



Ein SLA-basiertes Zusammenarbeitsmodell bei verteilten Teams

# Maximale Flexibilität und Kontrolle



©iStockfoto.com/Tijana87

Das Outsourcing von Softwareprojekten findet eine immer größere Verbreitung und erreicht zusehends höhere Ebenen der Wertschöpfungskette. Besonders in Offshoremodellen ist es jedoch ungleich anspruchsvoller, die auftretenden Herausforderungen erfolgreich zu bewältigen. Eine typische Maßnahme besteht in der Einführung strikter Kontrollen und sequenzieller Projektpläne, was wiederum viele Vorteile der hierzulande etablierten agilen Vorgehensmodelle negiert. Im Artikel wird ein praxiserprobtes Zusammenarbeitsmodell vorgestellt, das eine hohe Flexibilität erlaubt und zugleich eine zuverlässige Kontrolle des Projektfortschritts ermöglicht.

AUTOR: DR. ALAN ETLIN

Seit Anfang des Jahrtausends hat hierzulande eine rasanten Zunahme der Software-Offshoreentwicklung stattgefunden. Das zuerst wohl meistgenannte Argument dafür sind die tieferen Lohnkosten in typischen Offshoreländern, also eine taktische Kostensenkung im Projekt. Auch der hiesige Fachkräftemangel gilt als wichtige Motivation für Offshoreprojekte – es ist zum Teil schlicht nicht möglich, im lokalen Markt ausreichend gut qualifizierte Mitarbeiter für Softwareprojekte zu finden.

## VERSUCHUNG OUTSOURCING UND OFFSHOREOUTSOURCING

Zunehmend stehen jedoch neben diesen klassischen Argumenten differenziertere Motivationen für eine Offshoreentwicklung im Vordergrund. Anstelle des sprichwörtlichen Stopfens von Löchern, die durch den Fachkräftemangel verursacht werden, tritt der vielbeschworene „Global War for Talent“, also der weltweite Wettbewerb um die fähigsten Mitarbeiter, in den Vordergrund. Es geht nicht mehr primär darum, sich zusätzliche Kapazitäten für das eigene Vorhaben zu sichern, sondern

vielmehr darum, die Besten ihres Fachs für die eigene Sache zu gewinnen. Diese Topleute bewegen sich in einem zunehmend globalisierten Arbeitsmarkt, wodurch man automatisch in Konkurrenz zu global tätigen Arbeitgebern steht. Diese Entwicklung geht Hand in Hand mit der Evolution des Offshoreoutsourcings, das zusehends höhere Ebenen der Wertschöpfungskette erreicht. So ist es heute nicht länger erstaunlich, dass in der IT neben den ursprünglichen typischerweise simplen Programmieraufgaben auch Funktionen wie Softwaredesign, Architektur, Projektleitung oder gar Forschung und Entwicklung in Offshoreländer ausgelagert werden [1]. Unternehmen, die sich diese Entwicklung gezielt zunutze machen, können eine größere Flexibilität erreichen, indem beispielsweise die Organisation selbst sich auf ihre Kernkompetenzen fokussiert und ergänzende Tätigkeiten im Outsourcing erledigen lässt – oft auch Offshore. Insbesondere für international tätige Unternehmen kann ein derartiges Vorgehen durchaus zu einem signifikanten Erfolgsfaktor werden [2].

## DIE KEHRSEITE DER MEDAILLE

Die Erfahrung hat uns gelehrt, dass insbesondere das Offshoreoutsourcing neben den eben aufgeführten, verlockenden Versprechungen auch viele Tücken mit sich bringt. Nakatsu und Iacovu [3] haben in einer detaillierten Studie einen Vergleich der Risiken von Outsourcingsoftwareprojekten in lokalen und Offshoremodellen durchgeführt. Die Teilnehmer der Studie haben die signifikantesten Risiken von Offshoremodellen gemäß Tabelle 1 identifiziert.

