

ADNOVUM

NOTITIA

BEMERKENSWERTES VON UND ÜBER ADNOVUM

HERBST 2011, HEFT NR. 21



AGILE SOFTWARE-ENTWICKLUNG



Inhalt

REALITÄTSCHECK FÜR DAS AGILE MANIFEST

Zehn Jahre danach: Einsichten und Aussichten 3

VON SPRINT ZU SPRINT

Agil ans Ziel 7

LET'S SLIDE

Software für eine Touch-Screen-Armbanduhr:
ein Fall für AdNovum 10

POP MEETS CLASSIC

Scrum im grossen Kontext 15

Liebe Leserin, Lieber Leser

Agile Software-Entwicklung – was heisst das? Entwickeln wir nur dann agil, wenn wir es mit einer etablierten Methode wie Scrum tun? Oder reicht es, agile Prinzipien gekonnt mit eigenen Erfahrungen und verbreiteten Best Practices zu kombinieren? Bedeutet Agilität Flexibilität? Die Bereitschaft, jederzeit die Richtung zu wechseln, alles umzustossen und neu zu beginnen? Wie viel Agilität erträgt es?

In dieser Notitia ergründen wir, wie wir agile Vorgehensweisen am besten einsetzen, um ihr Potential zu nutzen. Einen Überblick über ihre bisherige und künftige Nutzung geben Bruno Kaiser und Christian Siffert in einer einleitenden Standortbestimmung. Über praktische Erfahrung mit agilen Methoden im Projektkontext erzählen im anschliessenden Interview Andreas Fürer und Thomas Meyhack. An relevanten Aspekten bei der Anwendung kommen etwa die Methodenwahl, das Tooling und die Vertragsform zur Sprache, aber auch die Projektorganisation und das Mindset des Projektteams und des Kunden.

Damit sind auch die Unternehmenskultur und die Beziehungsebene angesprochen: Um Agilität wirkungsvoll einzusetzen, braucht es eingespielte Teams und ein klar abgestecktes Spielfeld mit akzeptierten Regeln und einer Fairness-Kultur. Auf dieser Basis kann man sich in den schnellen und direkten Ballabtausch wagen.

Agilität impliziert bei vielen Unternehmen einen kulturellen Wandel. Zugleich ist sie für den Kunden eine grosse Chance. Er kann in seine Lösung laufend die neusten Anforderungen einbringen und weiss diese schnell umgesetzt.

Ein besonders schönes Beispiel agiler Entwicklung dürfen wir Ihnen im Hintergrundartikel vorstellen: Orsika Labuda und Sándor Bartha aus Budapest und Gérard Roos haben die Desktop-Applikation für eine innovative Touch-Screen-Armbanduhr entwickelt und berichten darüber aus erster Hand.

Zum Schluss bringen Judith Fothern und Markus Kuhnt von unserer Partnerfirma triple dot in einem flammenden Plädoyer die Sicht des Auftraggebers ein. Sie zeigen, wie man die Rolle des agilen Entwicklungspartners geschickt eingrenzt und ein agiles Projekt in einen grösseren Organisationskontext integriert.

Nun wünsche ich Ihnen viel Lesevergnügen!

Ruedi Wipf

CEO AdNovum Informatik AG

REALITÄTSHECK FÜR DAS AGILE MANIFEST

2001 wurden erstmals die Werte und Prinzipien agiler Software-Entwicklung festgehalten. Was ist daraus geworden? Was hat sich bewährt, was braucht es in Zukunft?

Von Bruno Kaiser und Christian Siffert

Um Software-Entwicklungsprojekte effizienter und flexibler zu gestalten, sind in den 90er Jahren agile Entwicklungsmethoden entstanden. Das Agile Manifest¹ definierte dafür 2001 vier Werte und zwölf Grundprinzipien. Als Erstes wollen wir die vier Werte gestützt auf konkrete Projekterfahrungen kommentieren.

Individuen und Interaktionen höher als Prozesse und Werkzeuge

In komplexen Projekten im Enterprise-Umfeld ist das gemeinsame Verständnis aller Beteiligten vom Sponsor bis zum Tester und Systemadministrator entscheidend. Kontinuierlich müssen Erwartungen abgestimmt werden, damit auf Abweichungen möglichst früh reagiert werden kann. Projekte scheitern gemäss unserer Erfahrung nicht mangels effizienter Prozesse und guter Werkzeuge, sondern wegen zu wenig Kommunikation, Fachwissen und Commitment.

FUNKTIONIERENDE SOFTWARE HAT ERSTE PRIORITÄT.

Funktionierende Software höher als umfassende Dokumentation

Die Erstellung funktionierender Software hat auch bei AdNovum erste Priorität. Dennoch bleibt die Dokumentation auch in agilen Projekten wichtig für die Weiterentwicklung und Wartung des Systems. Sie soll die verschiedenen Abstraktionsbereiche abdecken, gleichzeitig aber auch Redundanzen vermeiden und damit einfach pflegbar bleiben.

Zusammenarbeit mit dem Kunden höher als Vertragsverhandlungen

Dies ist stark abhängig vom Geschäftsmodell und vom Vertragstyp. Bei Fixpreis-Werkverträgen sind und bleiben Vertragsverhandlungen zur klaren Regelung der Verantwortlichkeiten rechtlich, organisatorisch und technisch wichtig. Die Erstellung

einer Spezifikation im Rahmen einer Vorphase kann in diesem Sinne als Teil der Vertragsverhandlungen verstanden werden und die Voraussetzung für die Offerierung eines Fixpreis-Werkvertrags darstellen. Unabhängig vom Vertragsmodell ist die längerfristige und vertrauensvolle Zusammenarbeit für AdNovum ein wichtiges Prinzip, da sie ein lösungsorientiertes Vorgehen erleichtert.

Reagieren auf Veränderungen höher als das Befolgen eines Plans

Das Business unserer Kunden ist schnelllebig und das Umfeld meist komplex. Ein klareres Bild über das Zielgebiet und den sinnvollsten Fokus entsteht oft erst, wenn man sich mit der Lösung befasst. Die Lösung sollte somit auch die Erkenntnisse während des Projektablaufs berücksichtigen. Das Einfrieren der Spezifikation bei Projektbeginn ist zwar aus Budgetgründen nachvollziehbar, kann jedoch durch die damit verbundene Starrheit den Erfolg des Projekts gefährden. Um einen gewissen Spielraum für Richtungsänderungen zu gewinnen, basieren agile Methoden auf möglichst grosser Transparenz der Aktivitäten und Aufwände und früher Lieferung von Teillösungen. Damit wird der Kunde kontinuierlich miteinbezogen und Planänderungen lassen sich rasch umsetzen. Dies erfordert aber die Bereitschaft des Kunden für eine solche aktive Rolle bereits während des Projekts und ein entsprechendes Controlling, um geänderte Rahmenbedingungen fair zu kompensieren. Schliesslich sind der Flexibilität auch bei agilen Methoden klare Grenzen gesetzt durch die Länge der Sprints und das Projektbudget. Im Übrigen kann auch in traditionellen Ansätzen bei genügendem Vertrauen mittels Kulanz oder Absprachen das Projektziel an geänderte Kundenbedürfnisse angepasst werden.

