

Priorisieren in Softwareprojekten

Sisyphusarbeit vermeiden

In der griechischen Mythologie muss Sisyphus jeden Tag einen Stein den Steilhang eines Berges heraufwälzen, wo eben dieser Stein, knapp unterhalb des Gipfels, zuverlässig wieder ins Tal rollt.

In manchen IT-Projekten muss das Team jeden Tag an neuen Features arbeiten, die es zuverlässig nicht über den Berg – ins Reich der Kundenzufriedenheit – schaffen. Sie werden nicht einmal benutzt.

„Priorisierung“ heißt der Begriff, der diese Verschwendung verhindern soll. Anforderungen zu priorisieren ist folglich eine der Kernaufgaben des Product Owners in Scrum. Sie zieht sich durch das ganze Projekt und sie ist bestimmd am Projektstart.

Doch über die Vorgehensweise beim Priorisieren ist damit noch nichts gesagt. Die Erfahrung zeigt, dass Kundenwünsche manchmal erheblich von dem abweichen, was ein Unternehmen vermutet hat. Wenn keine detaillierten Marktforschungen vorliegen und noch kein Prototyp existiert, den die Zielgruppe bewerten kann, was können Product Owner dann tun, um richtig zu priorisieren? Sie können beispielsweise das Kano-Modell einsetzen, ein Modell zur Analyse von Kundenwünschen [1].

Das Kano-Modell

Zugrunde liegt die Annahme, dass sich Kundenwünsche danach klassifizieren lassen, wie Kunden auf ihre Erfüllung reagieren. Kano unterscheidet Basis-, Leistungs- und Begeisterungsmerkmale. Zusätzlich nennt er auch unerhebliche und Rückweisungs-Merkmale. Sie spielen für das Backlog insofern eine Rolle, als zumindest die Rückweisungs-Merkmale auf keinen Fall auftreten sollten.

Aus rein pragmatischen Gründen werden sich Product Owner primär auf die drei „positiven“ Typen von Anforderungen konzentrieren. Basisanforderungen sind unerlässlich, ihr Fehlen würde sofort bemängelt. Dennoch gilt für sie, was der schwäbische Grundsatz „ned gschompfa isch globt gnuag“ (auf Hochdeutsch: Nicht geschimpft ist gelobt genug) ausdrückt. Das heißt, dass für ihr Vorhandensein kein Lob zu erwarten ist. Deshalb sind für den Product Owner die beiden anderen Merkmale so wichtig: die Leistungs- und die Begeisterungsmerkmale.

Auch bei den Leistungs-Merkmalen gilt Abwesenheit als Mangel, ihr Vorhandensein kann über bloße Neutralität hinaus jedoch zu deutlicher Kundenzufriedenheit führen.

Begeisterungs-Merkmale können die Erwartungen übertreffen, daher unterscheiden sie das Produkt vom Wettbewerb und faszinieren die Kunden.

Wie wichtig ist wichtig?

Um zu erfahren, ob eine Anforderung für die Kunden eher selbstverständlich oder ein Sahnehäubchen ist, sollte die Nutzerbefragung sowohl positiv (funktional) als auch negativ (dysfunktional) formulierte Fragen aufweisen. Im Idealfall besteht die Möglichkeit, die künftige Zielgruppe direkt zu befragen. Erfahrungsgemäß reicht es, 20 bis 30 ausgewählte Personen um ihre Meinung zu bitten. Nehmen wir ein einfaches Beispiel für eine funktionale Frage: Wie finden Sie es, wenn das Programm über eine Autokorrektur bei Schreibfehlern verfügt? Die dysfunktionale Frage ist folglich: Wie finden Sie es, wenn das Programm nicht über eine Autokorrektur verfügt? Antwortmöglichkeiten sind jeweils folgende: Das würde mich sehr freuen / Das setze ich voraus / Das ist mir egal / Das nehme ich gerade noch hin / Das würde mich sehr stören. Dysfunktionale Fragen werden bei Analysen der Kundenzufriedenheit häufig außer Acht gelassen. Gerade sie liefern jedoch sehr wichtige Ergebnisse.

Spätestens nach so einer Umfrage scheint die Priorisierung einfach zu sein. „Das setze ich voraus“ markiert Basismerkmale, „das würde mich sehr freuen“ steht für Leistungs- oder sogar Begeisterungsmerkmale. Also übernimmt der Product Owner die Basismerkmale, ergänzt einige der beiden höheren Kategorien und fertig ist das Backlog, oder?

WEB-TIPP:

