

KOOPERATION STATT MISSTRAUEN – KUNDENBEZIEHUNG IN AGILEN PROJEKTEN

Agile Verfahren zur Softwareentwicklung – wie „Crystal“, „Extreme Programming“ oder „Scrum“ – haben in den letzten Jahren zunehmend an Bedeutung gewonnen. Im Mittelpunkt des Interesses stand dabei meist die Veränderung der Entwicklungstechnik. Der wirtschaftliche Vorteil eines agilen Verfahren kann jedoch erst vollständig ausgeschöpft werden, wenn sich auch die Zusammenarbeit zwischen Auftraggebern und Auftragnehmern ändert. „Play to win“ statt „Play not to loose“ pointiert dieses Umdenken in der plakativen Sprache des Marketing. Dieser Artikel beleuchtet den dafür notwendigen kulturellen Wandel auf beiden Seiten.¹⁾

Als sich die Urheber der agilen Verfahren im Februar 2001 in Salt Lake City trafen, um ihre Verfahren zu vergleichen und Differenzen auszuräumen, stellten sie fest, dass bei allen Unterschieden jene vier Grundwerte alle ihre Verfahren tragen, die auf diesem Treffen im agilen Manifest festgehalten wurden (siehe www.agilemanifesto.org). Den ersten Platz auf dieser Liste erhielt die Feststellung „We value close customer collaboration over contract negotiation“ – eine Feststellung, die vielfach als Provokation verstanden wurde, insbesondere in Kulturen mit einer ausgeprägten juristischen Tradition.

Seitdem wurde in Anwender- und Diskussionsforen viel über einzelne Praktiken der agilen Entwicklung gestritten, bis hin zu Verirrungen, wie der Frage, ob die Karteikarten zur Aufzeichnung von Anforderungen im *Extreme Programming* (XP) unbedingt das im zugehörigen Buch (vgl. [Bec00]) empfohlene amerikanische 3x5-Zoll-Format brauchen oder man auch die europäischen DIN-A6-Karten einsetzen könne. Wenig Beachtung fanden dagegen bisher die Konsequenzen, die ein agiles Vorgehen auf die Zusammenarbeit zwischen Auftragnehmern und Auftraggebern hat.

Wird ernsthaft erwartet, dass agile Projekte im vertragsfreien Raum stattfinden? Natürlich nicht, schließlich handelt es sich um Verfahren für eine professionelle Softwareentwicklung, die sich in normalen Geschäftsbeziehungen abspielt. Überworteilen die Verfahren den Auftraggeber,

indem sie ihm die Kontrollmöglichkeiten entziehen? Auch das nicht, ein solches Vorgehen wäre mittelfristig zum Scheitern verdammt. Dennoch ist es wenig sinnvoll, die in der Industrie üblichen Mechanismen der Zusammenarbeit bei agilen Projekten unverändert zu übernehmen. Zwar sind solche Projekte durchaus möglich, doch verzichten beide Seiten dabei auf wirtschaftliche Vorteile.

Es erstaunt ein wenig, dass die Diskussion über agile Verfahren sich dennoch bisher weitgehend auf technologische Fragen konzentriert. Erst im letzten Jahr hat eine umfangreichere Diskussion zu dem Thema begonnen. Hervorzuheben sind dabei vor allem die Arbeiten von Mary Poppendieck, die Lieferantenbeziehungen aus der Automobilindustrie analysiert und auf Softwareprojekte übertragen hat, sowie das derzeit an der University of Wellington in Neuseeland unter der Leitung von James Noble laufende Forschungsprojekt von Angela Martin zur Kundenbeziehung bei XP. Auch Jim Highsmith nimmt sich im Rahmen des „Agile Project Management Advisor“ der Cutter IT Gruppe immer wieder des Themas an.