Weitere Inspiration und Anstösse zur Reflexion der Software-Entwicklung geben die zwölf Prinzipien. Nachfolgend unsere wichtigsten Erkenntnisse dazu.

Frühe und kontinuierliche Lieferungen

Mit der frühen Lieferung von Teillösungen und dem kontinuierlichen Ausbau in Folgelieferungen lassen sich Projektrisiken reduzieren und die Lösungen schrittweise optimieren. Ein produk-

¹ <http://agilemanifesto.org/iso/de/>

tiver Release verliert so viel von seiner bedrohlichen Kraft und wird schlicht zum nächsten Schritt, der jederzeit zu schaffen ist. Das Entwicklungstempo wird damit gleichmässiger und nachhaltiger. Die kurzen Iterationen zwingen das Projektteam zur Fokussierung aufs absolut Wesentliche und tragen so entscheidend zum Projekterfolg bei.

OFT IST SOGAR DIE LIEFERUNG EINES LEEREN PACKAGES SINNVOLL.

Für AdNovum seit langem selbstverständlich, ist dieses Prinzip in den letzten Jahren noch bedeutender geworden, da der Integrationsbedarf durch die gesunkene Fertigungstiefe auch im Informatik-Projektgeschäft markant gestiegen ist. Oft ist sogar bereits die Lieferung eines leeren Packages sinnvoll, z. B. für das Einüben von Delivery- und Installationsprozessen. Zusätzlich wird die tägliche Lieferbereitschaft bei AdNovum durch einen voll integrierten Nightly-Build-Prozess unterstützt.

Kanban und Kaizen²

In der Software-Entwicklung bezeichnet Kanban ein Vorgehen, das durch Begrenzung paralleler Arbeiten die Durchlaufzeiten verkürzt und Probleme schnell sichtbar macht. Die Wertschöpfungskette mit den einzelnen Prozessschritten (Anforderungsdefinition, Programmierung, Dokumentation, Test, Inbetriebnahme) wird in einem Kanban-Board visualisiert. Darauf durchwandern die einzelnen Anforderungen (Tasks) als Tickets die für die Prozessschritte zuständigen Stationen. Indem man die Anzahl gleichzeitig bearbeiteter Tickets – den Work in Process (WiP) – pro Station limitiert, wird ein Pull-Prinzip implementiert. Erst wenn ein Ticket erledigt ist, wird das nächste bei der Vorgängerstation abgeholt. So werden Engpässe (Bottlenecks) schnell sichtbar. Dies ermöglicht es, gezielte Massnahmen für einen gleichmässigen Fluss (Flow) zu ergreifen: z.B. Ticketlimiten oder Mitarbeiterzahl für einzelne Stationen ändern, Puffer einführen oder technische Probleme beheben. Kanban bietet somit eine ideale Grundlage, um das Prinzip der kontinuierlichen Verbesserung (Kaizen) zu leben.

Scrum²

Scrum ist eine Methode zur Entwicklung von Produkten hoher Komplexität. Um diese zu reduzieren, werden Projektfortschritt und -hindernisse täglich aufgelistet, Funktionalitäten regelmässig geliefert und beurteilt und Anforderungen laufend bewertet und ggf. angepasst. Alle Aktivitäten sind zeitlich fest beschränkt («timeboxed»). Mehr zu Scrum im Interview Seite 7.

² nach Wikipedia, gekürzt

Persönlicher und regelmässiger Kontakt

Persönliche Bekanntschaft und regelmässiger Kontakt zwischen Fachexperten und Entwicklern und im Entwicklungsteam sind wichtig. Bei AdNovum pflegen wir dazu Kurzvisiten in Budapest für Kickoffs bei Projektstart und für Wrapups am Schluss, bei längeren Projekten auch zwischendurch. Mit diesem Vertrauensaufbau können die Informationen dazwischen auch per Telefon, Videochat und Screensharing ausgetauscht werden. Der direkte Kontakt darf jedoch nicht dazu führen, dass jegliche Dokumente zu «Überfluss» erklärt werden. Gerade für langlebige Enterprise-Software ist konserviertes Wissen unabdingbar.

Tooling

Ein wesentlicher Beitrag der agilen Methoden ist das neue Instrumentarium, das eine genauere Sicht auf den Fortschrittsgrad erlaubt, z. B. JIRA, das AdNovum konsequent auch für traditionelle Projekte einsetzt. Das Tooling erfordert zwingend das Herunterbrechen von Arbeitsaufträgen auf eine verdaubare Grösse und Form (1-3 PT) und erleichtert durch die Konkretisierung die erfolgreiche Projektabwicklung.

Reduce to the max

Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist nach wie vor schwierig zu erreichen, Software-Entwickler suchen noch allzu oft die perfekte Implementierung. Auf der optimalen Umsetzung im Rahmen der vorgegebenen Restriktionen wie Zeit, Geld, Ressourcen, Knowhow und Technologie zu bestehen, sehen wir als wichtige Funktion des Projektleiters, aber auch des Kunden, der den Aufwand ja schliesslich bezahlt.

SOFTWARE-ENTWICKLER SUCHEN ALLZU OFT DIE PERFEKTE IMPLEMENTIERUNG.

Hürden und Grenzen

Die Werte und Prinzipien des Agilen Manifests sind heute in vielen Projekten eine Selbstverständlichkeit, werden aber kaum je in Reinform und exklusiv umgesetzt. Meist werden Muster aus verschiedenen Projektmethoden gemischt, z. T. auch mit Selbsterfundenem, das sich bewährt hat.

Aktuell stossen die agilen Methoden an folgende Grenzen:

- Limitierter Fokus auf Produktentwicklung: Mit seinen grundlegenden Aussagen ist das Agile Manifest nach wie vor hilfreich und dürfte es bis auf weiteres bleiben. Scrum als eine konkrete Methode zu seiner Umsetzung ist dagegen stark auf die Implementierung eines Produkts zugeschnitten. AdNovum als Projektfirma hat aber andere Bedürf-



Agilität bietet viel Gesprächsstoff: Christian Siffert und Bruno Kaiser.

nisse und bietet zudem auch Wartung und Support, Integration, Business-Analyse und Beratung an. Dafür eignen sich Konzepte aus Methoden wie z. B. Kanban und Kaizen, die mit ihrem weiteren Fokus z. B. auch die Wartung und die Ent-sorgung der Software umfassen.