**Mangelnde Unterstützung seitens des Topmanagements** bezieht sich auf die bekannte Tatsache, dass ein Projekt ohne adäquate Managementunterstützung zum Scheitern verurteilt ist, sei es, weil z. B. benötigte

Ressourcen abgezogen werden oder das Projekt in den Strudel politischer Spannungen gerät. Dieses Risiko hat sowohl für lokal durchgeführte als auch Offshoreprojekte eine große Bedeutung, wurde jedoch in der zitierten Studie für Offshoreprojekte als noch signifikanter eingeschätzt.

**Fehlkommunikation der ursprünglichen Anforderungen** unterstreicht die Bedeutung eines gemeinsamen Verständnisses aller Projektbeteiligten bezüglich der Projektanforderungen. Bei einem verteilten Projektteam wird dieses Risiko durch eingeschränkten direkten Kontakt der einzelnen Akteure vergrößert.

**Sprachbarrieren in der Projektkommunikation** stellen eine weitere bekannte Hürde im Offshorebusiness dar. Selbst bei einer vorhandenen gemeinsamen (Fremd-) Sprache entstehen Missverständnisse durch unterschiedliches Verständnis von Ausdrücken.

**Inadäquater Einbezug der Endnutzer** entsteht oft durch die komplexere Projektorganisation in Offshoreprojekten, die eine zusätzliche Distanz zwischen Endnutzern des Systems und anderen Projektbeteiligten erzeugt.

**Mangelnde Erfahrung in der Offshoreprojektleitung seitens des Auftraggebers** unterstreicht die Tatsache, dass die komplexeren Organisationen in Offshoreprojekten sowie die technischen wie menschlichen Hürden in der Kommunikation anspruchsvoll zu bewältigen sind, wozu vielen Unternehmen die Erfahrung fehlt.

**Erwartungen der Endnutzer werden nicht zufriedenstellend gepflegt:** Dieses Risiko betrifft jegliche Projekte, tritt jedoch in einem Umfeld, in dem Nutzer (-vertreter) nur erschwert mit den Entwicklern kommunizieren können, verstärkt auf.

**Mangelndes Domänen-Know-how der Offshoreteams:** Neben einem ursprünglich fehlenden Verständnis der Domäne fällt gegenüber einem lokal tätigen Team

Rang	Risiko	Risikoverminderung
1	Mangelnde Unterstützung seitens des Topmanagements	
2	Fehlkommunikation der ursprünglichen Anforderungen	✓
3	Sprachbarrieren in der Projektkommunikation	✓
4	Inadäquater Einbezug der Endnutzer	✓
5	Mangelnde Erfahrung in der Offshoreprojektleitung seitens des Auftraggebers	
6	Erwartungen der Endnutzer werden nicht zufriedenstellend gepflegt	✓
7	Mangelndes Domänen-Know-how der Offshoreteams	✓
8	Ungenügendes Change-Management	✓
9	Mangelndes technisches Können der Offshoreteams	
10	Falschkalkulation der anfallenden Gesamtkosten	

Tabelle 1: Bedeutendste Risiken in Offshoresoftwareentwicklungsprojekten gemäß [3], die Spalte „Risikoverminderung“ kennzeichnet Risiken, deren Risikopotenzial durch das vorgestellte SLA-basierte Zusammenarbeitsmodell für verteilte Softwareteams reduziert wird

der ineffektive Zugriff auf das seitens Auftraggeber vorhandene Wissen ins Gewicht.

**Ungenügendes Change-Management** bezieht sich auf die zusätzlichen Schwierigkeiten, Änderungen der ursprünglichen Anforderungen durch die längeren und indirekteren Kommunikationskanäle von Offshoreprojekten zu kommunizieren.

**Mangelndes technisches Können der Offshoreteams** beinhaltet auch die Unsicherheit, die Fähigkeiten eines Offshoreprojektteams richtig einzuschätzen.

**Falschkalkulation der anfallenden Gesamtkosten** erinnert an die Tatsache, dass vielfach versteckte Kosten in Offshoremodellen auftreten können, oder dass bekannte Posten, wie etwa Reisekosten oder der Mehraufwand, seitens Auftraggeber unterschätzt werden.

## **DIE KLASSISCHE ANTWORT AUF DIE HERAUSFORDERUNGEN**

In der Praxis kann man oft beobachten, wie Organisationen, die vor den oben aufgeführten Herausforderungen stehen, durch immer striktere Kontrollmechanismen versuchen, diesen Herr zu werden.