Die Beziehung zwischen Auftraggebern und -nehmern spielt eine zentrale Rolle in der agilen Entwicklung. Diese geht davon aus, dass Softwareprojekte einem ständigen Wandel unterworfen sind. Projekte, die in der Lage sind, schnell darauf zu reagieren, wenn sich die wirtschaftlichen oder technischen Rahmenbedingungen ändern oder wenn sich im Rahmen der Projektarbeit neue Erkenntnisse ergeben, sind nach dieser Auffassung wirtschaftlicher als Projekte, die versuchen, einmal aufgestellte Pläne

► der autor



Jens Coldewey (E-Mail: jens_coldewey@acm.org) ist unabhängiger Berater. Er hilft großen Organisationen bei der Einführung agiler Entwicklung und ist als Vorstandsmitglied der Agile Alliance Non-Profit Organisation für die Koordination agiler Konferenzen zuständig.

und Verträge möglichst unverändert umzusetzen. „Wenn Sie einen Plan verfolgen,“ so Jim Highsmith, „bekommen Sie das System, das Sie geplant haben, nicht das, was Sie [zum Auslieferungszeitpunkt] brauchen“ (vgl. [Hig00]). Das folgende Beispiel macht den Unterschied deutlich²⁾:

Eine große Organisation erweitert ihre Vertriebswege und gibt dafür ein neues Vertriebssystem in Auftrag. Um die neuen Vertriebspartner mit möglichst wenig technischen Vorgaben einbinden zu können, wird der Zugang auf das System sowohl über das Intranet als auch über das Internet in Auftrag gegeben. Der Vertrieb benötigt für seine Arbeit den Zugriff auf hoch sensible Daten, weshalb erhebliche Aufwände für die Absicherung des Internet-Zugangs anfallen. Das gesamte Projekt steht unter starkem Zeitdruck, da bei einer verspäteten Inbetriebnahme gesetzliche Vorgaben nicht mehr eingehalten werden können. Der Auftragnehmer übernimmt den Auftrag zum Festpreis und bindet weitere Unterauftragnehmer in das Projekt ein – unter anderem zur Absicherung des Internet-Zugangs. Das Projekt selbst folgt einem klassischen Wasserfallmodell.

Soweit die Situation zu Projektbeginn. Nach Abschluss der Konzeptphase signalisiert der Auftraggeber, dass entgegen der ursprünglichen Planung die Vertriebspartner doch wenige Monate nach Inbetriebnahme des neuen Vertriebssystems Zugriff auf das Intranet bekommen sollen. Der traditionell arbeitende Projektleiter des Auftragnehmers stellt zu diesem Zeitpunkt fest, dass der Internet-Zugang bereits in allen Konzepten

¹⁾ Dieser Beitrag ist die leicht gekürzte Fassung eines Vortrags, der im Rahmen des PMForums 2004 der Gesellschaft für Projektmanagement (GPM) zwischen dem 4. und 6. Oktober 2004 in Nürnberg gehalten wird. Der Abdruck erfolgt mit freundlicher Genehmigung der GPM.

²⁾ Die Beispiele in dieser Arbeit entstammen realen Projekten, wurden aber aus Vertraulichkeitsgründen anonymisiert und verfremdet.

vorgesehen sei. Änderungen daran würden Veränderungen am Vertrag erfordern, die der Kunde möglicherweise nutzen würde, um Preisabschläge auszuhandeln. Zudem sei das System mit Internet-Zugang eine Obermenge, sodass der Kunde höchstens mehr ausgeliefert bekäme, als er benötigt. Er veranlasst keine weiteren Aktionen in dieser Angelegenheit.

