

Budgetgerecht in agilen Projekten?

Stephan La Rocca
PITSS GmbH
Bielefeld

Schlüsselworte

Projektmanagement, Projektcontrolling, Scrum

Einleitung

In der Software-Entwicklung haben agile Projektmethoden schon lange Einzug gehalten und viele Teams arbeiten in der Produktentwicklung nach Scrum. Recht schnell und begeistert ist das Team von Product Backlog, Sprint-Meetings und daily Scrum. Doch wenn es an die Budget-Planung geht, vor allem um die Abschätzung der Aufwände im Vorfeld und um die Einhaltung von Zeit- und Kostenrahmen verliert zunächst der Charme der Methode ihren Reiz. Agile Projektentwicklung und dennoch ein Festpreis für das Produkt scheinen in der ersten Betrachtungsweise im Widerspruch zu stehen.

Agiles Schätzen

Für eine Budget-Planung steht am Anfang die Kunst, die Kosten des vorliegenden Produktes zu bemessen und damit in eine Angebotsphase für die Zusammenarbeit zwischen Kunde und Lieferant zu treten.

Leider basieren solche Abschätzung meist auf der Erstellung eines Pflichten- oder Lastenheftes, welches an sich schon einen enormen Aufwand beziffert, am Ende aber dennoch nicht fein genug definiert wurde, um einem möglichen Streit der Parteien stand zu halten. Darüber hinaus ist bei umfangreichen Produkten damit jegliche Chance genommen, aus den Erfahrungen der ersten Projektphasen zu lernen und auf eventuelle geänderte Wünsche des Kunden einzugehen.

Sicherlich lässt sich der genaue Preis eines *neuen* Produktes, und an dieser Stelle ist *neu* so zu interpretieren, dass es im Vorfeld kaum eine Chance gibt, die Aufwände der Produktentwicklung genau zu bestimmen, erst nach den ersten Sprints bemessen.

Jedoch bestimmt der Kunde am Ende den Preis dadurch dass er dem Produkt einen gewissen Mehrwert zuordnet den er in seinem Geschäftsumfeld erzielen möchte. Nur wenn mit diesen Kosten auch die Scrum-Entwicklung getragen werden kann, kommen beide Seiten in einem erfolgreichen Projekt zusammen.

Benötigt ein agiles Projekt ein agiles Budget?

Die Frage entsteht aus der Betrachtung des sogenannten *Minimum Viable Product*, also dem kleinsten lauffähigen Produkt, mit dem der Kunde sich bereits einen Mehrwert für seinen Geschäftsprozess Unterstützung vorstellen kann. Sicherlich haben wir das gesamte Produkt vor Augen, aber in einem ersten Schritt betrachten wir zunächst die kleinstmögliche Menge die Der Kunde zum Einsatz bringen kann. Worin liegt der Vorteil dieser Betrachtungsweise?

- [Fokus auf ein kleines erstes Ziel
- [Zielgerichtete Kommunikation
- [Es werden nur Entscheidungen getroffen die auch tatsächlich benötigt werden
- [Erste Ergebnisse sind schnell sichtbar
- [Kleine Investitionen

| Schnelles Feedback vom Anwender

Für den Kunden bedeutet das, dass er ein Budget jeweils nur in kleinen Paketen beantragen und jeweils freigeben muss.

Große Projekte erfahren ihren Finanzdruck nicht am Anfang. Ist das Budget in einer ausreichenden Größe vorhanden, kann das Team schon mal eine lange Zeit in die falsche Richtung laufen, bevor ein Projekt Controlling Alarm schlägt.

Wird das Budget hingegen jeweils von Sprint zu Sprint freigegeben (mit einem gewissen Vorlauf um nicht in Finanzierungs-Genehmigung-Engpässe zu geraten) ist auch die Budget Kontrolle für beide Seiten transparenter und einfacher.

Wie bestimme ich das Budget

Bei der Bestimmung des Budgets fallen viele Projekte immer wieder darauf zurück, Aufwände und Zeiten zu schätzen und diese Stunden/Tagessätze mit einem „marktüblichen“ Satz zu versehen und dann damit in die Verhandlungen zu gehen. Auf Rabatten, Reisekosten, Junior/Senior-Tagessätze wird dann solange rumgeritten, bis beide Parteien halbwegs zufrieden sind, oder doch einer sich über den Tisch gezogen fühlt. Dieses Vorgehen weist deutliche Schwächen auf:

Die Consulting Falle:

Wir ein Consultant in seiner Dienstleistung besser, bekommt sein Kunde qualitativ bessere Leistungen in kürzerer Zeit und ... muss weniger zahlen. Das ist Nonsense.

Die Auftragsfalle

Der Kunde will ein Festpreis (selbst ein Zeit&Aufwands-Projekt mit Budget ist am Ende ein Festpreis) – aber in einem agilen Vorgehensmodell will er immer noch entscheiden, wie das Produkt am Ende aussehen soll. Auch das ist Nonsense.

Wie kommen in diesem Dilemma Kunde und Dienstleister zusammen?

Die einzige Chance ist es, nicht mehr die Stunden bezahlen zu lassen, sondern die umgesetzten User-Stories. Dabei wird werden die User-Stories nicht gleichbewertet, sondern an Hand ihres Business Value für den Kunden gewichtet. Für eine User-Story mit höherem Business Value bezahlt der Kunde mehr, als für eine User Story mit geringerem Business Value.

Wie kann der Business Value ermittelt werden?

