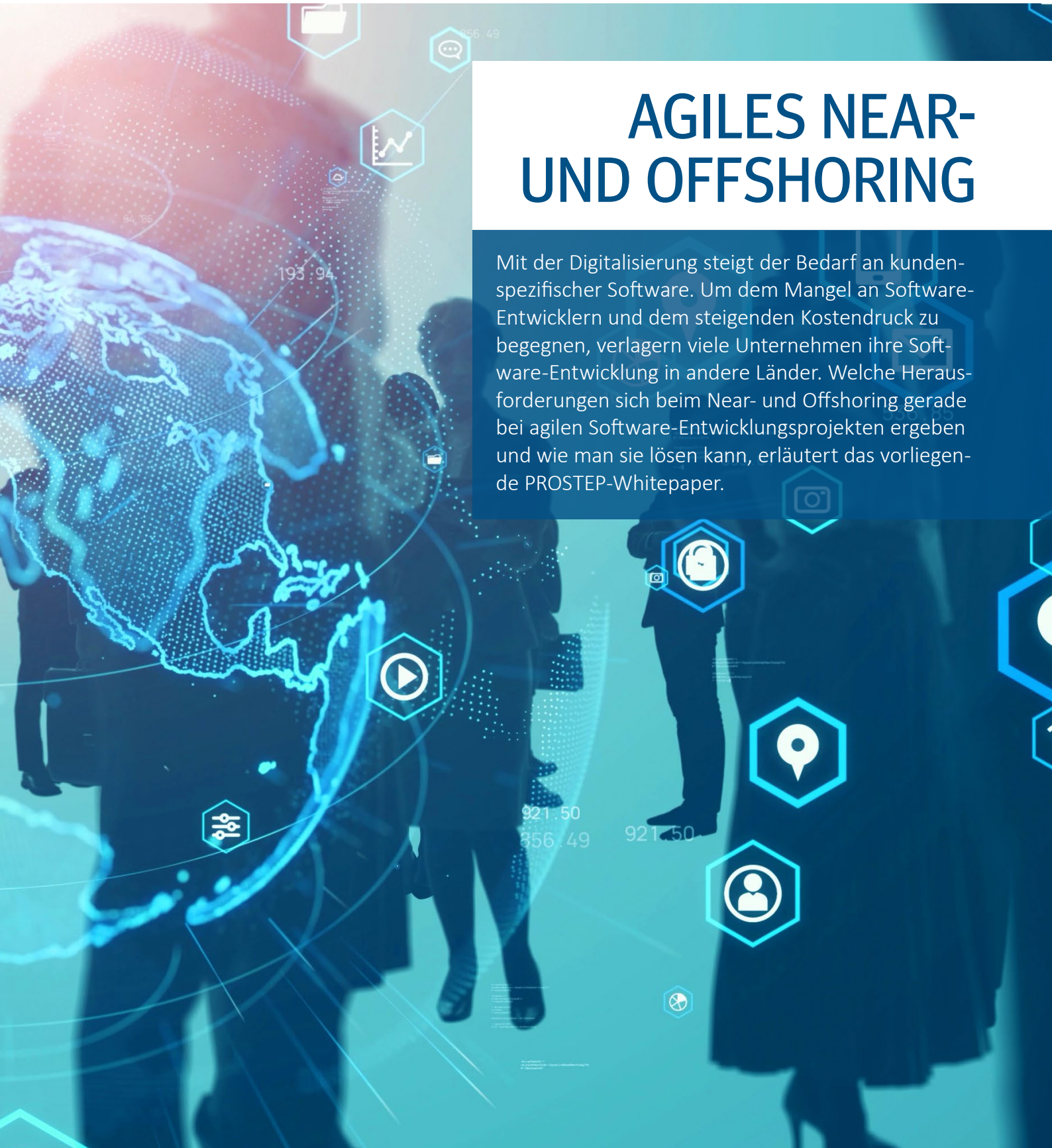


AGILES NEAR- UND OFFSHORING

Mit der Digitalisierung steigt der Bedarf an kunden-spezifischer Software. Um dem Mangel an Software-Entwicklern und dem steigenden Kostendruck zu begegnen, verlagern viele Unternehmen ihre Software-Entwicklung in andere Länder. Welche Herausforderungen sich beim Near- und Offshoring gerade bei agilen Software-Entwicklungsprojekten ergeben und wie man sie lösen kann, erläutert das vorliegende PROSTEP-Whitepaper.