Agile Entwicklung ist darauf ausgelegt, den maximalen Mehrwert für den Auftraggeber zu erwirtschaften. Aus dieser Perspektive hätte ein Projektleiter in einem agilen Projekt in dieser Situation anders entschieden. Er hätte sich mit dem Auftraggeber in Verbindung gesetzt, um eine Neubewertung des Internet-Zugangs zu erreichen. Möglicherweise wäre es mit kleineren Anpassungen an dem neuen Vertriebskonzept möglich gewesen, vollständig auf den sensiblen Internet-Zugang zu verzichten bzw. das Thema erst dann anzugehen, wenn der Betrieb ohne Internet gesichert ist. Eine solche Veränderung der Prioritäten hätte verschiedene Konsequenzen:

- Die Implementierungsarbeiten für die Absicherung waren noch nicht gestartet. Könnte auf diese Arbeiten bis zum Auslieferungstermin verzichtet werden, würden Kapazitäten frei, um den Termin besser abzusichern. Die Einhaltung der gesetzlichen Vorgaben wäre besser abgesichert. Zusätzlicher Spielraum in der Planung kommt natürlich primär dem Auftragnehmer zugute, aber eine höhere Wahrscheinlichkeit für eine pünktliche Auslieferung ist auch für den Auftraggeber von Vorteil. Zudem lässt sich natürlich auch ein einseitiger Vorteil finanziell für die andere Seite attraktiv machen.
- Insgesamt wäre der Aufwand für die Entwicklung und den Betrieb des Systems deutlich geringer, wenn auf die aufwändige Absicherung des Internet-Zugangs verzichtet werden könnte. Schließlich sind ähnliche Absicherungsmechanismen ja schon in der Firewall des Intranets in Betrieb. Dies käme sowohl dem Auftraggeber als auch dem Auftragnehmer zugute.
- Der Wegfall des Internet-Zugangs würde ein potenzielles Sicherheitsrisiko im Betrieb eliminieren. Ein Einbruch in das System hätte einen erheblichen Imageverlust des Auftraggebers zur Folge und wäre wirtschaftlich kaum abzuschätzen. Dieser Vorteil wäre ausschließlich auf Seiten des Auftragnehmers.

Die Vorteile für den Auftraggeber könnten also durchaus die Nachteile kompensieren, die er hätte, stünde die Internet-Anbindung nicht zur Verfügung – eine Entscheidung, die der Auftraggeber nur treffen kann, wenn er die dafür notwendigen Informationen vom Auftragnehmer erhält. Eine Abänderung des Lieferumfangs wäre also unter Umständen im Interesse beider Parteien, wäre da nicht der Festpreisvertrag, der einen bestimmten Lieferungsumfang festlegt.

In dem Beispiel erkennt man einige Kernpunkte agilen Kundenmanagements: Der Auftragnehmer sieht sich als Anwalt des Auftraggebers. Nur wenn das Projekt vom Auftraggeber als erfolgreich bewertet wird, ist es auch für den Auftragnehmer erfolgreich. Man beachte, dass dies eine andere Auffassung von Projekterfolg ist als die übliche Forderung, den versprochenen Umfang zum versprochenen Termin mit dem versprochenen Aufwand auszuliefern. Ein gemäß dieser klassischen Definition weitgehend gescheitertes Projekt kann nach agilen Maßstäben sehr erfolgreich sein, wenn die Interessen des Auftraggebers gut vertreten wurden und die ausgelieferte Software dem Auftraggeber einen entsprechenden Mehrwert für seine Investitionen geliefert hat.

Die wichtigste Kenngröße des Projekts wird von Anfang an festgelegt. Ihr müssen sich alle anderen Entscheidungen und Interessen unterordnen. In unserem Beispiel wäre dies der durch gesetzliche Vorgaben bedingte Auslieferungstermin. Ihm werden Umfang und Aufwand untergeordnet. Allen Beteiligten ist klar, dass es Situationen gibt, in denen ein Ziel nur noch auf Kosten der anderen Ziele erreicht werden kann. Die Position, „Das haben Sie doch vertraglich zugesichert, das ist Ihr Problem“, weicht der Erkenntnis, dass Softwareprojekte schwierig sind und es eben nicht nur eine Frage des guten Willens ist, sie genau so abzuwickeln, wie sie geplant wurden.