Eine typische Maßnahme besteht darin, auf vollständigen formalen Spezifikationen von hohem Detaillierungsgrad zu bestehen, bevor in Erwägung gezogen wird, mit der eigentlichen Entwicklung zu beginnen. Derartige Detailspezifikationen werden vielfach auch gleich zur Grundlage eines starren Vertragswerks zwischen den einzelnen Parteien gemacht. Andere Projektaktivitäten, wie etwa die Qualitätssicherung, werden ähnlich behandelt, wodurch ein von rigiden Meilensteinen gekennzeichneter sequenzieller Projektplan entsteht.

Solche Maßnahmen dienen primär der verstärkten plangetriebenen Kontrolle, ein Ansatz, der natürlich durchaus Offshoreprojekte zum Erfolg führen kann. Nichtsdestotrotz spricht auch die Anzahl gescheiterter Projekte für sich und zeigt auf, wie solche Methoden teilweise auch zu einem irreführenden Gefühl der Kontrolle führen können [3], sowie beinhaltete Referenzen.

Signifikant erscheint jedoch, dass solche Ansätze in der heutigen Zeit, in der sich agile Vorgehensmodelle und deren Vorteile zusehends etablieren, diesen diametral gegenüberstehen. So entfallen bei einer derart sequenziell geführten Offshorepartnerschaft viele der hierzulande zum Teil hart erarbeiteten Vorteile erfolgreich gelebter agiler Methoden. Ein solcher Vorteil ist die frühzeitige und regelmäßige Auslieferung lauffähiger, in sich abgeschlossener Software – also just jenes Mechanismus, der die zuverlässige Kontrolle des Projektstands anhand des eigentlichen Projektergebnisses er-

laubt. Weiter entfällt die feingranulare Möglichkeit, die Arbeiten nach damit zu erzielendem Geschäftsnutzen zu priorisieren. Dies erlaubt es, ein Projekt im Prinzip zu jeder Zeit abbrechen zu können – mit der Gewissheit, einerseits ein lauffähiges Produkt zu erhalten und andererseits für die getätigte Investition das bestmögliche Resultat erreicht zu haben. Nicht zu unterschätzen ist auch die Tatsache, dass die Zusammenarbeit in einem Gesamtprojektteam ungleich schwieriger wird, wenn ein Teil der Partner nach agilen Methoden vorgeht und beispielsweise die Offshorepartner plan- und kontrollgetrieben arbeiten.

## **EINE SLA-BASIERTE AGILE ALTERNATIVE**

Im Folgenden wird eine leichtgewichtige und praxiserprobte Alternative zum klassischen Ansatz präsentiert. Sie erleichtert maßgeblich die erfolgreiche Anwendung agiler Methoden in einem Offshoreumfeld und unterstützt somit die Realisierung der entsprechenden Vorteile im Gesamtprojekt. Gleichzeitig wird ein Mechanismus geboten, der eine zuverlässige Kontrolle des Projektfortschritts ermöglicht.

Das vorgeschlagene Zusammenarbeitsmodell basiert auf der Definition eines Service Level Agreements (SLA) zwischen Auftraggeber und Auftragnehmer einer Softwareentwicklung. Ein SLA besteht grundsätzlich aus zwei Komponenten: dem zu erbringenden *Service* und dessen *Level* bzw. die Güte der vereinbarten Dienstleistung.

Der Service des SLA wird im Folgenden für das Outsourcing von Softwareentwicklungsdienstleistungen generisch als „Implementation von Arbeitspaketen“ definiert. Jedes der Arbeitspakete beinhaltet konkrete funktionale Anforderungen und muss nicht funktionale Ansprüchen genügen.

Der Service Level für die definierte Implementationsdienstleistung wird, wie im Folgenden beschrieben, anhand eines Komplexitätsmaßes der umzusetzenden Arbeiten festgelegt.

In realen Projekten konnte mittels dieses Ansatzes eine erfolgreiche Zusammenarbeit über Organisationsgrenzen hinaus in global verteilten Teams erreicht werden. Die Spalte „Risikoverminderung“ in „Error! Reference source not found.“ illustriert die sechs Risiken aus der Top-10-Liste von Risiken in Offshoresoftwareprojekten [3], deren Risikopotenzial durch den beschriebenen Ansatz direkt reduziert wird.

