehler dieser Klasse beeinträchtigt den Test, ist jedoch mit vertretbarem Aufwand kompensierbar – kosmetische Fehler, die nicht die Funktionsweise beeinträchtigen, sondern sich lediglich auf die optischer Erscheinung auswirken.		Lösungszeit: x Werktage (avg. p.m.) und max. x Werktage
Service Level	Zeiten ¹		Spezifikation ²																
Fehlerklasse E1	Reaktionszeit: x Stunden		Fehlerklasse E 1 beschreibt Fehler in Liefereinheiten (der Größe > 10 Function Points) mit folgender Auswirkung: – der Test einer gesamten Liefereinheit oder wesentlicher Teile wird verhindert, oder – die Liefereinheit ist instabil, so dass es immer wieder zu Programmabstürzen kommt, bis der Fehler behoben ist																
	Wiederherstellungszeit: x Stunden																		
	Lösungszeit: x Werktage																		
Fehlerklasse E2	Reaktionszeit: x Stunden (avg. p.m.) und max. x Stunden	Fehlerklasse E 2 beschreibt Fehler mit der folgenden Auswirkung: – der Test von Liefereinheiten (der Größe > 10 Function Points) ist stark eingeschränkt, oder – der Fehler verhindert den Test einer gesamten Liefereinheit (bei Größen < 10 Function Points und kleinere Erweiterungen), oder – der Fehler kann nur mit signifikantem Aufwand oder signifikantem Zeitverzug kompensiert werden.																	
	Lösungszeit: x Stunden (avg. p.m.) und max. x Werktage																		
Fehlerklasse E3	Reaktionszeit: x Stunden (avg. p.m.) und max. x Werktage	Fehlerklasse E 3 umfasst alle sonstigen Fehler. Beispiele sind: – der Fehler dieser Klasse beeinträchtigt den Test, ist jedoch mit vertretbarem Aufwand kompensierbar – kosmetische Fehler, die nicht die Funktionsweise beeinträchtigen, sondern sich lediglich auf die optischer Erscheinung auswirken.																	
	Lösungszeit: x Werktage (avg. p.m.) und max. x Werktage																		
First Time Right	> x % p.m.																		
Antwortzeiten der ausgelieferten Anwendung	x % der Transaktionen < 1s																		

1) Sofern nicht anders gekennzeichnet verstehen sich die Werte als maximale Grenze für 100 % aller Incidents; Ausnahme sind Zeiten, die wie folgt gekennzeichnet sind:
avg. p.m. = average per month, d.h. in diesem Fall wird die Kennzahl über alle Incidents im Monatsdurchschnitt berechnet.
Sofern nicht abweichend gekennzeichnet werden die in Stunden angegebenen Zeiten durch das Ende der Servicezeit unterbrochen und laufen erst mit dem Beginn des nächsten serviceintervalls weiter. Dies gilt nicht bei Zeitangaben in Werktagen.
2) Der Auftraggeber ist berechtigt, die erste Klassifikation durchzuführen. Der Auftragnehmer ist berechtigt, Klassifikationen in Abstimmung mit dem Auftraggeber zu ändern, sofern die Klassifizierung durch den Auftragnehmer von den hier dargestellten Definitionen abweicht. In Zweifelsfällen entscheidet der Auftraggeber nach Eskalation in die gemeinsamen Gremien über die Fehlerklasse.

Bild 4: SLAs in der Anwendungsentwicklung.

ferant die ausgelieferten Fehler behebt (Beispiele für SLAs siehe Bild 4). Dadurch lässt sich der Test und Koordinationsaufwand im eigenen Hause reduzieren und eine pünktliche Produktivsetzung sicherstellen.

SLAs - Wartung

Leistungsscheine für die Wartung sollten die Servicezeiten regeln, in denen die Annahme von Fehlern möglich ist. Dies ist nicht immer problemlos. Da die Programmierer vieler Offshore-Provider in anderen Zeitzonen arbeiten. Daher besteht Regelungsbedarf. Viele Offshore-Provider haben jedoch Brückenköpfe im Heimatland des Auftraggebers. Die Brückenköpfe besitzen dieselben Arbeitszeiten wie der Auftraggeber können jedoch aufgrund der höheren Lohnkosten teurer sein. Auch eine 7x24h-Wartung lässt sich über eine sinnvoll geregelte Arbeitsteiligkeit zwischen Brückenkopf und Offshore-Entwicklungszentrum gut abbilden.