Prioritäten und Umfang eines Projekts werden angepasst, wenn sich neue fachliche oder technische Erkenntnisse ergeben. Dies geschieht in Zusammenarbeit von Auftragnehmer und -geber. Die gute und offene Kommunikation zwischen beiden Parteien ist dabei oberstes Gebot. Insbesondere müssen Informationen offen weitergegeben werden und dürfen nicht nach politischen Erwägungen gefiltert werden. Nach wie vor ist es natürlich wichtig, die Informationen nach Relevanz zu filtern, um zu verhindern, dass sie vom Management überbewertet werden: Wer es gewohnt ist, nur von projektgefährdenden Problemen unterrichtet zu werden, neigt leicht dazu, die

alltäglichen Problemchen, die in jedem Projekt auftreten, überzubewerten und entsprechende Überreaktionen zu veranlassen.

Vertragsrechtliche Überlegungen treten in den Hintergrund. Dies ist vermutlich der schwierigste und zugleich wichtigste Wechsel im Denken. Ein agiles Team folgt der Maxime, dass noch niemals ein Projekt vor Gericht zum Erfolg geführt wurde. Vielmehr werden beide Parteien als vertrauenswürdige Partner angesehen, die daran interessiert sind, das Projekt zum Erfolg zu bringen.

Win-Win-Szenarien

So platt ausgedrückt ist das natürlich eine unrealistische Utopie, die höchstens von Unerfahrenheit zeugt. Beide Seiten haben weitere – legitime – Interessen, die nicht mit den Projektinteressen identisch sind. So ist dem Auftragnehmer daran gelegen, die eigenen Mitarbeiter auszulasten, vielleicht das eigene Engagement zu vergrößern, Teile der Software auch in anderen Projekten einsetzen zu können und gerade bei einem Festpreis den eigenen Gewinn möglichst groß zu halten oder manchmal auch einfach nur den eigenen Ruin abzuwenden. Der Auftragnehmer ist im Gegenzug daran interessiert, die eigenen Kosten möglichst niedrig zu halten, das Know-how, das in das System geflossen ist, Mitbewerbern nicht zugänglich zu machen und die Software pünktlich und reibungslos in Produktion setzen zu können. Noch schwieriger wird die Situation durch persönliche Interessen der einzelnen Projektbeteiligten.

Einige dieser Interessen stehen miteinander im Konflikt, was nicht automatisch bedeutet, dass sie nicht legitim wären. Wie in jeder guten Geschäftsbeziehung erkennt man auch in einem agilen Projekt an, dass der Partner diese Interessen verfolgt. Wo es zu Konflikten mit den Projektzielen kommt, werden diese offen angesprochen und es wird gemeinsam nach einer konstruktiven Lösung gesucht. Eine solche Lösung sollte eine klassische *Win-Win*-Lösung sein, also eine Situation, in der beide Parteien gewonnen haben. Steht zum Beispiel das Interesse des Auftragnehmers, Software wieder zu verwenden, im Konflikt mit dem Interesse des Auftraggebers auf Schutz seines Know-hows, so sind verschiedene Lösungen denkbar. Der Auftraggeber könnte am Erlös einer Wiederverwendung beteiligt werden oder der Auftragnehmer könnte auf die Wiederverwendung verzichten, dafür aber vom Auftraggeber bei der Vermarktung seines Know-hows unterstützt werden. ▶

In den letzten Jahren leider wieder in Mode gekommene Versuche, den Verhandlungspartner „über den Tisch zu ziehen“, also den eigenen Vorteil auf Kosten des Geschäftspartners zu maximieren, werden in agilen Projekten als schädlich für das Projekt und damit auch als schädlich für beide Seiten angesehen. Dies entspricht auch aktuellen Forschungen, die zeigen, dass sich kooperatives Verhalten langfristig für beide Parteien auszahlt, während egoistisches Verhalten nur kurzfristig Erfolge erzielt. Langfristig verliert vor allem der vermeintlich für seinen Vorteil handelnde Partner. Wer seine Auftraggeber oder Lieferanten erpresst, wird vielleicht kurzfristig seinen Gewinn vergrößern, langfristig zerstört aber ein solches Gebaren eine gute Geschäftsbeziehung, was für alle Seiten hohe Kosten verursacht.