- **Fehlende Wiederverwendbarkeit:**

Agile Methoden sind quasi ein Kochbuch mit Rezepten, die einem Projekt als Ideen für die Durchführung dienen können. Jedes Projekt ist anders. Statt einer dogmatischen Anwendung von agilen Methoden ist deshalb eine pragmatische Umsetzung im konkreten Projektkontext gefordert. Wichtig ist die Kenntnis von bewährten Verhaltensmustern wie Pair Programming, testgetriebene Entwicklung mit Unit Tests, Daily Scrums und Sprint-Phasen. Diese Muster sind jedoch als Teil einer bestimmten agilen Methode noch nicht ohne weiteres auf andere agile Methoden anwendbar, sie werden auch nicht

GEFRAGT IST EIN TAILORING ÜBER VERSCHIEDENE METHODEN HINWEG.

von unterschiedlichen Werkzeugen unterstützt. Die meisten in den Projekten gelebten Methoden sind heute, wenn auch in den Grundwerten ähnlich, durch Massschneidung (Tailoring) stückweise einzigartig. Gerade mit Kanban wird sehr kontextsensitiv am Entwicklungsprozess optimiert. Die Methoden lassen sich somit ohne Kontext nicht kopieren. Ein Mitarbeiter muss sich in jedem Projektumfeld neu einarbeiten und kann nicht überall die gleichen Metriken anwenden.

- **Schwierige Abbildung in Verträgen:**

Software-Entwicklung wird juristisch meist in Form eines Werkvertrags oder Auftrags abgebildet. Ein Fixpreis-Werkvertrag passt gut auf ein traditionelles Wasserfall-Vorgehen mit eingefrorener Spezifikation. Die in agilen Projekten gewünschte Flexibilität lässt sich jedoch nur schlecht in einen Werkvertrag überführen. In der Praxis werden deshalb agile Projekte meist als Aufträge mit Entschädigung nach geleistetem Aufwand (Time & Material) formuliert. Dadurch verliert der Kunde jedoch die Budgetsicherheit eines Fixpreis-Vertrags und die damit verbundene Risikoteilung. Ein Kompromiss sind kombinierte Vertragswerke, die einzelne Phasen nach Aufwand oder als Fixpreis-Werkverträge abbilden. AdNovum hat solche agilen Vertragstypen erarbeitet, sie sind jedoch bisher noch nicht üblich und werden entsprechend wenig verwendet.

- **Unterschiedliche Verständnisse von Agilität:**

Agile Methoden sind eine Erfindung der IT. Agilität wird von Personen in verschiedenen Rollen jedoch unterschiedlich verstanden. Business-Abteilungen beispielsweise wünschen sich heute mehr Flexibilität und geringere Kosten von ihrer IT-Ab-

teilung und sind deswegen schnell von agilen Methoden begeistert. In der konkreten Umsetzung stösst die Anwendbarkeit dann aber rasch an Grenzen wegen vertraglichen und administrativen Regelungen oder Governance und Regulatorien.

Agil in die Zukunft

Der Kunde fordert bei der heutigen Komplexität und Dynamik des Geschäfts zu Recht mehr Transparenz und schnellere Resultate, auch in Fixpreis-Projekten. Wir Software-Ingenieure müssen also Ansätze finden, um die mit agilen Methoden angebotene Flexibilität im Abwickeln von Software-Projekten gewinnbringend umzusetzen, die Anwendung von agilen Methoden aber gleichzeitig zu vereinfachen und zu standardisieren und dabei für jedes Projekt und seine Rahmenbedingungen die richtigen Methoden und Mittel zu wählen.

Das Massschneidern auf den Projektkontext beschränkt sich heute jeweils auf die Elemente einer bestimmten agilen Methode. Gefragt ist aber ein Tailoring über verschiedene Methoden hinweg. Dafür müssen die Methoden in generische, methodenunabhängige Muster aufgebrochen werden. Liesse sich ein solches Vorgehen industrieweit standardisieren, könnte man dessen Anwendbarkeit auch auf andere Gebiete erweitern und beispielsweise die Verträge aufgrund der angewendeten Muster zu einem grossen Teil automatisch generieren. Erste Lösungsansätze im Hinblick auf eine Standardisierung und Vereinheitlichung von agilen Verhaltensmustern entstehen gegenwärtig z. B. in SEMAT (Software Engineering Method and Theory)³ oder bei OMG (Object Management Group)⁴. ■

Bruno Kaiser

Bruno Kaiser, lic. oec. publ. IMBL HSG, war seit 1994 an der Entwicklung der AdNovum-Produkte beteiligt, zuerst auf Kundenseite, ab 1999 bei der AdNovum USA. Seit seiner Rückkehr in die Schweiz 2002 ist er als Mitglied der GL und Projektleiter für zahlreiche AdNovum-Aktivitäten zuständig, so unter anderem auch für die agilen Projekte Sense und CLT Japa. Agilität lebt er auch privat mit seiner Familie in den Bergen und beim Reisen.

Christian Siffert

Christian Siffert verantwortet seit dem Sommer 2006 den Application-Engineering-Bereich der AdNovum. Der ETH-Informatiker ist seit 1999 für die AdNovum aktiv. Als Entwickler und technischer Projektleiter sammelte er Erfahrung in zahlreichen Projekten und bringt insbesondere die agile Sichtweise mit ein. Zum Ausgleich lüftet er seine Gedanken beim Mountainbiken im Jura-Gebirgszug aus oder lässt seinen Modellhelikopter in den Himmel steigen.

³ www.semat.org

⁴ RFP «A Foundation for the Agile Creation and Enactment of Software Engineering Methods», <http://doc.omg.org/ad/2011-6-26>

VON SPRINT ZU SPRINT

Von der User Story zur Business Solution. Andreas Fürer und Thomas Meyhack über Agilität in der Software-Entwicklung.

Entwickelt AdNovum agil?

A.F.: Diese Frage kann man nicht einfach mit Ja oder Nein beantworten. Kann man bereits von agiler Entwicklung sprechen, wenn die Zusammenarbeit in einer Entwicklungsfirma auf den Werten des Agilen Manifests beruht? Wenn ja, entwickelt AdNovum schon lange agil, denn AdNovum ist mit agilen Werten gross geworden. Zusammenarbeit zählt bei uns mehr als das Festhalten an Prozessen, und funktionierende Software und Qualität sind uns wichtig. Wir suchen laufend den Dialog mit unseren Kunden und reagieren wo immer möglich flexibel auf veränderte Anforderungen.

T.M.: Das stimmt, wir setzen auch in der Entwicklung seit langem agile Methoden wie Pair Programming, Code Reviews und Rapid Prototyping ein. Und unser Entwicklungsprozess umfasst agile Aspekte wie kurze Entwicklungszyklen, regelmässige Builds und automatisiertes Testen mit Unit Tests.