Inhalt

Vorwort	3
Kosten sparen durch Near- und Offshoring.....	4
Agile Vorgehensweise als Herausforderung.....	5
Bewusstsein für die kulturellen Unterschiede	7
Vor- und Nachteile von Zeitunterschieden.....	8
Erfolgreich bei klar umrissenen Aufgaben	9
Gute Erfahrungen mit Onsite Delegates	11



Vorwort

Der Bedarf an kundenspezifischer Software ist mit der Digitalisierung in den letzten Jahren stetig gestiegen. Gartner, Inc. zufolge wuchs der weltweite Markt für Software-Entwicklungsdienste vor der COVID-19-Pandemie jährlich um fünf Prozent und erreichte 2019 ein Volumen von 70 Milliarden US-Dollar. Zeitgleich mit diesem Wachstum hat eine Verlagerung der Software-Entwicklung in andere Länder (Near- und Offshoring) stattgefunden. Verantwortlich dafür ist unter anderem der eklatante Mangel an Software-Entwicklern in Europa. Allein in Deutschland waren laut bitkom (Bundesverband Informationswirtschaft, Telekommunikation und neue Medien) Ende letzten Jahres 124.000 Stellen für IT-Fachkräfte unbesetzt; besonders begehrt sind Software-Entwickler. „Haupttreiber für das Near- und Offshoring von Entwicklungsaufgaben ist jedoch der steigende Kostendruck, dem unsere Kunden und auch wir ausgesetzt sind“, sagt Rainer Zeifang, Chief Technology Officer Daimler Projects bei der PROSTEP AG. „Unsere Kunden aus der Automobil- und Fertigungsindustrie sind global aufgestellt und nutzen die Einsparungspotenziale, die sie durch diese globale Aufstellung erreichen. Dasselbe erwarten sie auch von uns.“



Rainer Zeifang bringt aus seiner aktuellen Funktion bei PROSTEP und seiner langjährigen Tätigkeit in Forschung & Entwicklung bei verschiedenen PLM-Softwareherstellern viel Erfahrung aus Near- und Offshore-Projekten mit. In diesem Whitepaper erläutert er die Herausforderungen, die sich bei der Zusammenarbeit mit Near- und Offshore-Partnern vor allem in agilen Software-Entwicklungsprojekten ergeben, und wie man sie lösen kann.



Kosten sparen durch Near- und Offshoring

Die Unterscheidung zwischen Near- und Offshoring ist nicht nur eine Frage der Entfernung, auch wenn diese sicher ein entscheidendes Kriterium ist. Wichtig sind daneben auch andere Kriterien, wie zum Beispiel die kulturellen Unterschiede zwischen den Partnern oder die Frage, wie groß die Zeitunterschiede zwischen den beteiligten Standorten sind. Je größer die kulturelle Nähe und je geringer die Entfernung, desto eher spricht man von Nearshoring. In den letzten Jahren ist in Europa ein klarer Trend zum Nearshoring festzustellen, was unter anderem mit der wachsenden Bedeutung des Datenschutzes zusammenhängt, der bei der Zusammenarbeit mit Offshore-Partnern in Fernost schwieriger zu gewährleisten ist. Dessen ungeachtet ist Indien gemessen an der Zahl der Arbeitsstunden nach wie vor der mit Abstand größte Exporteur von Software-Entwicklungsdiensten weltweit.

PROSTEP arbeitet sowohl bei der Entwicklung der eigenen Software-Produkte als auch bei Kundenprojekten seit geraumer Zeit mit ausgewählten Near- und Offshore-Partnern zusammen. Außerdem praktiziert das PLM-Beratungs- und Softwarehaus das Nearshoring auch intern. Seit einem Jahr unterhält es eine Niederlassung im polnischen Wroclaw (ehemals Breslau), die die Entwicklungsmannschaft der Berliner PROSTEP-Niederlassung bei Software-Entwicklungsprojekten für große Automobilkunden durch agile SCRUM-Teams unterstützt. „Als mittelständisches Unternehmen können wir nicht alle Fähigkeiten, die wir für komplexe PLM-Implementierungen benötigen, immer beim Kunden vor Ort vorhalten“, sagt Zeifang. „Benötigen wir in einem Projekt die Expertise eines Teams über einen längeren Zeitraum, kann man auch hier von Near- bzw. Offshoring sprechen.“

Frage:

Wie groß sind die Kosteneinsparungen durch Near-/Offshoring?