Spielen, um zu gewinnen

Das bedeutet natürlich nicht, dass man auf eine vertragliche Absicherung der Zusammenarbeit gänzlich verzichtet. Beide Seiten müssen sich lediglich darüber im Klaren sein, was ein Vertrag in einem Softwareprojekt leisten kann und was nicht. Ein Vertrag kann helfen Missverständnisse über die Aufgabenstellung oder die Konditionen zu vermeiden. Ein Vertrag kann aber nicht erzwingen, dass ein Softwareprojekt erfolgreich wird. Schäden, die durch gescheiterte Projekte verursacht werden, sind oft in einer Größenordnung, die nicht mehr eingetrieben werden kann, weil der Auftragnehmer schlichtweg nicht über das Kapital verfügt und ein Konkurs des Auftragnehmers bestenfalls die Wartungsfrage offen lässt.

Ein agiles Projekt versucht, solche Schäden nicht über vertragsrechtliche Daumenschrauben abzuwälzen, sondern setzt wesentlich früher an: Durch die Auslieferung lauffähiger Software in kurzen Inkrementen von zwei bis sechs Wochen soll die Höhe des möglichen Schadens von vorne herein minimiert werden. Bereits nach kurzer Projektlaufzeit kann abgeschätzt werden, ob das Projekt den richtigen Weg nimmt. Ein „Totalschaden“ in der Form, dass man nach vielen Monaten oder gar Jahren Entwicklungszeit bemerkt, dass gar nichts geht, kann praktisch ausgeschlossen werden. In agilen Projekten wird also nicht gefragt „Wie kann ich die Verantwortung für ein eventuelles Scheitern möglichst bereits zu Anfang vertraglich verteilen?“, sondern „Wie setze ich das Projekt auf, um den maximalen Nutzen zu bekommen?“. „Spielen um zu gewinnen“ ist eine andere Herangehensweise, als „Spielen, um nicht zu verlieren“.

Worin der Unterschied zwischen diesen beiden Strategien liegt, lässt sich an einem Projekt verdeutlichen, das ich vor einiger Zeit bei einem Kunden beobachten konnte:

In einem Festpreisprojekt erging der Auftrag, ein bestehendes Softwareprodukt an eine innovative Komponente anzukoppeln. Das Projekt war auf zehn Bearbeiterjahre geschätzt worden, die Ankopplung war sowohl technisches als auch fachliches Neuland und der Auftraggeber versprach sich von ihr einen erheblichen Wettbewerbsvorteil. Um das Risiko vermeintlich zu minimieren, wurde eine umfangreiche Spezifikation erarbeitet, in der unter anderem die genauen Versionen der beteiligten Komponenten festgelegt wurden. Nach etwa sechs Monaten Entwicklungszeit stellte sich allerdings heraus, dass für das Projekt einige Features notwendig waren, die in den vertraglich vereinbarten Versionen noch nicht zur Verfügung standen, wohl aber in den jeweils neuesten Versionen. Rein projektechnisch wäre ein Versionswechsel durchaus zu vertreten gewesen und alle Beteiligten waren sich sowohl in der Einschätzung der Notwendigkeit als auch der Machbarkeit eines solchen Wechsels einig. Dies hätte aber eine Änderung des von den jeweiligen Rechtsabteilungen aufwändig ausgehandelten Vertrags zur Folge gehabt. Beide Seiten hatten Angst davor, dass die andere Seite bei einer solchen Änderung versuchen würde, inzwischen aus anderen Gründen aufgetretene Probleme jeweils auf den anderen abzuwälzen, weshalb die Verträge nicht neu verhandelt wurden und die notwendige Änderung unterblieb. Nach einem Jahr war das Projekt so verfahren, dass es gestoppt wurde und mit flexibleren Vereinbarungen und neuen Projektleitern neu gestartet wurde. Wenige Monate später konnte die neue Software mit großem Erfolg in Betrieb genommen werden, nicht zuletzt wegen der mit den neuen Versionen deutlich verbesserten Leistungsfähigkeit der Komponenten.