ADNOVUM IST MIT AGILEN WERTEN GROSS GEWORDEN.

A.F.: Und trotzdem würde ich nicht sagen, AdNovum entwickle schon lange agil, denn bis jetzt war die Agilität eher einseitig. Das bedeutet, wir haben bei der Entwicklung zwar agile Aspekte eingesetzt, der Vertragsrahmen war aber statisch. Das heisst, wir entwickelten unter klassischen Festpreisverträgen mit Termin und Pflichtenheft. Bei agiler Entwicklung werden die Anforderungen jedoch erst im Laufe des Projekts spezifiziert, der Scope kann also nicht bereits Teil des Vertrags sein. Damit das Potential von Agilität voll ausgeschöpft werden kann, braucht es die Bereitschaft des Kunden, sich darauf einzulassen, die Anforderungen erst im Laufe des Projekts auszuformulieren.

Wie sieht denn ein Vertrag für agile Software-Entwicklung aus?

T.M.: Der Vertrag soll den flexiblen Umgang mit Anforderungen erlauben und doch einen verbindlichen Rahmen bieten. Dies wird erreicht durch einen Rahmenvertrag, in dem das Ziel definiert und das Vorgehen geregelt wird. Dazu kommt eine Reihe von kleinen Verträgen für die jeweils nächste Umsetzungsetappe mit Termin und Preis, aber ohne exakten Scope. Das Entwicklungsteam legt den Scope der jeweils nächsten Umsetzungsetappe fest. Dabei

werden die Anforderungen Brocken für Brocken vom Team geschätzt, vom Product Owner priorisiert und danach spezifiziert und umgesetzt. Im Fokus stehen dabei immer die Tasks, die gerade anstehen.

Die Spezifikation entsteht also erst während der Entwicklung. Ist das nicht etwas riskant?

A.F.: Im Gegenteil, es ist weniger riskant. Wenn wir wie beim klassischen Vorgehen drei Monate lang Papier produzieren, bevor wir mit der Umsetzung beginnen, spezifizieren wir unter Umständen Funktionen, die es am Ende gar nicht braucht. Bei einem agilen Vorgehen spezifizieren wir entlang von User Stories immer nur den jeweils nächsten Schritt. User Stories beschreiben in einem Satz etwas, das der Benutzer machen möchte, wie etwa sich in die Applikation einloggen.

Bei AdNovum setzen wir Scrum ein. Weshalb Scrum?

A.F.: Welches Vorgehen man wählt, ist nicht so wichtig. Wir setzen Scrum ein, weil Scrum gut etabliert ist. Scrum definiert einen agilen Entwicklungsprozess mit einem Set von Begriffen. Das vereinfacht die Kommunikation. Und es lässt sich besser verkaufen als ein AdNovum-eigener agiler Prozess, den niemand kennt.

Wie sieht ein Scrum-Team aus?

T.M.: Im Scrum gibt es drei Rollen, den Product Owner, den Scrum Master und das Entwicklungsteam. Das Team besteht aus allen Leuten, die in die Entwicklung involviert sind, inklusive Testern und Business-Analysten. Wenn ein Scrum-Team also aus sieben Leuten besteht, dann sind das fünf Entwickler, nicht etwa sieben. Grösser sollte das Team nicht sein, sonst wird die Abstimmung zu aufwändig und das Team muss aufgeteilt werden.

Was macht der Product Owner?

A.F.: Der Product Owner vertritt den Kunden. Er gibt das gemeinsame Ziel vor, indem er die User Stories im sogenannten Product Backlog priorisiert, und er prüft, welche Features den grössten geschäftlichen Nutzen bringen. Er ist bei den Sprint-Planungsmeetings dabei und muss in der Lage sein, fachliche Fragen der Entwickler zu den User Stories zu beantworten.

T.M.: Wichtig ist auch, dass der Product Owner versteht, welche technischen Auswirkungen Entscheidungen haben. Und dass er



Agilität ist ihr Daily Business: Andreas Fürer ...

die Kompetenz hat, Entscheidungen zum Funktionsumfang zu fällen und dem Management gegenüber zu vertreten.

Der Product Owner definiert die Ziele, das Team spezifiziert und entwickelt. Was macht denn der Scrum Master?

A.F.: Der Scrum Master ist der Moderator. Streng nach Scrum ist er nur für die organisatorischen Aspekte verantwortlich. Er sorgt dafür, dass das Team effizient arbeiten kann, ist aber nicht Teil des Teams. Er ist nicht in die Entwicklung involviert und trifft keine technischen Entscheidungen. Wenn der Scrum Master aber auch bei der Entwicklungsfirma arbeitet, ist es nach unserer Erfahrung speziell in kleinen Teams effizienter, wenn er auch mitentwickelt. Allein mit organisatorischen Tasks ist er nicht ausgelastet, es sei denn, die Organisation ist sehr kompliziert.

DIE ENGE ZUSAMMENARBEIT VERLANGT SOZIALKOMPETENZ UND DIALOGBEREITSCHAFT.

T.M.: Das kann ich bestätigen. In einem Projekt hatten wir einen externen Scrum Master. Seine Aufgabe bestand darin, uns die Scrum-Methodik einzuhämmern und Hindernisse in der Kundenkommunikation und der Koordination aus dem Weg zu räumen. Das hat sehr gut funktioniert, das Team hat die Prinzipien von

Scrum schnell verstanden und es gab nicht allzu viele organisatorische Impediments. Der Scrum Master hatte deshalb bald nicht mehr viel zu tun.

Was ist aus eurer Sicht wichtig für die Zusammenarbeit im Team?

T.M.: Die enge Zusammenarbeit verlangt von den Entwicklern hohe Sozialkompetenz und Dialogbereitschaft. Einzelkämpfer tun sich unter Umständen anfänglich schwer damit und empfinden die täglichen Meetings als zu aufwändig. Ein einzelner Task lässt sich isoliert in manchen Fällen ja tatsächlich schneller umsetzen. Aus der Perspektive des Projekts ist das aber nicht unbedingt effizienter. Zudem sorgt ein agiles Vorgehen für einen besseren Knowhow-Transfer. Dabei hilft es, wenn alle Entwickler am selben Ort sind, am besten sogar im selben Büro. Sonst wird die Kommunikation schnell sehr aufwändig.

LEUTE KÖNNEN NICHT EINFACH MITTEN IN EINEM SPRINT AUSGEWECHSELT WERDEN.

A.F.: Ebenfalls wichtig ist, dass auch der Kunde vor Ort präsent ist. Wir haben ein Scrum-Projekt umgesetzt, da waren die Entwickler am selben Ort und haben agil zusammengearbeitet. Aber der Kunde war nicht am selben Ort. Das Team hat dadurch seine Entscheidungen oft nicht verstanden.