Ein guter Leistungsschein für die Wartung sollte darüber hinaus unbedingt die Reaktions-, Wiederherstellungs- und Fehlerbeseitigungszeiten detailliert regeln (siehe Bild 3). Diese müssen deutlich kürzer gewählt werden, als diejenigen im Entwicklungsleistungsschein, da der Provider bei schwerwiegenden Fehlern im Produktivsystem

unbedingt schnell reagieren muss. Weitere sinnvolle SLA-Kennzahlen sind u.a. „Anzahl der zurückgewiesenen Bug-Fixes“ und die „Mittlere Zeit zwischen zwei Fehlern (nach Fehlerklasse/Priorität)“.

Weitere Erfolgsfaktoren

Organisatorische Veränderungen sind unbedingt notwendig, um mit dem Offshoring erfolgreich zu sein. Gute Offshoring-Provider bringen geeignete Verfahren mit. Allerdings ist in diesem Fall der Veränderungswille des Auftraggebers gefragt. Wer als Auftraggeber bereit ist, seine Prozesse und Verfahren an die des Providers anzupassen ist in der Regel erfolgreicher.

Die Verlagerung von Entwicklungs- und Wartungsressourcen nach Offshore macht professionelle und formale Prozesse unbedingt erforderlich. Beispielsweise müssen Anforderungsspezifikation, Dokumentation und Abnahmeverfahren detailliert geregelt werden. Die meisten Offshore-Provider besitzen zusätzlich zu Ihren Offshore-Entwicklungszentren auch Vor-Ort-Organisationen als „Brückenköpfe“ im Heimatland des Auftraggebers. Die Arbeitsteiligkeit zwischen Kunden, Brückenkopf und Offshore-Organisation muss sauber geklärt werden. Bestimmte Tätigkeiten

lassen sich trotz Lohnkostenvorteile nicht oder nicht vollständig nach Offshore verlagern. Dazu gehören Schlüssel-tätigkeiten wie die Anforderungsaufnahme, Projektmanagement, Controlling und Akzeptanztests. Diese werden daher häufig Onsite über die Brückenkopf-Organisationen abgebildet.

Das sie über gute Methoden und Verfahren verfügen, lässt sich an den Zertifizierungen eremessen, in denen vor allem die Inder den europäischen IT-Organisationen weit voraus sind. Beim Auswahlprozess sollte daher abgefragt werden, ob die konkrete Delivery-Unit des Providers, die den Auftraggeber künftig beliefern soll, eine entsprechende Zertifizierung besitzt. Wichtige Qualitätszertifikate in diesem Zusammenhang sind: CMM Levelx, CMMI Levelx und ggf. auch ISO900x.

Ein weiterer wichtiger Aspekt ist die Schaffung und vertragliche Vereinba-



zung von Infrastrukturvoraussetzungen. Da die meisten Offshore-Transaktionen Anwendungsentwicklung und -wartung zum Inhalt haben, werden Regelungen zur Infrastruktur in vielen Vertragswerken vergessen oder sträflich vernachlässigt. Vertraglich geregelt werden sollten u.a. folgende Punkte:

- Garantiert der Auftragnehmer die WAN-Anbindung an die Offshore-Entwicklungszentrum? Wo ist der Leistungsübergabepunkt? Häufiges Problem sind instabile Leistungen und zu geringe Bandbreiten. Dies ist selbst im besser entwickelten Indien ein häufiges Problem und kann ein hohes Risiko für Aufwand und pünktliche Auslieferung darstellen.
- Wo stehen die Test- und Entwicklungsserver? Wer hat Zugriff? Wie sind die Servicezeiten und Service Level dieser Server?
- Wie ist das Verfahren zur Softwareauslieferung? Welche Toolunterstützung und Infrastrukturvoraussetzungen sind dazu notwendig?
- Welcher der beiden Partner trägt die Kosten für Lizenzen, WAN-Anbindung, Entwicklungsserver und sonstige Infrastrukturvoraussetzungen?

In jedem Fall sollte der Betrieb der Test- und Entwicklungsserver mit Service Levels abgesichert werden. Dazu gehören Betriebs- und Servicezeiten (insbesondere wenn verschiedene Zeitzeonen abgedeckt werden müssen), Verfügbarkeit, Max. Ausfalldauer und -häufigkeit. Für die WAN-Anbindung an das Offshore-Entwicklungszentrum sollten mindestens Verfügbarkeiten und Latenz-/Antwortzeiten definiert sein. Redundante Auslegung der Leitungen sind meist erforderlich.