Zeifang:

Das hängt sehr stark von der Aufgabenstellung ab. Handelt es sich um Standard-IT-Aufgaben, die das Near- oder Offshore-Team eigenständig und mit wenig Interaktion mit dem Auftraggeber erledigen kann, machen die möglichen Einsparungen fast die Differenz der Personalkosten beim Kunden beziehungsweise dem Onsite-Team aus. Bei PLM-Implementierungsprojekten, in denen wir unterwegs sind, ist jedoch meist eine Transferleistung zwischen dem Kunden und seinen Anforderungen aus den Fachbereichen beziehungsweise der IT und den Mitarbeitern des Near- oder Offshore-Teams erforderlich. Deshalb arbeiten wir in der Regel mit gemischten Teams on-site und offshore. Dadurch ergeben sich immer noch Einsparpotentiale von 30 bis 50 Prozent gegenüber einem reinen Onsite-Team.



Agile Vorgehensweise als Herausforderung

Frage:

Beißt sich das Near- und Offshoring nicht eigentlich mit dem agilen Vorgehen?

Zeifang:

Große Softwareprojekte werden eh in kleinere, überschaubare Teilprojekte unterteilt. Dabei muss nicht jeder den Überblick über alle Teilprojekte haben. Wählt man den Schnitt zwischen den Teilprojekten so, dass sie möglichst unabhängig voneinander sind, kann jedes Team für sich agil arbeiten. Die agile Abstimmung zwischen den verteilten Teams erfordert einen gewissen Mehraufwand, der aber aus meiner Sicht durchaus vertretbar ist.

Unabhängig von der Vorgehensweise gibt es beim Near- bzw. Offshoring eine Reihe von Herausforderungen, die durch das agile Vorgehen natürlich akzentuiert werden. Die Partner müssen ein gemeinsames Verständnis für das Kundenprojekt entwickeln und Know-how austauschen, das normalerweise in den Köpfen der Entwickler steckt. Sie müssen einheitliche Vorgehensweisen etablieren, um sicherzustellen, dass die zu entwickelnde Software trotz verteilter Teams und großer Entfernung aus einem Guss ist und eine einheitliche User Experience ermöglicht. Und sie müssen Kommunikationshürden abbauen beziehungsweise neue Formen der Kommunikation finden, die mit den agilen Vorgehensweisen kompatibel sind.

Gemäß dem Manifest der agilen Software-Entwicklung liegt die Betonung bei agilen Projekten auf den Individuen und ihrer Interaktion und nicht auf den Werkzeugen und Prozessen. Wichtiger als Vertragsverhandlungen ist die vertrauensvolle Zusammenarbeit mit dem Kunden. Typischerweise werden diese Prinzipien in SCRUM-Teams umgesetzt und gelebt, die nahe bei ihrem Auftraggeber ansässig sind und mit den Mitarbeitern des Kunden sehr eng zusammenarbeiten. Bei verteilten Teams ist das schwieriger, aber es ist möglich.

Zu jeder agilen Praktik, die vor Ort genutzt wird, gibt es eine Entsprechung für verteilte Teams. So werden beispielsweise die Daily SCRUM Meetings, in denen sich die Teammitglieder darüber austauschen, was sie am Vortag erreicht haben, was sie als nächstes tun werden und wo sie blockiert sind, durch kurze Telefon- oder Video-Konferenzen ersetzt. Dank Screen-Sharing- und Videokonferenz-Tools können die Entwickler sogar paarweise am Code arbeiten, wie beim eXtreme Programming (XP) üblich. Entscheidend ist, dass die Mitarbeiter der Near- und Offshore-Teams die Möglichkeiten und das Gefühl haben, wie die Kollegen vor Ort in die Projektarbeit eingebunden zu sein. Regelmäßige Kommunikation ist das A und O eines funktionierenden Near- oder Offshore-Projekts.