T.M.: Wichtig ist auch, dass die Entwicklungsfirma die Stabilität des Entwicklungsteams über die ganze Projektlaufzeit oder zumindest über den jeweils nächsten Sprint hinweg gewährleisten kann. Der Erfolg agiler Entwicklung beruht auf dem Commitment der einzelnen Entwickler und dem Vertrauen, das der Kunde in das Team investiert. Da können Leute nicht einfach mitten in einem Sprint ausgewechselt werden.

Der Aufwand wird vom Entwicklungsteam geschätzt. Wie gut funktioniert das?

T.M.: Am Anfang hat es zum Teil nicht wirklich funktioniert. Die Entwickler waren es nicht gewohnt, eine Schätzung abgeben zu müssen, ohne genauere Informationen zu haben. Dabei geht es wirklich nur um eine erste relative Schätzung im Vergleich zu einer Referenzstory.

A.F.: Diese Erfahrung habe ich auch gemacht. Als aber einmal alle im Team verstanden hatten, was die Idee ist, funktionierte es gut. Die Schätzungen waren erstaunlich präzise und es gab kaum Abweichungen, das heißt, meistens haben alle im Team den Aufwand etwa gleich geschätzt.

T.M.: Ja, auch die Velocity lässt sich mit der Zeit gut einschätzen. Das heißt, das Team weiß relativ genau, wie lange es für einen Task braucht. Allerdings wirken sich Wechsel im Team oder Ferienabwesenheiten stark auf die Velocity aus.

Ist der Product Owner bei der Schätzung dabei?

T.M.: Der Product Owner muss dabei sein, weil er das Detailwissen hat, dass die Entwickler brauchen, um die Stories zu schätzen. Umgekehrt geben die Schätzungen des Teams dem Product Owner Aufschluss über die Komplexität einer Story. Das hilft ihm dabei, Entscheidungen glaubwürdig an das Management zu kommunizieren.

Braucht es spezielle Tools für Scrum?

A.F.: Nach Scrum-Philosophie ist das Tooling Nebensache. Wir könnten theoretisch alles auch auf Papier machen oder in einem Excel Sheet. Allerdings ist das aufwändig und fehleranfällig, nur schon wegen der Versionierung. Aus meiner Sicht ist das Tooling deshalb sehr wichtig, vor allem, wenn nicht alle im selben Büro sitzen.

T.M.: Definitiv. Wir müssen die User Stories, Tasks und das Product Backlog zentral verwalten können. Alles andere bedeutet unnötigen Aufwand.

Welche Tools habt ihr in euren Projekten eingesetzt?

A.F.: In der Regel setzen wir für die agilen Aspekte JIRA Greenhopper ein, ein JIRA-Plug-in. Greenhopper bietet zum Beispiel eine Planning-Ansicht, in der man Stories mittels Drag & Drop priorisieren kann.

T.M.: Je nach Umfeld des Kunden setzen wir auch andere Tools ein, in einem Projekt zum Beispiel Rational Team Concert (RTC) von IBM. Das Tool zeigt unter anderem an, wie viel Arbeit bereits getan ist und was noch fehlt (Burndown Chart). RTC war in die Entwicklungsumgebung integriert, wir konnten also unsere Source-Code-Änderungen direkt an die Tasks linken. Der Scrum Master konnte dadurch jederzeit überprüfen, wie viel Zeit für

einen Task effektiv schon aufgewendet worden war. Und wir konnten die Stories direkt in der Entwicklungsumgebung verwalten. Das ist sehr praktisch.

A.F.: Eine voll integrierte Umgebung bringt allerdings auch Abhängigkeiten und damit Risiken mit sich. Ausschlaggebend für die Wahl des Toolings sind letztlich die Anforderungen des Kunden.

NICHT JEDES PROJEKT IST FÜR EIN AGILES VORGEHEN GEEIGNET.

Wann könnt ihr ein agiles Vorgehen wie Scrum empfehlen?

A.F.: Ein agiles Vorgehen ist nicht in jedem Fall geeignet. Wenn ein Projekt aus einem reinen Umsetzungstask besteht wie etwa ein Migrationsprojekt, bringt der Einsatz eines agilen Vorgehens wie Scrum wenig.

T.M.: Scrum empfehlen wir für Projekte, bei denen der Kunde die Anforderungen an die Software noch nicht komplett kennt. Also für fast alle Projekte von einer gewissen Komplexität. Allerdings muss der Kunde wissen, was Scrum ist, und es auch wirklich wollen. Wenn diese Voraussetzungen erfüllt sind, ist der Einsatz von Scrum sinnvoll und effizient. In meinem aktuellen Projekt zum Beispiel ist der Kunde begeistert und möchte mit Scrum weitermachen. ■

Andreas Fürer

Andreas Fürer, MSc ETH in Computer Science, seit 2007 bei AdNovum. In seiner Rolle als Software Engineer und technischer Projektleiter ist er verantwortlich für die erfolgreiche Lieferung massgeschneiderter Business-Applikationen auf Basis von Java EE, Eclipse Rich Client Platform (RCP) und JavaFX. Wenn er gerade nicht am Planning Poker spielen ist, verbringt er seine Freizeit auch gerne bei einer Runde echtem Texas Hold'em.

Thomas Meyhack

Thomas Meyhack, dipl. Wirtschaftsinformatiker Uni ZH, ist seit 2001 als Software Engineer in diversen Projekten bei der AdNovum tätig. Anfang 2010 begann für ihn das Abenteuer agile Entwicklung mit der besonderen Herausforderung Scrum. Privat gibt er sich gern dem Spiel, dem Genuss und dem Dolcefarniente hin.



... und Thomas Meyhack tauschen Projekterfahrungen aus.

LET'S SLIDE

Die Touch-Screen-Armbanduhr Slyde lässt sich via PC-Applikation konfigurieren und nach Wunsch mit Echtzeit-Animationen aus dem Web bestücken – ein Fall für AdNovum, Cutting-Edge-Technologien und agile Methoden.

Von Orsika Labuda, Sándor Bartha und Gérard Roos

Die rasante technologische Entwicklung macht auch vor so traditionellen Dingen wie der Armbanduhr nicht Halt. Slyde, die innovative Touch-Screen-Armbanduhr der Firma Comme Le Temps (www.commeletemps.com), lässt sich via USB mit dem PC verbinden und via Desktop-Applikation individuell konfigurieren. Ein Online-Webshop-Portal erlaubt es zudem, vorgefertigte «Virtual

DIE UHR WIRD TAKTIL BEDIENT, SIE HAT KEINE MECHANISCHEN BEDIENELEMENTE.

Engines» (Echtzeit-Animationen für die Darstellung von Uhrzeit usw.) für die Uhr einzusehen und herunterzuladen. AdNovum hatte die einmalige Gelegenheit, die Applikation und das Portal für diese High-End-Armbanduhr zu entwickeln.