## **DER SERVICE – IMPLEMENTATION VON ARBEITSPAKETEN**

Die Dienstleistung, die im Rahmen des SLA die Zusammenarbeit der Vertragsparteien umschreibt, wird als

„Implementation von Arbeitspaketen“ definiert. Ein solches Arbeitspaket beinhaltet immer funktionale Anforderungen, die das geforderte Verhalten des Systems beschreiben. Die aus agilen Vorgehensmodellen bekannten User Stories [4] sind ideal geeignet, um derartige Arbeitspakete leichtgewichtig zu spezifizieren.

In einer User Story wird eine kleine, in sich abgeschlossene Funktionalität beschrieben, die vertikal integriert ist. Ein Vorteil dieser Struktur ist, dass die einzelnen Arbeitspakete – so weit wie möglich – voneinander unabhängig definiert werden. Dies erlaubt es verhältnismäßig einfach, die Prioritäten der einzelnen Arbeiten – und dank iterativ inkrementellem Vorgehen deren Inbetriebnahme – den sich ändernden Anforderungen im Geschäftsumfeld anzupassen. Weiter kann mit Methoden des Agilen Requirements Engineering die Ausarbeitung der Details bewusst zeitnah und parallel zur Umsetzung vorgenommen werden [5]. Dies entspricht agilen Prinzipien sowie den Paradigmen des Lean Management. Letztere beschwören etwa die Verhinderung von Verschwendung oder befürworten den Just-in-Time-Gedanken, um maximalen Nutzen

aus den Erkenntnissen des fortlaufenden Projekts zu erzielen.

In einer User Story liegt das Gros der Information in den beinhalteten Akzeptanzkriterien. Diese beschreiben, welche Bedingungen die Implementation der umzusetzenden Funktionalität erfüllen muss, um vom Auftraggeber abgenommen zu werden. Derartige Akzeptanzkriterien bilden die Grundlage der im Rahmen der SLA zu erstellenden Funktionalitäten.

In verteilten Teams eines Offshoremodells stehen Schwierigkeiten in der Kommunikation hinter vielen der bedeutendsten Risiken (Tabelle 1). Eine Technik, die geradezu prädestiniert ist, Missverständnisse bezüglich der Akzeptanzkriterien von User Stories zu verhindern, ist die verhaltensgetriebene Entwicklung (Behaviour-driven Development, BDD) [6]. BDD beruht auf der Definition einer allen Projektbeteiligten gemeinsamen Domänensprache sowie der Formulierung von Akzeptanzkriterien als Akzeptanztests. Ein Akzeptanztest ist im Gegensatz zu einem Akzeptanzkriterium nicht allgemein formuliert, sondern stellt ein konkretes Beispiel des gewünschten Verhaltens einer

Funktionalität dar. Dies geschieht in der etablierten, allen Projektbeteiligten geläufigen Domänensprache anhand von Vorbedingungen, einzelnen Testschritten sowie den erwarteten Nachbedingungen. „Error! Reference source not found.“ zeigt ein triviales Beispiel eines derartigen BDD-Akzeptanztests; die „gegebene“ Klausel stellt hier die einzige Vorbedingung dar, „Wenn“ beinhaltet die beiden Testschritte und „Dann“ die einzelne Nachbedingung (Kasten: „Fiktiver BDD-Akzeptanztest, [...]“).

Ein Akzeptanzkriterium kann somit typischerweise mit einer überschaubaren Anzahl an Akzeptanztests konkret und unmissverständlich beschrieben werden.

Ein Vorgehen nach BDD beinhaltet weiter, dass die Akzeptanztests bei der Implementation der entsprechenden Funktionalität von der Entwicklung automatisiert werden. Hinter jeder Klausel wird die entsprechende Funktionalität im echten System ausgeführt bzw. überprüft. Somit ist es möglich, die Einhaltung aller Akzeptanztests der umgesetzten Funktionalität zu jeder Zeit mit geringem Aufwand zu überprüfen. Im Rahmen einer Continuous-Integration-Lösung ist es so auch möglich, bei jeder Änderung des Quellcodes sicherzustellen, dass bestehende Funktionalität durch die Anpassung nicht tangiert worden ist.

