

Nutzerfeedback – Schwarmintelligenz richtig nutzen

Hilft das Feedback von hunderten Nutzern die Software zu verbessern? Bauen wir am Ende eine Software für die, die am lautesten gerufen haben? Weichen wir vom Auftrag/ Zweck ab, mit dem unsere Software einst angetreten ist? Was können wir tun, damit diese wertvollen Reaktionen auf unsere Software bestmöglich genutzt werden?

Verbesserungen an der Software leben auch vom Nutzerfeedback. Die Anwender arbeiten täglich mit der Software. Sie wissen bei welchen Aufgaben die Software sie unterstützen muss. Allerdings verändern sich die Anwendungsfälle mit der Zeit. Die zu unterstützenden Aufgaben oder die Arbeitsweisen/ Arbeitsroutinen haben sich seit der Einführung der Software wahrscheinlich geändert. Des Weiteren kann sich das Marktumfeld geändert haben und die Funktionen bringen jetzt weniger Nutzen als in der Vergangenheit. Auch kann neuer Bedarf entstanden sein, den wir als Anbieter noch nicht wahrgenommen haben. Diesen Punkt schätze ich besonders für Inhouse-Entwicklungen als Gefahr. Hier wird die Software mehr als Kostenfaktor, denn als wertvoller Beitrag zum Kundennutzen betrachtet und Konkurrenz führt nicht zu einer Auslese des schlechteren Produktes. Im Ergebnis heißt das für uns, dass:

- a) Funktionen obsolet sind
- b) Funktionen nicht mehr den aktuellen/ zukünftigen Anforderungen genügen
- c) zukünftig benötigte Funktionen noch nicht existieren

Auswirkungen von Änderungen ohne Berücksichtigung von Visionen, Zielen und Value

Es gibt zahlreiche Möglichkeiten das Nutzerfeedback aufzunehmen. Eine Hotline, Feedbackformulare, Workshops mit Nutzern und einige andere mehr.

Die Feedbacktickets sind hierbei meist isoliert aufgetretene negative Nutzungserlebnisse. Diese haben den Nutzer zu einem bestimmten Zeitpunkt oder auch schon über einen längeren Zeitraum genervt. Irgendwann macht

er sich dann die Mühe das Problem zu melden. Aus Sicht des Produktmanagements/ Product Owners sollten Änderungen unter anderem folgende Punkte erfüllen:

1. Generiert Mehrwert für eine größere Nutzergruppe
2. Ist mit der Unternehmensstrategie abgestimmt
3. Ist in der richtigen Sortierung (nach Value)
4. Ist eine ganzheitliche Lösung (Abhängigkeiten zu anderen fachlichen und technischen Themen betrachtet)

Die vier oben aufgeführten Punkte werden in den Tickets meist nicht berücksichtigt und es ist auch nicht die Aufgabe des Nutzers dies zu tun. Langfristiges Schaffen von Mehrwert und eine strategische Weiterentwicklung unserer Software wird aber ohne Berücksichtigung der vier Punkte zu einem Lotteriespiel. Setzen wir die Anfragen aus Tickets ohne Abstimmung mit den Zielen um, haben wir unseren Auftrag augenscheinlich erfüllt. Unseren Nutzern als auch unserem Unternehmen haben wir damit ggf. einen Bärenienst erwiesen.

Aus meiner Erfahrungen haben wir dann meist eine Software, die aus Nutzer- bzw. Techniksicht folgende Merkmale aufweist:

1. Aus Nutzersicht
 - I. gleiche Funktionen an mehreren Stellen
 - II. ähnliche Funktionen, die sich lediglich durch Nuancen unterscheiden
 - III. Bugs und nicht zu Ende gedachte/ entwickelte Funktionen
2. Aus Sicht der Technik
 - I. gleiche Funktionen an mehreren Stellen mit unterschiedlicher Logik
 - II. einfache Logik ist verloren gegangen und weicht einem Strauß von unwartbaren Varianten
 - III. aufwendiges Testen, der zahlreichen Varianten (wenn überhaupt möglich)
 - IV. fehlende fachliche und technische Verantwortung für Domänen und Prozesse, z.B. fehlen von Komponenten und Modulen, definierten Services

Wie kann ein mögliches Vorgehen aussehen?