Hohe Ansprüche, schlankes Budget

Die Desktop-Applikation Slyder musste technisch wie auch konzeptuell hohe Anforderungen erfüllen. So sollte sie möglichst weitgehend aus bereits bestehenden Komponenten zusammengesetzt werden, intuitiv und benutzerfreundlich sein, auf Windows wie auch auf Mac laufen, den hohen Designansprüchen der Uhr entsprechen und nur einen Bruchteil dessen kosten, was manch bekannter Weltkonzern für die Entwicklung seiner Produkte und Software-Lösungen auszugeben bereit ist.

100 % Touch

Die Uhr besteht aus einem weltweit einzigartigen gebogenen Touch-Screen-Display mit hoher Bildauflösung. Sie besitzt keinerlei mechanische Bedienelemente, sondern kommt einzig und allein mit dem taktilen Display aus. Die Bedienung erkennt Streichbewegungen (slide) sowie kurzes und langes Antippen (tap, press). Damit die Bedienelemente einfach erreichbar sind, wurden sie auf einer vertikalen und einer horizontalen Achse angeordnet. Zwischen den verschiedenen Bildschirmmasken für die Uhrenfunktionen wie Zeitanzeige, Zeitzonen, Kalender, Stoppuhr und

Chronograph kann der Benutzer durch Streichbewegungen nach oben und unten hin und her wechseln. Drücken eines Screens aktiviert Spezialfunktionen wie die Auswahl einer Zeitzone oder das Starten und Stoppen der Stoppuhr.

Auf der horizontalen Achse werden benutzereigene Bilder angeordnet. Diese Galerie lässt sich mit Streichbewegungen durchblättern. Ein optionaler Zähler auf den Bildern zeigt die Tage bis zu/seit einem festgelegten Datum an.

Einfach, auch in der Konfiguration

Die Bedienung der Uhr erlaubt nur rudimentäre Einstellungsänderungen wie Zeitzonenwechsel. Die eigentliche Konfiguration erfolgt über einen Computer. Die Uhr wird via USB-Kabel mit dem Computer verbunden, worauf die zugehörige Desktop-Applikation (Slyder) den Inhalt der Uhr grafisch anzeigt und die Elemente in einer Übersicht kreuzartig flach auslegt. Einzelne Elemente können verändert, hinzugefügt oder entfernt werden. Ganze Gruppen von Elementen können zur einfachen Austauschbarkeit als «Engine» abgelegt werden. Sogenannte «Graphic Engines» bestehen aus Elementen mit einem Hintergrundbild und einem «Font» zur Darstellung von Zahlen. Weiter gibt es «Virtual Engines», die aus Elementen bestehen, die zur Anzeige komplette Animationen verwenden. «Virtual Engines» können nicht verändert, sondern nur vorgefertigt installiert werden. Ein Webshop

SLYDER LEGT DIE ELEMENTE AUF DEM DESKTOP KREUZFÖRMIG AUS.

dient als Plattform für solche «Virtual Engines»; Anbieter stellen «Engines» kostenlos oder gegen Entgelt bereit, Uhrenbesitzer laden sie herunter und installieren sie mit der Desktop-Applikation.

Spezifikation: Just in Time

In einer ersten Phase schrieben wir eine Anforderungsspezifikation mitsamt Aspekten zur technischen Machbarkeit – jedoch bewusst noch nicht zu detailliert – und sammelten Use Cases. Wir

skizzierten zusammen mit dem Kunden den Look and Feel und zeigten mögliche Lösungen zu technischen Fragen auf. Damit sollten die Idee und das gemeinsame Verständnis festgehalten werden.

DIE UHR WAR BEI PROJEKTSTART NOCH NICHT FERTIG, DIE ANFORDERUNGEN WURDEN ERST IM LAUFE DES PROJEKTS FESTGELEGT.

Die Einzelheiten und insbesondere die konkreten technischen Lösungen erarbeiteten wir dann Phase für Phase während der gesamten Entwicklungszeit. Dies nicht zuletzt deshalb, weil sich die Geräteherstellung (Uhr) selbst auch noch in Entwicklung befand und sich somit die Anforderungen im Vorfeld nicht bis ins letzte Detail festlegen liessen. Dieses Vorgehen erlaubte es uns, die initiale Spezifikation in nur einem Monat fertigzustellen, ohne dabei den Überblick über den Funktionsumfang zu verlieren.

Angelehnt an Scrum unterteilten wir die Entwicklungsarbeit in regelmässige Schritte (Sprints). Für jeden Monat definierten wir die zu behandelnden Themen und den Zielumfang mit dem Kunden. Die Entwickler waren in die Aufwandschätzung involviert, was ein verstärktes Commitment und somit genauere Schätzungen brachte. Den Projektfortschritt verfolgten wir mit zwei simplen Arbeitswerkzeugen: mit einer Taskliste in Form eines Spreadsheets, auf dem die Tasks über die Monate verteilt und ihre Aufwände monatlich aufsummiert wurden, und einer parallelen Aufstellung der für denselben Zeitraum eingeplanten Manpower. Dank einfacher Differenzrechnung und farblicher Hervorhebung von Diskrepanzen liess sich sofort erkennen, in welchem Monat mehr/weniger Entwickler bzw. weniger/mehr Arbeit einzuplanen war.

Projektfortschritt immer im Blick

Im Anschluss an die Planung eines Monats erfassten wir die Tasks mitsamt Aufwandschätzungen in einem weiteren Tool, dem Ticketing Tool (JIRA). Die Entwickler führten dort täglich Buch über die erfüllten Tasks und die aufgewendete Zeit. So konnten wir in der Hour-Burndown-Chart praktisch in Echtzeit einen Überblick über den Projektfortschritt gewinnen und bei starken Abweichungen rasch und gezielt reagieren.

In kritischen Phasen und insbesondere während des Projektstarts hielten wir täglich kurze Meetings (Daily Scrums) ab, um uns über die Ziele und die einzelnen Arbeiten zu verständigen. Dies half, Doppelspurigkeiten zu vermeiden, förderte den Austausch und ermöglichte es den Entwicklern, den Überblick über das Projekt zu behalten. Diese Meetings reduzierten wir je nach Projektstand und -fortschritt auf einen wöchentlichen Austausch und führten sie bei Bedarf wieder häufiger durch. Wegen der

geografischen Trennung der Entwickler (Zürich und Budapest) fanden die Meetings telefonisch und vereinzelt via Videokonferenz statt.