Nicht funktionale Anforderungen an das System können, wie in agilen Vorgehensmodellen üblich, als Rahmenbedingungen für die Entwicklung festgelegt werden. Oft ist es sogar möglich, deren Einhaltung ebenfalls mittels automatischen Tests analog den funktionalen Akzeptanzkriterien sicherzustellen.

Das beschriebene Vorgehen bietet für viele der eingangs aufgeführten Risiken der Offshoresoftwareentwicklung direkte Maßnahmen zur Reduktion des Risikopotenzials (Tabelle 1).

**Fehlkommunikation der ursprünglichen Anforderungen:** Die konkreten Beispiele der BDD-Akzeptanztests leisten einen großen Beitrag zur unmissverständlichen Kommunikation von Anforderungen.

**Sprachbarrieren in der Projektkommunikation:** Die gemeinsame Projektsprache, die im Zuge eines BDD-Vorgehens etabliert wird, erlaubt es dem Projektteam, ein gemeinsames Verständnis der relevanten Domänenbegriffe zu erreichen. Dies kann zwar grundsätzliche Sprachbarrieren nicht entfernen, bringt jedoch im Projektalltag Vorteile für das gegenseitige Verständnis.

**Inadäquater Einbezug der Endnutzer/Erwartungen der Endnutzer werden nicht zufriedenstellend gepflegt:** Der Anspruch, konkrete Akzeptanztests zu formulieren, erzeugt durch den notwendigen Detaillierungsgrad

## Fiktiver BDD-Akzeptanztest, der anhand eines konkreten Beispiels ein Akzeptanzkriterium illustriert

**Szenario:** Fahrzeugvermietung mit Anforderungen bezüglich Anzahl Passagiere, Gepäck und Preis

**Gegeben:** Folgende Fahrzeuge stehen in der Vermietung zur Auswahl

Name	Passagiere	Gepäck [kg]	Preis [Euro/Tag]
Fahrrad	1	25	10
Rikscha	3	25	20
Motorrad	2	50	30
Kleinwagen	4	100	50
Kombi	5	500	80
Minibus	9	300	120
Pick-up	3	1 000	90

**Wenn** ein Kunde ein Fahrzeug für 3 Passagiere und 200 kg Gepäck bestellt

**Und** der Maximalpreis bei 100 Euro liegt

**Dann** können folgende Fahrzeuge angeboten werden

Name	Passagiere	Gepäck [kg]	Preis [€/Tag]
Kombi	5	500	80
Pick-up	3	1 000	90

eine starke Motivation, die Endnutzer bei der Spezifikation der Anforderungen miteinzubeziehen. Das Schreiben der präzisen Akzeptanztests unterstützt einen dabei, Missverständnisse durch versteckte Annahmen zu vermeiden.

**Mangelndes Domänen-Know-how der Offshore-teams:** Das relevante Domänenwissen wird durch die Beispiele der Akzeptanztests sehr präzise beschrieben. Dies ist eine wertvolle Unterstützung bei der Vermittlung der entsprechenden Kenntnisse.

**Ungenügendes Change-Management:** Änderungen der Anforderungen werden durch dasselbe Vorgehen wie die ursprünglichen Anforderungen gehandhabt, wodurch sie sich nahtlos in die Gesamtspezifikation einfügen. Die automatisierten Tests der umgesetzten Anforderungen können zu jeder Zeit dazu verwendet werden, Widersprüche aufzudecken sowie den effektiv umgesetzten Stand der Software zu überprüfen.

## DER SERVICE LEVEL – KOMPLEXITÄT ALS MASS ALLER DINGE

Der Service Level für die definierte Dienstleistung „Implementation von Arbeitspaketen“ wird bezüglich Qualität und Quantität der Dienstleistung spezifiziert. Die Qualität wird im vorgeschlagenen Vorgehen, wie in agilen Vorgehensmodellen üblich, entsprechend den projektspezifischen Anforderungen als Rahmenbedingungen der zu leistenden Arbeiten definiert und kontrolliert.