Ein Notfallkonzept sollte zudem festlegen, wie die Daten gesichert und an verschiedene Standorte verlagert werden und wie und wie schnell die

Wiederaufnahme des Betriebs erfolgt, wenn im Ernstfall das Entwicklungszentrum nicht zur Verfügung bzw. zerstört ist. Dies ist beim Offshoring besonders wichtig, da das Risiko von Naturkatastrophen beispielsweise in Indien höher ist.

Darüber hinaus sind einige juristische und datenschutzrechtliche Aspekte zu beachten. Es gibt Mustervorlagen von EU-Gremien, die man als Anlage an den Rahmenvertrag mit aufnehmen kann, um sicherzustellen, dass er rechtlichen Anforderungen genügt (etwa die Standardvertragsklauseln für die Übermittlung personenbezogener Daten an Auftragsverarbeiter in Drittländern nach der Richtlinie 95/46/EG).

Vertragsgestaltung

Kunden sollten in jedem Fall aktiv Rahmenvertrag, Leistungsscheine, Preismodelle und sonstigen Regelungen mitgestalten. Am besten fährt, wer dem Provider fertige Vorlagen vorgibt. Wer nur reaktiv auf Providervorschläge eingeht, verschenkt viele Vorteile. Stammen sowohl Rahmenvertrag als auch Leistungsscheine vom Provider, dann sind viele Formulierungen enthalten, die den Provider einseitig begünstigen und absichern. Solche auftraggeber-unfreundlichen Klauseln sind oft subtil, schwer zu erkennen und müssen mühsam „raus“-verhandelt werden. Dies resultiert in hohen Zeitverlusten und einer deutlichen Schwächung der Verhandlungsposition des Auftraggebers. Am gefährlichsten an den Provider-Vertragsvorlagen sind die fast immer fehlenden Regelungen zur Absicherung des Auftraggebers.

Dennoch baut die Mehrzahl der Unternehmen ihre Vereinbarungen auf den Provider-Angeboten und Vertragsvorschlägen auf. Wen wundert's? Schließlich hat der Provider im Gegensatz zu seinem Kunden mehr Erfahrungen mit

Outsourcing-Verträgen während der Kunde meist auf keine oder nur vereinzelte Erfahrungswerte zurückgreifen kann. So mancher ist daher dankbar für Vertragsvorschläge vom Provider und allzu schnell bereit diesen zuzustimmen, mit signifikanten Nachteilen.

Fazit

Aufgrund der dramatischen Lohnkostenvorteile lohnen sich Offshoring-Transaktionen für viele Unternehmen. Dennoch werden bewährte Instrumente nicht ausreichend eingesetzt. Häufig schöpfen daher viele Kunden die Potenziale des Offshorings nicht voll aus. So sind beispielsweise reine Festpreis- oder aufwandsbezogenen Preismodelle fast immer suboptimal für den Kunden. Kommen Sie zum Einsatz, profitiert nicht selten der Offshoring-Provider überdurchschnittlich, während die Erwartungen des Auftraggebers nicht in voller Höhe erfüllt werden.

Eigentlich ist dies unnötig. Denn wie zuvor aufgezeigt, existieren genügend Instrumente wie ein geeignetes Preismodell und SLAs, um Offshoring-Projekte mit vergleichsweise geringem Risiko zum Erfolg zu führen. Unternehmen, die organisatorische Flexibilität beweisen und sich gut auf die Transaktion vorbereiten (zum Beispiel in dem Sie vor dem Offshoring Function-Point-basierte Kennzahlen etablieren) können überdurchschnittlich profitieren.

Neben dem organisatorischen Veränderungs willen und der zugehörigen Flexibilität ist die vertragliche Vereinbarung von SLAs eine Haupterfolgskriterium, um die Kontrolle über die Lieferqualität nicht zu verlieren.

Besonders wichtig ist bereits die Abfrage von Produktivitätskennziffern und SLAs im Rahmen des Auswahlprozesses. Denn nur diese macht die tatsächliche Eignung des Partners transparent. Die Spreu lässt sich am besten vom Weizen trennen, wenn man darauf achtet, ob ein Provider bei der vertraglichen Zusicherung bestimmter SLAs oder Produktivitätskennzahlen anfängt zu zaudern. Provider, die nicht bereit sind professionelle SLAs oder Produktivitätskennzahlen vertraglich zuzusichern, sollten gemieden werden.

Carsten Glohr,
carsten.glohr@detecon.com

Rahmenverträge und Leistungsscheine vom Provider, sind in der Regel äußerst auftraggeber-unfreundlich. Am besten ist es daher, wenn man als Kunde dem Provider selbst fertige Vorlagen vorgibt. Ohne eine gute vertragliche Grundlage wird ein Offshoring sonst schnell zum Desaster, denn Kunde und Provi-