Erfolgsentscheidend waren zweifellos auch die enge Zusammenarbeit mit dem Kunden und der rege Austausch mit dem Uhrenhersteller selbst. Es galt zu vermeiden, aufgrund einer Spezifikation komplett vom Kunden losgelöst zu entwickeln, um am Ende eine Lösung fernab seiner Vorstellungen zu implementieren. Deshalb haben wir regelmässig Anfang und Mitte Monat mit dem Kunden unsere gemeinsamen Ziele und Fortschritte abgesprochen. So konnte er die Priorisierung der Arbeiten beeinflussen, Änderungswünsche zeitgerecht einbringen und sich über den Projektfortschritt stets genauestens informieren. Das äusserst pragmatische Vorgehen aller involvierten Parteien sowie die kon-

DER EINSATZ VON JAVAFX UND ILLUSTRATOR ERLAUBT EINE ENGE EINBINDUNG DES GRAFIKDESIGNERS.

sequente Absprache und technische Koordination direkt mit Drittlieferanten wie dem Uhrenhersteller haben ebenfalls massgeblich dazu beigetragen, dass aus sieben Monaten Entwicklungszeit zwei perfekt aufeinander abgestimmte Systeme – Desktop-Applikation und Uhr – resultierten.



Die Slyde Watch reagiert auf Streichbewegungen und Antippen.



Entwickelten die Applikation Slider von Budapest aus: Sándor Bartha und Orsika Labuda.

Technologie vom Feinsten

Technologisch begegneten wir den hohen Anforderungen an Grafik und Portierbarkeit auf unterschiedliche Plattformen, pri-

ANFORDERUNG WAR, DIE WELT NICHT ZUM X-TEN MAL NACHZUBAUEN.

mär Windows und Mac, mittels Java und JavaFX. Diese Technologie zusammen mit dem Illustrator-JavaFX-Exporter erlaubt zudem die enge Einbindung des Grafikdesigners in den Entwicklungsprozess. So kann er die einzelnen grafischen Elemente eigenständig im Illustrator zeichnen und im entsprechenden JavaFX-Endfor-

mat abspeichern. Daraufhin können die Entwickler diese Dateien direkt in ihren Code einbauen. Die Kundenanforderung, möglichst viele Open-Source-Routinen zu integrieren und die Welt somit nicht zum x-ten Male nachzubauen, erfüllt die Lösung mit der Einbindung kompletter Open-Source-Programme wie ImageMagick zur Bild- oder mencoder zur Videobearbeitung. Um Software zu integrieren, die der restriktiveren Open-Source-Lizenz GPLv2 oder neuer unterliegt, wird sie als eigenständiges Programm ausgeführt, bei dem der In- und Output ausschließlich in Form von Dateien in gängigen Formaten wie JPEG-Bildern erfolgt.

Webshop: Buy statt Make

Für den Online-Webshop galt ebenfalls wo sinnvoll das Credo «Buy statt Make», um von bereits existierenden Komponenten zu



profitieren und die Aufwände klein zu halten. Als Grundlage für die Umsetzung der projektspezifischen Anforderungen diente der

**DAS AGILE VORGEHEN UND
DIE AKTIVE MITWIRKUNG DES
KUNDEN TRUGEN
WESENTLICH ZUM ERFOLG BEI.**

Open-Source-Webshop ZenCart – aufbauend auf Apache, PHP und MySQL. ZenCart bietet alle zentralen Aspekte einer Webshop-Applikation, von der Registrierung über den Produktkatalog bis zur Online-Kreditkartenzahlung. Verfügbar als Open Source

wurde es gezielt an die eigenen Bedürfnisse angepasst bzw. durch zusätzliche Funktionen erweitert. Da der Kunde über keine eigene IT-Infrastruktur verfügt, musste eine für den Webshop passende Plattform identifiziert werden. Verschiedene Firmen bieten weltweit Hosting-Plattformen an, um genau solche Webshops und Onlineportale remote übers Internet zu betreiben. Dabei wird der Betrieb der Hardware und des Betriebssystems vom Provider sichergestellt, die Installation und Wartung der eigentlichen Software aber vom Lieferanten/Kunden selbst vorgenommen. Bei der Auswahl des Providers wurden Kriterien wie Verfügbarkeit, Betriebskosten, Kompatibilität bzw. Unterstützung für die Webshop-Technologie berücksichtigt.

Das gesamte Vorhaben umfasste alle Aspekte eines Projekts im Rahmen einer Produktentwicklung, von der Spezifikation über



Gérard Roos: Architekt, Projektleiter und Bindeglied zum Schweizer Kunden.

die Realisierung bis hin zum Betrieb. Dabei waren ein agiles Vorgehen und die aktive Mitwirkung des Kunden mit regelmässigen Meetings und kurzen Antwortzeiten wichtig. So konnten wir auf die sich ändernden Anforderungen aus der Geräteentwicklung reagieren und erst im Laufe der Entwicklung reifende Ideen noch aufnehmen. ■

Gérard Roos

Gérard Roos, dipl. Informatik-Ingenieur ETH und seit 2001 bei der AdNovum, hat in den letzten Jahren die technische Umsetzung diverser Web- und Sicherheitsapplikationen, SSO-Portale und Logistiklösungen geleitet. Er ist spezialisiert auf die Umsetzung von Erstreleases umfangreicher Projekte. In der Freizeit tummelt er sich gerne mit der jungen Familie oder hangelt sich möglichst hohe Fels- und Eiwände hoch.

Orsika Labuda

Orsika Labuda, MSc in Mathematics and Computer Science, ELTE Budapest, arbeitet seit Februar 2011 als Senior Java Developer bei AdNovum. Von ihrer langjährigen Erfahrung und Kreativität profitiert das Projekt CLT Japa an breiter Front, vom technischen Detail zur bestmöglichen User Experience. In ihrer Freizeit ist sie gerne auf Reisen und entdeckt Naturschönheiten in aller Welt.

Sándor Bartha

Sándor József Bartha, dipl. Informatikingenieur Sapientia University, arbeitet seit 2007 bei AdNovum. Als Senior Software Engineer engagiert er sich in grossen Business-Applikations- und Integrationsprojekten. In seiner Rolle als Peer für das ungarische Entwicklerteam war er massgeblich an der Entwicklung von CLT Japa beteiligt. In seiner Freizeit spielt er gerne Tennis und geht mit Freunden aus.

POP MEETS CLASSIC

Alle sprechen von Agilität, aber keiner traut sich so richtig. Was theoretisch und in kleineren Projekten sowie Start-ups ganz hervorragend funktioniert, führt in einem klassischen Organisationsumfeld und in einem erweiterten Kontext schnell zu Problemen. triple dot hat es gewagt und zusammen mit AdNovum ein spannendes Grossprojekt zum Erfolg geführt.