Kultureller Wandel

Der erste Anlauf für dieses Projekt ist gescheitert, weil die Beteiligten Angst davor hatten Risiken einzugehen. Der Vertrag war als (scheinbar) stabile Plattform für das Projekt verstanden worden, die die Projektleiter beider Seiten nicht verlassen wollten. Ein gezielter Verstoß gegen den Vertrag oder eine kurze schriftliche Ergänzung hätten das Problem gelöst. Statt dessen wurde „Schwarzer Peter“ gespielt, keiner der Beteiligten wollte der erste sein, der klar sagt, dass man bei Abschluss des Vertrags einen Fehler gemacht hatte. Eine Kultur, in der Projekte entsprechend dem agi-

len Verständnis erfolgreich sein sollen, braucht also mindestens zwei Voraussetzungen: Risikobereitschaft muss belohnt werden, zumindest solange sie verantwortungsbewusst gelebt wird, und Fehler müssen als ganz normaler Bestandteil der Projektarbeit gesehen werden, nicht als Versagen, das mit negativen Konsequenzen verbunden ist.

Es geht dabei nicht darum, einer leichtsinnigen, unprofessionellen Arbeitsweise das Wort zu reden, sondern zu verstehen, dass es weder wirtschaftlich sinnvoll noch möglich ist, jede Entscheidung in einem Projekt so abzusichern, dass keine Fehler gemacht werden. Aus der noch immer verbreiteten Vorstellung, man könne durch eine entsprechend ausführliche Analysearbeit alle Fehler vermeiden, spricht eher die historische Verwandtschaft zwischen der akademischen Informatik und der Mathematik als die Erfahrung industrieller Softwareentwicklung. Agile Entwicklung sieht den Erfolg eines Projekts nicht darin, möglichst wenig Fehler zu machen, sondern darin, auf die ohnehin unvermeidbaren Fehlern möglichst professionell zu reagieren. Belohnt wird nicht das Team, das mit einem oft unwirtschaftlichen Aufwand versucht, jeden Fehler zu vermeiden, sondern das Team, das am schnellsten auf auftretende Probleme reagieren kann. Dazu gehört natürlich auch, Fehlersituationen zu analysieren und Maßnahmen zu ergreifen, solche Probleme in Zukunft zu vermeiden.

In vielen Organisationen ist dies ein kultureller Wandel, weil all zu oft noch die Suche nach einem Verantwortlichen für ein Problem die Suche nach einer Lösung verdrängt – insbesondere wenn es darum geht, ob die „Schuld“ innerhalb oder außerhalb der eigenen Organisation zu suchen ist. Nach agilem Verständnis ist aber nicht derjenige „schuld“, der etwas übersehen oder falsch entschieden hat, sondern derjenige, der keine ausreichende Flexibilität zeigt, notwendige Veränderungen vorzunehmen oder zu genehmigen.

Veränderungen vorzunehmen bedeutet aber Entscheidungsfähigkeit und das wiederum bedeutet Risikobereitschaft. Wer Angst davor hat, einen Fehler zu machen, wird auch keine Entscheidungen treffen. In vielen Projekten ist aber genau das der größte Fehler, den man machen kann. Organisationen, die ihren Mitarbeitern viel Freiraum lassen, und wo diese im Gegenzug viel Verantwortung übernehmen, sind hier wesentlich besser gerüstet als Organisationen mit langen Entscheidungswegen und umfangreicher formaler Absicherungskultur.