Frage:

Wie wichtig ist dabei der persönliche Kontakt und Austausch?

Zeifang:

Er ist sehr wichtig und trägt entscheidend zum Projekterfolg bei. Wir stellen insbesondere zu Beginn des Projekts, aber auch über die Projektlaufzeit sicher, dass ein regelmäßiger persönlicher Austausch stattfindet. Wir besuchen die Offshore-Teams, um uns im persönlichen Gespräch über Motivationen, Ansichten und Ziele auszutauschen. Gerade zu Projektbeginn ist es wichtig, dass sich die Key-Player persönlich kennenlernen, um Know-how auszutauschen, aber auch um zu verstehen, wie die anderen ticken, was sie bewegt und wie sie arbeiten.



Bewusstsein für die kulturellen Unterschiede

Sprachbarrieren lassen sich nach Einschätzung von Zeifang leichter überwinden als kulturelle Barrieren. Englisch ist die Lingua franca in Near- und Offshore-Projekten, um sicherzustellen, dass alle Beteiligten in der gleichen Sprache kommunizieren können. PROSTEP achtet außerdem darauf, dass nicht nur die Anforderungen in Form von Epics und User Stories in Englisch formuliert werden, sondern auch die Dokumentation des Software-Designs, die Variablen und Kommentare im Sourcecode und die Protokolle der Meetings.

Der Abbau kultureller Barrieren setzt Offenheit und Toleranz bei allen Beteiligten voraus, und er fängt zu Hause an. PPROSTEP ist zwar in vielerlei Hinsicht ein typisch deutsches Unternehmen, es profitiert aber diesbezüglich von seiner multikulturellen Zusammensetzung mit Mitarbeiterinnen und Mitarbeitern aus 20 Nationen. Das schärft das Bewusstsein für die kulturellen Unterschiede. Nicht nur die Onsite-, sondern auch die Offshore-Teams müssen sich dieser Unterschiede bewusst sein und offen damit umgehen. Das gilt auch für unterschiedliche Gepflogenheiten, wie Zeifang am Beispiel eines Offshore-Teams erläutert, das gewohnt war, mit einem Software-Tester zu arbeiten. Dieser Umstand wurde beim Team-Setup ignoriert, was prompt zu schlechter Software-Qualität führte. Als das Offshore-Team das offen ansprach, ergänzte PROSTEP das Team um einen Tester und die Qualität wurde unverzüglich besser.

Frage:

Wie wichtig ist ein offener Umgang mit Fehlern für das agile Arbeiten?

Zeifang:

Eine offene Fehlerkultur und flache Hierarchien erleichtern das agile Arbeiten ungemein. Die Teams müssen sich selbst organisieren können und fähig sein, in kurzen Iterationen neue Software zu produzieren und auch wieder zu verwerfen, wenn sie nicht funktioniert oder die Anforderungen der Anwender nicht erfüllt. Das ist aber weniger eine Frage kultureller Unterschiede als der unterschiedlichen Firmenkulturen, auf die man bei der Wahl der Partner achten muss.



Vor- und Nachteile von Zeitunterschieden

Zeitunterschiede spielen bei der Zusammenarbeit mit Offshore-Partnern in Fernost eine wichtige Rolle. Sie müssen aber nicht immer ein Hindernis sein, sondern bieten durchaus Vorteile im Sinne des Arbeitens rund um die Uhr. So können beispielsweise die Kollegen in Indien etwaige Probleme, die ein deutscher Kollege beim Check-in seines Software-Codes kurz vor Arbeitsschluss festgestellt hat, schon in den verbleibenden Stunden des indischen Arbeitstags beheben. Das Team in Deutschland kann dann gleich am nächsten Morgen auf einem verbesserten Code-Stand weiterarbeiten.

Bei allen Themen, die einen Austausch zwischen Onsite- und Offshore-Team erfordern, müssen die virtuellen Meetings natürlich während der gemeinsamen Arbeitszeit durchgeführt werden. Dies ist bei einem Zeitunterschied von 3,5 Stunden im Sommer und 4,5 im Winter zwischen Deutschland und Indien noch gut möglich. Schwieriger wird es bei ganztägigen Events wie der Program-Increment-Planung im Rahmen skalierter Vorgehensweisen a la SAFe, die Flexibilität auf beiden Seiten erfordern. Man kann etwa den Termin nach deutscher Zeit so früh wie möglich ansetzen und die indischen Kollegen bitten, ihre Arbeitszeiten etwas nach hinten zu verschieben.