Für die quantitative Sicht auf den Service Level muss eine Aufwandsschätzung der zu leistenden Arbeiten vorgenommen werden. Die große Herausforderung in der Offshoresoftwareentwicklung ist dabei, dass jedes Entwicklungsteam eine unterschiedliche Entwicklungsgeschwindigkeit leistet. Weiter kann sich diese über Zeit auch signifikant ändern, etwa weil das Team an Erfahrung gewinnt oder weil personelle Wechsel stattfinden.

In der Praxis haben sich daher Aufwandsschätzungen basierend auf einem Komplexitätsmaß bewährt. Gegenüber zeitbasierten Schätzungen bieten sie den Vorteil, dass die Größe der zu leistenden Arbeit unabhängig von der Entwicklungsgeschwindigkeit des Entwicklungsteams geschätzt wird [4]. Im vorgeschlagenen SLA hat dies zur Folge, dass der Auftragnehmer anhand der Komplexität der in einer Iteration ausgelieferten Funktionalität entlohnt wird, unabhängig vom geleisteten Zeitaufwand. Um zeitliche Planungssicherheit zu erlangen, können diesbezüglich zusätzlich Rahmenbedingungen im SLA verankert werden. In realen Projekten pendelt sich jedoch die Entwicklungsgeschwindigkeit eines Teams bezüglich Implementation anhand des verwendeten Komplexitätsmaßes über einige wenige Planungsiterationen ein.

Das grundsätzliche Vorgehen bei Komplexitätsschätzungen besteht darin, eine zu schätzende Arbeit mit bekannten (Referenz-)Arbeiten zu vergleichen, um die Komplexität zu bestimmen. In der Praxis kann ein vertrautes Entwicklungsteam herangezogen werden, um während einer Kalibrierungsphase ein unbekanntes Team mit unabhängigen Schätzungen zu begleiten. Je nach Projektkontext kann das neue Team mit fortschreitender Vertrautheit selbstständig Aufwandsschätzungen vornehmen. Alternativ kann das vertraute Team weiterhin, zumindest stichprobenartig, zur Plausibilisierung der Aufwandsschätzungen beigezogen werden.

## FAZIT

Im Artikel ist ein praxiserprobtes Modell zur Zusammenarbeit von verteilten Teams in Offshoreprojekten vorgestellt worden, das auf der Definition eines Service Level Agreements zwischen den Vertragsparteien beruht. Der Service ist als „Implementation von Arbeitspaketen“ definiert, deren Größe durch die beinhaltete Komplexität geschätzt wird. Dieser Ansatz erlaubt es, auch in Offshore-Outsourcing-Projekten eine hohe Flexibilität in der Projektausführung und zugleich eine zuverlässige Kontrolle des Projektfortschritts zu erreichen.

## Links & Literatur

- [1] Couto, V.; Mani, M.; Lewin, A. Y.; Peeters, C. (2006): „The Globalization of White-Collar Work, Booz Allen Hamilton“
- [2] Lewin, A. Y.; Zhong, X.; Gallagher, K. (2011): „Organizational Flexibility: The Strategic Differentiator of Global Sourcing Effectiveness“, Duke University Offshoring Research Network (ORN)
- [3] Nakatsu, R. T.; Iacovou, C. L. (2009): „A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study“, Information & Management, 46 (1), pp. 57–68
- [4] Cohn, M. (2004): „User Stories Applied: For Agile Software Development“, Addison-Wesley Professional
- [5] Leffingwell, D. (2011): „Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise“, Addison-Wesley Professional
- [6] Ettl, A. (2012): „Eine gemeinsame Sprache – Automatisierte Akzeptanztests bilden eine lebende Softwarespezifikation“, Business Technology 1.2012, S. 51–56



### Dr. Alan Ettl

ist Projektleiter, Berater und Coach bei bbv Software Services AG ([www.bbv.ch](http://www.bbv.ch)). Sein Fokus liegt auf agilen Methoden und deren holistischen Einsatz von Technik bis Management. Er hat bei einer Vielzahl von Kunden aus verschiedenen Sektoren solche Methoden eingeführt und „optimiert“.