Kulturelle Kompatibilität

Am einfachsten sind diese Prinzipien natürlich umzusetzen, wenn sowohl Auftraggeber als auch Auftragnehmer eine risikobereite, pragmatische und professionelle Kultur pflegen. In solchen Konstellationen kann man sich auf die Umsetzung der agilen Techniken konzentrieren, ohne mit „politischem Sperrfeuer“ rechnen zu müssen. Umgekehrt ist die Situation auch klar, wenn keine der beiden Parteien über diese Voraussetzungen verfügt: Hier kann agiles Arbeiten nicht funktionieren und die Beteiligten werden bei zweifellos zuvor stattfindender eingehender Analyse zu dem Ergebnis kommen, dass das Risiko eines neuen und schwerer vorher zu sagenden Verfahrens zu hoch ist. Für das verringerte Risiko zahlen solche Organisationen mit entgangenen Gewinnchancen: Innovative Systeme, die erhebliche Wettbewerbsvorteile bieten, sind in einer solchen Kultur nur schwer umzusetzen und mit einem hohen Risiko des Scheiterns versehen. Solche Organisationen werden dann überleben, wenn Innovation für sie kein wesentlicher Bestandteil ihrer Strategie ist.

Relativ eindeutig ist auch eine Situation, in der der Auftragnehmer eher über eine Absicherungskultur verfügt, während die Organisation des Auftraggebers eher pragmatisch geprägt ist. In solchen Fällen dürften sich die Vertragsverhandlungen so schwierig gestalten, dass eine Zusammenarbeit vermutlich kaum zustande kommt. Sollten die Parteien trotz aller Widrigkeiten dennoch zusammen kommen, wird sich der Auftragnehmer vermutlich bald über einen Mangel an Vorgaben beklagen und versuchen, die Verantwortung des Auftraggebers für sich daraus eventuell ergebende Probleme durch entsprechende Notizen festzuschreiben. Gelingt es in einer solchen Situation nicht, den Auftragnehmer zum Umdenken zu bewegen und die dafür notwendigen Techniken zu implementieren, so ist der Auftraggeber in der Regel gut beraten, sich für diese Aufgabe einen anderen Lieferanten zu suchen. Ist das nicht möglich, muss abgewogen werden, ob das Projekt in einer solchen Atmosphäre überhaupt erfolgreich durchgeführt werden kann oder nicht besser abgebrochen werden sollte. Eine Anpassung an die Absicherungskultur des Auftragnehmers erscheint meist unumgänglich, entscheidet man sich dafür, das Projekt unter diesen Bedingungen weiter zu führen.

Mehr Optionen stehen bei einer Konstellation zur Verfügung, in der der Auftragnehmer einen agilen Arbeitsstil

pflegt, während der Auftraggeber um Absicherung bemüht ist. Hier muss der Auftragnehmer zunächst entscheiden, ob er bereit ist, das Risiko eines solchen Spagats einzugehen oder den dabei absehbaren Konflikten lieber aus dem Weg geht – ein guter Test für die eigene Kultur. Kommt der Auftragnehmer zu dem Schluss, dass die pragmatische Vorgehensweise Vorteile für das Projekt bietet, muss er sich darüber im Klaren sein, dass er im Falle von Problemen mit wenig Verständnis des Auftraggebers rechnen kann. Zudem wird der Auftraggeber wenig Bereitschaft zeigen, die für agile Entwicklung eigentlich notwendige enge Kooperation aufzubauen. Zu fest ist oft die Vorstellung eines Werkvertrags in den Köpfen des Auftraggebers verhaftet: „Wir haben ein Stück funktionsfähige Software entsprechend den beigefügten Anforderungen bestellt und erwarten die termingerechte Auslieferung frei Haus.“