Frage:

Funktioniert das Near- und Offshoring mit allen agilen Vorgehensmodellen wie Scrum, LeSS, NEXUS oder SAFe?

Zeifang:

Wir haben sehr gute Erfahrungen mit SCRUM und SAFe gemacht und ich sehe keinen Grund, warum das nicht auch mit anderen Vorgehensmodellen funktionieren sollte. Ob die Vorgehensmodelle angepasst werden müssen, hängt nicht von der Verlagerung, sondern von den Geschäftsanforderungen des Kunden an. In unseren Projekten haben wir unabhängig vom Vorgehensmodell 2- bis 3-wöchige Sprints. In frühen Entwicklungsphasen sind auch kürzere Zyklen denkbar. Entscheidend ist, dass die CI / CD Pipeline (Continuous Integration / Continuous Delivery) gut funktioniert, dass alle Teams kontinuierlich ihren Code liefern und das Software-Produkt regelmäßig gebaut und getestet wird.



Erfolgreich bei klar umrissenen Aufgaben

Nach den Erfahrungen von PROSTEP lassen sich sowohl Aufgaben im Bereich der Entwicklung und Qualitätssicherung von Software als auch rund um das Thema Software-Entwicklungs-Tools und -Support gut zu Near- oder Offshore-Partner verlagern. Immer dann, wenn ein klar umrissener Aufgabenbereich mit eigener Verantwortung verlagert werden kann, funktioniert das Near- beziehungsweise Offshoring gut. Dies kann eine Softwarekomponente sein, ein Test bestimmter Funktionalitäten oder auch die Verantwortung für Entwicklungstools wie einen Buildserver oder ein Sourcecode-Analysetool. „Was aus meiner Sicht geringe Aussichten auf Erfolg hat, sind Projekte mit verteilter Verantwortlichkeit zwischen Onsite- und Offshore-Team, bei denen sich jede Seite auf die andere verlässt“, sagt Zeifang.

Frage:

Wie müssen die verteilten agilen Teams aufgestellt sein?

Zeifang:

Normalerweise gibt der Kunde in unseren Projekten die Anforderungen vor und definiert die Epics und User Stories für die agile Projektabwicklung. Um die Anforderungen optimal umsetzen zu können, müssen die Onsite-Teams gut in die Organisation des Kunden eingebunden sein. Bei Near- beziehungsweise Offshore-Projekten hat sich außerdem bewährt, wenn die Near- bzw. Offshore-Teams permanente Vertreter entsenden, die sogenannten Onsite Delegates, die als Bindeglied zum Onsite-Team beziehungsweise zum Kunden fungieren. Generell gilt, je unabhängiger das Team ist, desto agiler kann es arbeiten und desto größer sind die Erfolgsaussichten.

Frage:

Welche Rollen gibt es in den agilen Near- bzw. Offshore-Teams?

Zeifang:

Die Near- bzw. Offshore-Teams bestehen im Wesentlichen aus denselben Rollen wie die Onsite-Teams. Im SCRUM-Prozess sind das der Product bzw. der Proxy Product Owner (PPO), der SCRUM Master und unterschiedliche Rollen im Entwicklungsteam wie Entwickler, Software-Architekten oder Tester. Ob diese Rollen on-site oder offshore vergeben werden, hängt vom Projekt und den Skill Sets der beteiligten Personen ab. Was die Größe der Teams angeht, so gilt wie für alle agilen Teams die 2-Pizza-Regel, d.h. sie sollten mit zwei (großen) Pizzen satt werden. Die Offshore-Teams haben maximal eine Größe von acht bis zehn Leuten, einschließlich der Onsite Delegates, die unterschiedliche Rollen innehaben können.