Ein „first come, first served“ in der Umsetzung von Einzeltickets scheint aus meiner Sicht nicht hilfreich, da Änderungen dann meist nur dieses spezifische Nutzerproblem beheben. Die ganzheitliche fachliche und auch technische Betrachtung kommt mit diesem Ansatz meist zu kurz.

Stattdessen sollte das Nutzerfeedback in den kontinuierlichen und ganzheitlichen Verbesserungsprozess bzw. die reguläre Produktentwicklung integriert werden.

Wie bei jeder Produktentwicklung ist eine Vision für die Software der Startpunkt. Für die Vision könnten Antworten auf folgende beispielhafte Fragen gefunden werden.

- I. Was möchte ich mit der Software erreichen/ nicht erreichen?
- II. Welches Problem soll für den Nutzer gelöst werden/ nicht gelöst werden?
- III. Welche Aufgaben sollen den Nutzern leichter fallen, wenn sie unsere Software zur Hilfe nehmen?
- IV. Wie soll die Welt aussehen, wenn unser Produkt genutzt wird?

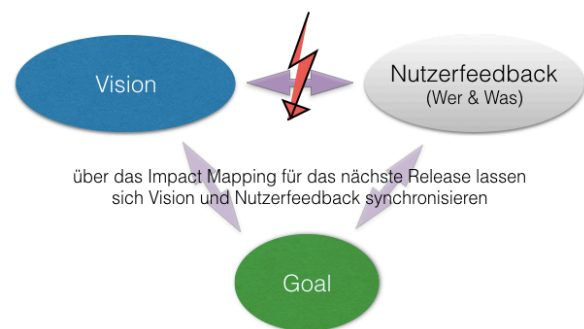
Nehmen wir an, wir sind eine Vertriebsorganisation FI-VER für Finanzprodukte mit einer CRM Anwendung (Fi-Sales). Die Vision könnte wie folgt lauten:

Fi-Sales soll unsere Vertriebsaktivitäten durch passgenaue Datenanalyse und Anzeige von Vertriebspotentialen unterstützen. Weiterhin soll Fi-Sales bei Routineaufgaben eine Option für die effiziente Bearbeitung der Aufgaben zur Verfügung stellen.

Daraus leitet der Produktmanager ab, dass neben der Datenanalyse und der aufbereiteten Anzeigen für die Kollegen vom Vertrieb auch die notwendigen Daten wichtig sind. Hierbei sollte sich auf die Daten konzentriert werden, die für eine Analyse und die Anzeige von Vertriebspotentialen nötig sind. Zur effizienten Bearbeitung von Routineaufgaben wird ein einfacher standardisierter Prozess angeboten. Daraus

folgt auch, dass alle anderen Funktionen so einfach wie möglich gehalten werden bzw. nicht entwickelt werden sollten.

Um eine zielgerichtete Weiterentwicklung der Software zu unterstützen werden die Feedbacktickets nicht einzeln betrachtet und bearbeitet, sondern als ein Teil des kontinuierlichen Verbesserungsprozesses bzw. als Teil der strategischen Ausrichtung verstanden. Daher wird eine Bewertung des Feedbacks und der möglichen Umsetzung vor dem Hintergrund der Vision und der gesteckten Ziele durchgeführt.



Hierfür verwendet F-VER das Impact Mapping¹. Es bringt das „Goal“ (Warum?), den Actor (Wer?), den Impact (Was?) mit dem „Deliverable“ (Wie?) in Einklang.

¹ <http://impactmapping.org>

Neuvertragsabschlüsse um 10% führen sollen. Hierfür haben die Produktmanagementteams und der Vertrieb verschiedene Dokumente, z.B. die Auswertung der Feedbacktickets und Unfallstatistiken vorbereitet. Diese sind dann Teil der Diskussion über neue die Maßnahmen in der Organisation und die neuen Funktionen in *FI-Sales*.

Beispielhaft werden zwei Tickets betrachtet.