Von Judith Fothern und Markus Kuhnt, triple dot Schweiz GmbH



Ausgehend von einer komplizierten und volatilen Anforderungslage bei einem unserer Mandanten wollten wir die Vorteile agiler Entwicklung nutzen und entschieden uns, ein Projekt mit Scrum durchzuführen. Über unseren spezifisch auf agile Projekte ausgerichteten Angebotsprozess stiessen wir auf die Experten von AdNovum. Das Unternehmen punktete mit seiner überragenden technischen Expertise und konnte uns, obwohl es noch wenig Erfahrung mit Scrum hatte, von seiner Bereitschaft überzeugen, in

Über triple dot Schweiz GmbH

triple dot berät Unternehmen in den Bereichen Business Fit und Lean IT in strategischen Fragen, komplexen Sachverhalten des IT-Managements und bei der Umsetzung grosser Transformationsprojekte. Das Unternehmen mit Sitz in St. Gallen wurde 2008 gegründet mit der Vision, seine Mandanten zu befähigen, den operativen Kurs optimal selbst zu steuern.

www.3dot.ch

diesem Kontext wirklich agil mit uns zusammenzuarbeiten. Das ist entscheidend, denn Scrum erfordert zwar technisches Können, funktioniert aber darüber hinaus nur mit Teamspielern.

Konsequenz 1: Der Auswahlprozess und der gewählte Partner sind insbesondere bei agilen Verfahren eminent wichtig.

Den Entwicklungspartner richtig einbinden

Häufig beobachten wir, dass Scrum einfach in einen klassischen Vorgehensansatz eingepasst wird. Dies führt in der Regel zum Verlust eines der wichtigsten agilen Vorteile, nämlich der Handlungsfähigkeit, und verdichtet so wieder die Integrationsrisi-

SCRUM ERFORDERT TECHNISCHES KÖNNEN UND FUNKTIONIERT NUR MIT TEAMSPIELERN.

ken am Ende des Vorhabens. Wir entschieden uns deshalb gegen Kompromisse und setzten auf 100% Scrum, um unser agiles Entwicklerteam sowie den Kunden in den komplexen Rahmen zu integrieren. Die in Scrum wirkenden Prinzipien und Verfahren sorgten schnell für erste Ergebnisse und förderten Vertrauen wie auch Verantwortungsbewusstsein bei allen Beteiligten. Damit entstand

in kurzer Zeit ein sehr gut funktionierendes Projektteam, das trotz wechselnder Besetzung und sich ändernder Schwerpunkte über das gesamte Projekt Bestand hatte und zunehmend schlagkräftiger wurde.

Konsequenz 2: In den Entwicklungsteams 100% Scrum – ohne Kompromisse.

WIE INTEGRIERT MAN SCRUM IN EINEN KONTEXT, DER PRAKTISCH NIE PER SE AGIL ORIENTIERT IST?

Scrum und die Organisation – Metarollen als Vermittler

Wir konnten zeigen, dass Scrum in der Entwicklung gut funktioniert und Rollen sowie Verantwortlichkeiten sehr geschickt gewählt sind. Aber wie integriert man Scrum in einen grösseren Organisationskontext, der praktisch nie per se agil orientiert ist? In unserer Erfahrung hat sich ein kleiner Kniff bewährt, der Raum für agiles Arbeiten schafft und der Umgebung wie dem Projekt hilft, sich gegenseitig zu verstehen und aufeinander zu verlassen.

So wie in Scrum klar getrennte Verantwortungen und strikte Regeln für die bekannten Effekte sorgen, bildeten wir mit zwei neuen Rollen eine Verständigungsbasis: Delivery Manager und Delivery Partner. Beide Rollen sind aus der Perspektive der Projektsponsoren gewählt und zeigen das Projekt in seinem grösseren Kontext, in dem das Projektteam die wesentliche – aber nur eine – Liefereinheit bildet. In diesem Sinn übernimmt der Delivery Manager gegenüber den Sponsoren die Gesamtverantwortung und entlastet mit einer Koordinationsfunktion den Product Owner.

Konsequenz 3: Der Product Owner ist fachlich verantwortlich für das Projekt und der Delivery Manager organisatorisch.

DIESE STRUKTUR ERMÖGLICHT FOKUSSIERTES ARBEITEN OHNE KOMPLIZIERTE REGELUNGEN.

Der Delivery Partner, in unserem Fall AdNovum, kann sich in dieser Konstellation voll und ganz auf die Entwicklung der Software konzentrieren und mit seiner Kreativität und seinen Möglichkeiten dem Product Owner sowie dem Delivery Manager den Handlungsspielraum bereitstellen, der schliesslich einen grossen Teil des Erfolgs ausmacht. Delivery Manager, Delivery Partner und Sponsoren bilden in diesem Verfahren einen Steuerungsausschuss, der die grundsätzliche strategische und operative Stossrichtung festlegt. Diese Struktur ermöglicht Personen mit unterschiedlichen Fähigkeiten und Kompetenzen fokussiertes Arbeiten ohne komplizierte Governance-Regelungen und Hierarchien.

Konsequenz 4: Geben Sie dem Entwicklungspartner eine klare Rolle und eine klare Verantwortung innerhalb definierter Regeln.

Orientierung schaffen – die Flughöhe anpassen

Zur weiteren Strukturierung wurde im Steuerungsausschuss das gesamte Vorhaben in Segmente unterteilt, in denen gemeinsam Schwerpunkte festgelegt wurden, ohne jedoch die agilen Prinzipien zu beschneiden. Im Unterschied zu klassischen Phasen bedeutet dies, dass in den einzelnen Segmenten kontinuierlich messbarer Wert, das heisst ein potentiell auslieferbares und vom Kunden getestetes Produkt, entsteht. Dank der Segmente können sich die Beteiligten im grösseren Projektkontext orientieren, ohne sich in den Details und der Priorisierung der einzelnen Anforderungen (User Stories) zu verlieren.

Konsequenz 5: Schaffen Sie eine Ebene, auf der Sie sich auch mit dem Rest des Unternehmens abstimmen können.

DIE ORGANISATORISCHE EINBETTUNG ERFORDERT INTENSIVE ARBEIT.

Natürlich ist dieses Vorgehen ebenfalls ein Prozess und die organisatorische Einbettung erfordert intensive Arbeit, um immer weiter vorangebracht zu werden. Dies konsequent agil zu tun hat jedoch den Vorteil, dass allfällige Probleme schnell erkannt und gelöst werden können. Probieren Sie es aus – aber richtig! ■

Impressum

Herausgeber:

AdNovum Informatik AG
Corporate Communication
Röntgenstrasse 22
CH-8005 Zürich
Telefon 044 272 61 11
E-Mail info@adnovum.ch
www.adnovum.ch

Verantwortung und Redaktion:


Manuel Ott
Feedback: notitia@adnovum.ch

Gestaltung und Realisation:

Rüegg Werbung, Zürich

Fotografie:

Patrick Rohner, Zürich (Titel); Gerry Nitsch, Zürich (Porträts); Márton Magócsi, Budapest (Porträts Ungarn)

Gedruckt auf Balance Pure  FSC