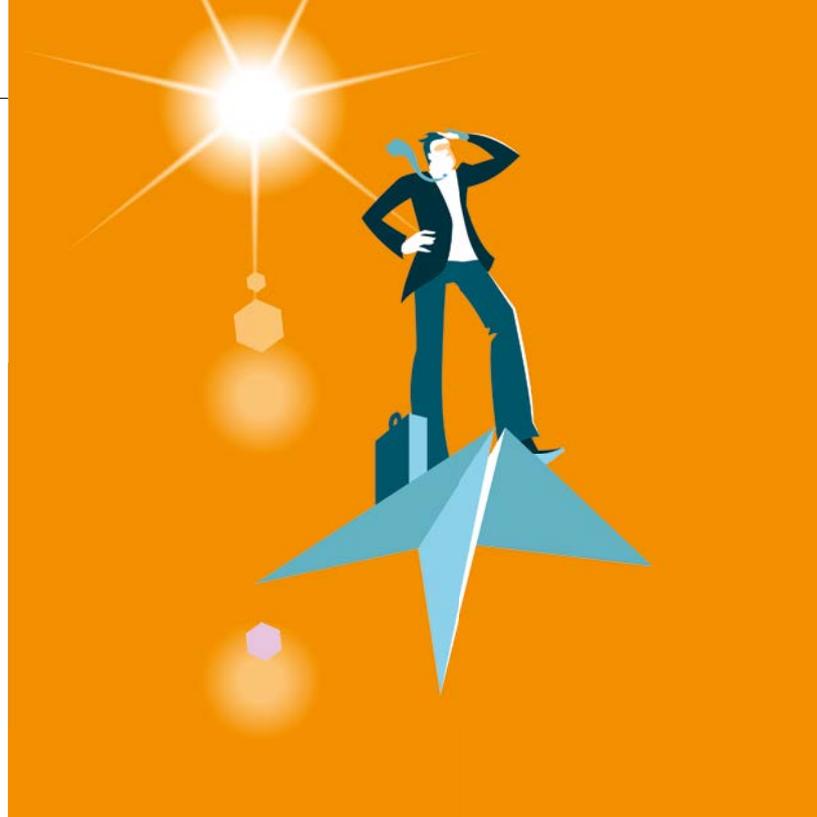
www.andrena.de



Spielerische Bewertung

Leider nicht. Die Problematik liegt im Überangebot an scheinbar unerlässlichen Anforderungen. Denn die Erfahrung zeigt, dass Nutzer zunächst dazu tendieren, viele oder sogar alle Funktionen als sehr wichtig zu bewerten. Daher bietet sich beispielsweise folgende „spielerische“ Bewertungsmöglichkeit an: Die Funktionen oder Anforderungen werden auf Karten geschrieben und auf dem Tisch ausgelegt. Wichtig ist dabei, dass sowohl der Product Owner als auch die teilnehmenden Nutzer ein gemeinsames Verständnis davon haben, was mit der jeweiligen Anforderung gemeint ist. Das klingt selbstverständlich, ist es jedoch erfahrungsgemäß nicht. Nehmen wir als Beispiel den Wunsch, sich direkt auf einer Webseite bei einem Event anmelden zu können. Demnach sollen die Entwickler ein Anmeldeformular in die Seite integrieren. Nicht automatisch geklärt ist jedoch, ob etwa der Erhalt einer Bestätigungsmail zu den Anforderungen gehört.

Den Nutzern sollten auch leere Karten zur Verfügung stehen, damit sie neue Ideen ergänzen können. Liegen alle Karten vor, erhält jeder Teilnehmer 10 Steine (z.B. Pokerchips) und darf sie auf die Karten verteilen: Je mehr Steine eine Funktion erhält, umso wichtiger ist sie für den jeweiligen Nutzer. Anders gesagt bewerten die Anwender die Funktionen jetzt also relativ zueinander. Erfahrungsgemäß erhalten dann auch Funktionen, die vorab als „sehr wichtig“ eingestuft wurden, gar keine oder wenige Steine. Im Gespräch wird deutlich, dass eine Funktion im direkten Vergleich eben doch weniger wichtig ist als andere – somit wertvollere - Funktionen. Zielführend kann auch sein, zwei oder drei Nutzer zusammen zu befragen. Jeder erhält eigene Chips (am besten in unterschiedlichen Farben) und verteilt diese für alle sichtbar auf die Karten. Die anschließende Diskussion der Nutzer über die Verteilung der Steine ist für den Product Owner häufig sehr interessant, bringt sie doch bislang unbekannt Informationen oder neue Anforderungen. Dementsprechend sollte der Product Owner und Vertreter des Entwicklerteams dabei immer anwesend sein.



Grundsätzlich muss der Product Owner bei der Priorisierung natürlich weitere Kriterien berücksichtigen. Dazu zählen unter anderem Kosten, Nutzen, Risiken, Abhängigkeiten oder die technische Notwendigkeit (z. B. Sicherheitsaspekte, gesetzliche Vorgaben).

Ein Widerspruch gegen die Nutzerbefragung im Sinne des Kano-Modells ergibt sich daraus nicht. Denn ein Product Owner kann noch so viel Sorgfalt auf die anderen Faktoren verwenden – keiner

von ihnen kann es ausgleichen, wenn aus Nutzersicht wichtige Funktionen fehlen.

Ein Aufwand, der sich lohnt

Die Priorisierung erweist sich in der Praxis schnell als Herausforderung, da plötzlich alles unentbehrlich zu sein scheint - oder zumindest oft mehr, als im vorgegebenen Rahmen möglich ist. Das Kano-Modell liefert meiner Erfahrung nach einen praxistauglichen Ansatz, herauszufinden, was für das Produkt aus Nutzersicht wirklich wichtig ist. Eine Einschränkung sei dennoch gemacht: Zu Priorisieren ist immer abhängig vom Produkt und der Situation. Es gibt kein Universalrezept, das immer passt. Meine Empfehlung lautet daher, verschiedene Möglichkeiten auszuprobieren und nach dem Prinzip „Inspect & Adapt“ zu verbessern.

Den eingangs erwähnten Anteil der nicht genutzten Funktionen zu reduzieren ist schlicht eine ökonomische Notwendigkeit. Man könnte sagen, richtig zu priorisieren bedeutet, nur an den richtigen Stellen zu sparen. Das schafft Wert für das Unternehmen und macht die Kunden zufriedener, die genau das erhalten, was ihnen wichtig ist. Kurz, es lohnt den Aufwand.

Und wer hat schon Sisyphus als Vorbild?

MARION GAKSTATTER

Quelle:

[1] <https://de.wikipedia.org/w/index.php?title=Kano-Modell&oldid=148836492>



„Das Kano-Modell liefert meiner Erfahrung nach einen praxistauglichen Ansatz, herauszufinden, was für das Produkt aus Nutzersicht wirklich wichtig ist.“

Marion Gakstatter, Scrum Master und Agile Coach, andrena objects