Die Lösung dieses Konflikts sollte Informatikern vertraut sein: Das Projekt erhält eine „Schnittstelle“, die das agile Arbeiten im Inneren abschirmt und dem Auftraggeber die geforderten Gremien und Mechanismen zur Verfügung stellt. Ist ein Auftraggeber nicht bereit, kurzfristig verfügbare Ansprechpartner zu benennen, so muss Ersatz geschaffen werden: Ein Schnittstellenteam spielt dann die Rolle des Kunden für das Entwicklungsteam. Das Projekt benötigt also Fachexperten, die den Auftraggeber und dessen Markt gut genug kennen, um die Fragen des Projektteams beantworten zu können. Dies ist eine heikle Gratwanderung, die deutlich schwieriger ist als ein kooperatives Verhältnis zum Auftraggeber und auch ein wesentlich größeres Risiko bedeutet. So kann es passieren, dass diese Fachexperten so oft zum Kunden zitiert werden, um Konzeptarbeiten zu übernehmen, dass sie dem eigenen Team nicht mehr zur Verfügung stehen – eine sehr gefährliche Situation für das ganze Projekt.

Es ist wenig ratsam, auf Dauer eine solche Konstellation ohne Wissen des Auftraggebers als „U-Boot“ zu leben. Die Abstimmungs- und Verständnisprobleme stellen ein zu großes Risiko dar. Durchaus sinnvoll kann es aber sein, eine solches U-Boot zunächst zur Überbrückung zu nutzen, bis der Auftraggeber Vertrauen in die neue Arbeitsweise aufgebaut hat. Häufig kann man offiziell zunächst einen „Prototyp“ starten, der dann dem Auftraggeber als produktionsfähige Software übergeben wird. In dieser Konstellation

arbeitet das „Schnittstellen-Team“ mit einem klassischen Wasserfallansatz, während die Realisierung agilen Prinzipien folgt, in der Hoffnung, dass der Auftraggeber früher oder später einen wirtschaftlichen Vorteil in dem Vorgehen sieht.

Um eine solche riskante Strategie verfolgen zu können, sollte der Auftragnehmer die Motivation für die andere Kultur beim Auftraggeber verstanden haben und sensibel auf die Umstände eingehen, die die Mitarbeiter des Auftraggebers zu ihrem Verhalten bringen. Schlechte Erfahrungen lassen sich oft durch entsprechende positive Erlebnisse abmildern, gesetzliche Vorgaben können sich fatal auswirken, wenn sie nicht beachtet werden. Agile Entwicklung verlangt Fingerspitzengefühl, insbesondere, wenn sie in einem kulturell schwierigen Umfeld betrieben wird. Mehrere agile Projekte sind an genau dieser Hürde gescheitert.

Zusammenfassung

Agile Entwicklung ist deutlich mehr als nur eine Sammlung von *Best Practices* in der Programmierung. Sie greift tief in das Verhältnis zwischen Auftraggeber und -nehmer ein. An Stelle einer vertragsrechtlichen Absicherung tritt eine enge Kooperation, die den wirtschaftlichen Mehrwert für den Auftraggeber in das Zentrum stellt.

Eine solche Konstellation erfordert auf beiden Seiten ein pragmatisches Vorgehen, das sinnvolle Risikobereitschaft mit einer Kultur der Verantwortung und des offenen Umgangs paart. Es ist wichtiger, auf tretende Probleme sinnvoll zu reagieren, als keine Fehler zu machen – ein in der Regel ohnehin sinnloses Unterfangen.

Ist diese Kultur nur beim Auftragnehmer etabliert, stellt ein agiles Vorgehen für den Auftraggeber eine erhebliche Herausforderung dar. Es kann dabei notwendig werden, eine „Kulturschnittstelle“ zu implementieren, an der zwischen den beiden Kulturen vermittelt wird. Langfristig sollte aber der Vorteil für den Auftraggeber so deutlich sein, dass dieser das Vorgehen zumindest akzeptiert und unterstützt. ■

Literatur

- [Bec00] K. Beck, *Extreme Programming Explained – Embrace Change*, Addison-Wesley, 2000
- [Hig00] J.A. Highsmith III, *Adaptive Software Development – A Collaborative Approach to Managing Complex Systems*, Dorset House 2000
- [Pop03] M. Poppendieck, T. Poppendieck, *Lean Software Development – An Agile Toolkit*, Addison-Wesley 2003