Das kann nur im strategischen Umfeld des Kunden entschieden werden. Einfach ist es, wenn durch das Produkt unmittelbar Arbeit eingespart werden kann, aber nicht immer ist es so einfach monetär zu bestimmen. Marktanteile, schnellere Zyklen, zufriedener Kunden, etc. sind alles Faktoren, denen der Kunde eine Zahl zuweisen muss.

Betrachte die Funktionen des Produktes und nicht den Aufwand

Um die Komplexität einer User-Story zu bestimmen, um die Komplexität der Applikation und damit des zugehörigen Projektes zu erkennen, ist es nicht primär entscheidend, den Aufwand zur Implementierung zu betrachten, sondern die Anzahl der Funktionalitäten, die in der Applikation stecken. Was kann ihre Applikation, oder was soll sie können? Nach dieser Betrachtungsweise können Sie die User-Stories nach klein, mittel, groß, oder anderen Skalen einordnen. Haben Sie dann ein Gefühl für die Größe des Projektes, können Sie auch leicht ableiten:

- ein Gefühl für die Komplexität der Technologie
- alle beteiligten Personen

- Abhängigkeiten auch zu Lieferanten
- Gefühl für die zeitliche Dimension
- Reicht das Skill des Entwicklungsteam
- Ist eine Story noch zu groß
- Ein Gefühl für die Architektur

Für die Bestimmung solcher Größen existieren verschiedene Schätzverfahren, die mit der Erfahrung des Product Owner aber auch mit der Erfahrung des Teams oder der Projekte steigt.

Da die Verfahren innerhalb des laufenden Projektes immer und immer wieder zum Einsatz kommen und gleichfalls über die Einhaltung des Budgets entscheiden, werden diese im Vortrag beleuchtet. Durchgesetzt haben sich Magic Estimation und einige Spielvarianten dazu sowie das bekannte Planning Poker. Darüber hinaus soll das Team Estimation Game und eine Reihe weiterer Mini-Schätzspiele vorgestellt werden.

Die Rolle des Product Owner

In all dieser Betrachtungsweise fällt dem Product Owner ein großes Maß an Verantwortung zu. Er ist das Verbindungsglied zwischen der Softwareentwicklung und dem Kunden und verkörpert alle Wünsche und Anforderungen an das Produkt. Er hat es vor Augen, träumt davon, betrachtet es wie sein eigenes Kind, welches er aufwachsen sehen und helfen will.

In vielen Fällen ist allerdings schon die Besetzung des Produkt Owner nicht immer trivial. Bei einem ISV ist es vergleichsweise einfach, der klassische Produkt-Manager ist gewohnt, hier die Rolle zu übernehmen. Etwas schwieriger wird es schon bei der reinen Inhouse-Applikation, da sich hier die Fachabteilung und die IT-Abteilung gerne „aus der Verantwortung“ ziehen. Zwingend zu klären ist die Rollenverteilung dann, wenn externe Lieferanten als Coach oder als verlängerte Werkbank zum Einsatz kommen. Durchaus akzeptable ist eine Doppelbelegung dieser Position durch Vertreter zweier Parteien (Kunde/Auftragnehmer), wenn diese in enger Abstimmung gemeinsam ein Business Ziel verfolgen. Sicherergestellt werden muss, dass folgende Tätigkeiten abgedeckt sind:

- [die Wünsche und Anforderungen des Kunden zu verstehen und zu berücksichtigen
- [wirtschaftlich rentable zu planen
- [das Entwicklungsteam begeistert und (anwendungs-)fachlich zu führen
- [gemeinsam mit dem Team eine Produkt-Vision zu erstellen
- [das Product Backlog zu priorisieren und zu pflegen
- [die Stories der nächsten drei Sprints zu schätzen
- [stets zur Verfügung stehen, um fachliche Fragen zu beantworten
- [mit dem Kunden Meilensteine zu definieren

Somit fallen dem Product Owner Intuition, „guten“ Perfektionismus, Gestaltungswillen und Freiräume dazu, Professionalität und Qualität zu.

Agile Migration

In Migrationsprojekten ist die Ausgangslage dadurch gekennzeichnet, dass eine Vielzahl der User-Stories schon in der Legacy-Applikation versteckt sind. Die Migration soll nun aber in der Regel nicht alle User-Stories ungeschoren von einer alten Architektur in eine neue transferieren, sondern dem Kunden die Möglichkeit geben, sich von alten Zöpfen zu trennen und neue Prozesse zu etablieren. Es sollen somit gut designte User-Stories aus der Legacy-Applikation übernommen, schlechte verworfen und neue erstellt werden.

Die Agilität in dem Projekt ist somit eine zwingende Notwendigkeit, um die Weiterentwicklung der

Applikation während des Migrationsprojektes zu gewährleisten und am Anfang eines jeden Sprints darf es keinen Unterschied geben, ob eine User-Story aus der Legacy-Applikation stammt, oder aus der Feder einer neuen Prozessaufnahme.

Gerade für die Migration von Forms-Applikation soll der Vortrag einen Weg aufzeigen, User-Stories aus Legacy-Applikationen ermitteln und bewerten zu können.

Kontaktadresse:

Stephan La Rocca
PITSS GmbH
Otto-Brenner-Str. 209
D-33604 Bielefeld

Telefon: +49 (0) 521-546 795-07
Fax: +49 (0) 521-546 795-01
E-Mail: slarocca@pitss.de
Internet: www.pitss.de