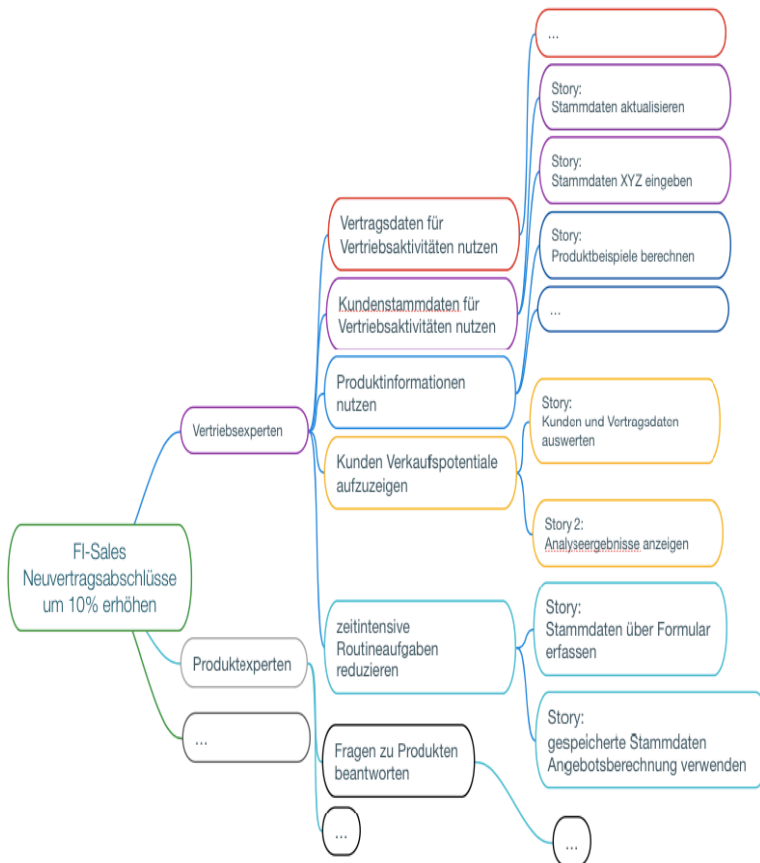
1. Versorgungslücke bei Sportunfällen aufzeigen
2. Geburtstagsgrüße automatisch versenden

Punkt 1 unterstützt hierbei direkt das Ziel der Erhöhung der Neuabschlüsse aus unserer Impact Map (Verkaufspotentiale aufzeigen). Punkt 2 ist zwar grundsätzlich ein Punkt der zur Verringerung des Aufwandes von Routineaufgaben fallen könnte, er zahlt aber aus Sicht des Produktmanagements nicht auf das aktuelle Releaseziel „Steigerung der Neuabschlüsse“ ein, sodass er für das aktuelle Release nicht berücksichtigt wird.

Alle Gedanken zur Produktweiterentwicklung sollten nach Prüfung auf ihre Relevanz für die Weiterentwicklung in geeigneter Form in das Backlog fließen. Mit der Einordnung der Feedbacktickets in unsere Impact Map schaffen wir eine effektive Weiterentwicklung unserer Anwendung.

So vermeiden wir, dass wir unser Backlog zu einer Stoffsammlung, mit Dingen machen, die wir nicht umsetzen sollten. Eine zu große Anzahl von Backlogitems erhöht dabei den Pflegeaufwand für das Backlog, was sich dann in mehr Meetings, mehr Abstimmung und verringerter Team-Performance niederschlägt².

Durch die Prüfung stellen wir sicher, dass wir ein Backlog haben, in dem sich nur zur Strategie/ den Zielen passende Stories



Durch die Nutzung des Impact Mapping kann das Produktmanagement von FI-VER leichter einschätzen, ob und für welche Ziele der Änderungswunsch einzahlt. Nehmen wir wieder Vision für FI-Sales als Richtschnur, so ergeben sich folgende Schwerpunkte:

Abbildung 1: Impact Map für FI-Sales

- I. die Datenerfassung von den Daten, die für die Analyse der Vertriebspotentiale verwendet werden
- II. die Funktionen, die die Vertriebskollegen von zeitraubenden Routineaufgaben befreien
- III. die Funktionen zur Berechnung und Anzeige der Analyseergebnisse

Funktionen, die die oben aufgeführten Schwerpunkte (I-III) nicht unterstützen, sollten daher entweder nicht berücksichtigt oder so einfach wie möglich (Basisfunktion) umgesetzt werden.