Gute Erfahrungen mit Onsite Delegates

Welche Rollen die Onsite Delegates übernehmen, hängt von der Projektart und -größe ab. Soll eine komplett neue Lösung entwickelt werden, bietet es sich an, dafür einen Technical Lead Developer oder Software-Architekten des Teams auszuwählen. Dieser übersetzt die Kundenanforderungen in eine Architektur und arbeitet zusammen mit dem Offshore-Team an der Implementierung der entsprechenden Lösung. Außerdem präsentiert er dem Kunden die Lösung und sorgt dafür, dass neue Anforderungen in kurzen Zyklen umgesetzt werden.

Arbeitet man im SCRUM-Prozess, so macht es bei größeren Teams Sinn, einen SCRUM Master als Onsite Delegate zu benennen. Voraussetzung ist, dass die Aufgaben klar umrissen sind und es ein hohes Implementierungsvolumen gibt.

In seiner Rolle als Onsite Delegate stellt der SCRUM Master sicher, dass das Product Backlog gut gefüllt ist, dass das Team die Ziele und den Umfang der darin enthaltenen User Stories versteht und das Product Backlog somit effizient abarbeiten kann. Er sorgt dafür, dass die User Stories einen Reifegrad haben, die den Anforderungen des Offshore-Teams entspricht, und dass die Akzeptanzkriterien klar umrissen sind. Er unterstützt den Product Owner bei der Priorisierung des Backlogs, um das bestmögliche Resultat zu erzielen. Er hilft dem Offshore-Team bei der Beseitigung von Hindernissen und bei der Durchführung von SCRUM Events. „In Organisationen, in denen SCRUM noch nicht vollständig implementiert und akzeptiert ist, fungiert er zudem als Coach des Entwicklungsteams und arbeitet mit anderen SCRUM Mastern zusammen, um die Implementierung in der Organisation zu verbessern“, führt Zeifang weiter aus.

Auch die Rolle eines Proxy Product Owners als Bindeglied zwischen dem Product Owner beim Kunden und dem Offshore-Team kann von einem Onsite Delegate ausgefüllt werden. Dieser arbeitet mit dem Product Owner auf Kundenseite zusammen, definiert mit ihm zusammen die User Stories, ist fachlicher Ansprechpartner für das Offshore-Team und beantwortet dessen Fragen gegebenenfalls in Rücksprache mit dem Product Owner.

In allen Rollen ist es sehr wichtig, dass der Onsite Delegate des Offshore-Teams über entsprechende Softskills verfügt. Er muss die kulturellen Unterschiede zwischen dem Kunden, den Onsite- und dem Offshore-Team verstehen, um die Akzeptanz des Offshore-Teams auf Seiten des Kunden sicherstellen zu können, und er muss seinem Team die Qualitätsansprüche des Kunden vermitteln können.

Frage:

Und Corona? Wie wird sich das auf das Near- und Offshoring auswirken?

Zeifang:

Generell hat Corona zu einem rasanten Wandel in der Arbeitswelt geführt. Dank Homeoffice ist das verteilte Arbeiten zum Regelfall und jedes Software-Entwicklungsprojekt zumindest zu einem Nearshore-Projekt geworden, bei dem sogar die Teams nicht mehr an einem Standort sitzen, sondern räumlich verteilt sind. Wer das Arbeiten in verteilten Teams gewohnt war, dem ist der Umstieg auf die neue Normalität sicher leichter gefallen. Uns kam in unseren Projekten zugute, dass wir gut eingespielte Onsite- und Offshore-Teams hatten. Das Neuaufsetzen eines verteilten Teams mit Mitarbeiter, die sich nicht kennen und erstmals zusammenarbeiten müssen, stelle ich mir allerdings wesentlich schwieriger vor.



PDF Version des Whitepapers:
www.prostep.com/whitepaper
oder scannen Sie den QR Code



Sie haben Anmerkungen oder Fragen?

Wir freuen uns auf Ihr Feedback an
infocenter@prostep.com

PROSTEP AG
Dolivostraße 11 · 64293 Darmstadt · Deutschland
Telefon +49 6151 9287-0 · Telefax +49 6151 9287-326 · E-Mail info@prostep.com

© 2021 PROSTEP AG. Alle Rechte vorbehalten.
Alle durch ® oder ™ gekennzeichneten Marken sind das Eigentum ihrer jeweiligen Inhaber.

IMPRESSUM

Herausgeber
PROSTEP AG

Ansprechpartner:
Rainer Zeifang
rainer.zeifang@prostep.com

Edition 1, 2021