Ziel für das nächste Release sind Maßnahmen, die zu einer Erhöhung der

² <https://age-of-product.com/28-product-backlog-anti-patterns/>
- Danke Stefan für den Link :)

befinden. So können wir uns jetzt für das nächste MVP³/ Release diejenigen Stories raussuchen, die den größten Mehrwert (Value) und damit den größten Impact für unser Ziel haben.

Ziele für eine Release/ MVP sind meist: Kosten senken, Umsatz erhöhen oder Service verbessern. Die Ziele für das nächste Release hat das Produktmanagement mit der Erhöhung der Neuvertragsquote ausgegeben.

Aus der Impact Map nehmen wir also die Stories heraus, die unser Ziel (Steigerung der Neuabschlüsse um 10%) unterstützen. Diese Stories sollten als Startpunkt für eine Diskussion zwischen den Produktpartner (Customer, Technology und Business) dienen, die bestmögliche Lösung für das nächste MVP zu entwickeln.

Hierfür eignen sich Workshops, die eine ganzheitliche Betrachtung der Themen durch Vertreter der Produktpartner ermöglicht. Mit der oben beschriebenen Annahme, das Tickets meist nur einen begrenzten Ausschnitt des Nutzerverhaltens darstellen, sollten wir einen Prozess etablieren, in dem wir einen kontinuierlichen Austausch mit den Nutzern pflegen, idealerweise als Teilnehmer von Workshops.

Durch die Zusammenarbeit des Teams im Workshop werden die Teilnehmer sehr schnell zum Nachdenken angeregt und der Blick geht über den Tellerrand des Einzeltickets hinaus. Statt Einzeltickets können jetzt mit Hilfe der Gruppe ganzheitliche Lösungsoptionen entwickelt werden, um die vorgegebenen Ziele zu erreichen. Dabei dienen die Tickets jeweils als Input für den Prozess der Anforderungserstellung.

Natürlich müssen wir auch für eine Transparenz gegenüber den Nutzern in den Feedbackkanälen sorgen. Hier sollte klar sein, was mit den Tickets passiert und wie der Prozess der Umsetzung erfolgt. Wichtig ist hierbei auch die Begründung für abgelehnte Tickets. Dies ist z.B. mit der Impact Map leichter und sollte besser akzeptiert werden können, als mit Standardantworten. Wir

wollen ja nicht die Nutzer verprellen, die uns wertvolle Anregungen für Weiterentwicklung der Software geben.

Zusammenfassung

Nutzerfeedback ist eine wertvolle Quelle zur Verbesserung unserer Software. Es sollte allerdings nicht losgelöst vom Zweck der Software und den Zielen des Unternehmens betrachtet werden, sondern in den regulären Prozess der Produktentwicklung einfließen. Die Zusammenarbeit von Customer, Business und Technology Kollegen bei der Erstellung der Lösungsoptionen führt dazu, dass die atomaren Probleme hinterfragt werden. Die Organisation und die Nutzer bekommen ein gemeinsames Verständnis von den Zielen und Wünschen der anderen Rollen und es kann eine bessere und ganzheitliche Lösung erarbeitet werden.

Der Fokus auf einen Bereich/ Schwerpunkt führt zwangsläufig dazu, dass andere Bereiche keine Veränderungen/ Verbesserungen erfahren. Da unser oberstes Ziel die Schaffung von Mehrwert sein sollte und unsere Sortierung nach Value für den Kunden erfolgt, ist das Prinzip Gießkanne oder „First come, first served“ nicht zu empfehlen.

Nutzerfeedback sollte daher vielleicht nicht als Problem verstanden werden, welches isoliert behoben werden muss, sondern besser als Anregung zur ganzheitlichen Lösungskonzeption verstanden werden.

Happy Engineering
René

³ Minimum Viable Product
https://de.wikipedia.org/wiki/Minimum_Viable_Product