

Ursachenforschung im agilen Umfeld

OKRs nicht erfüllt – woran kann es liegen?



Peter Rubarth
Systemischer Agile Coach

Management Summary

- Objectives und Key Results (OKRs) ist eine Zielmanagement-Methode, die zweigeteilt ist: Mit Objectives werden qualitative Ziele des Managements, mit Key Results quantitativ erfassbare Maßnahmen zur Umsetzung der Objectives bezeichnet.
- Ein Technologie-Startup wächst schnell, arbeitet agil und nutzt zur Zielerreichung OKRs. Trotz erfahrener und kompetenter Mitarbeiter:innen stellt sich bei der OKR Review eines Teams heraus, dass die Ziele 0% erreicht wurden.
- Die beteiligten Teams, ein Entwicklungsteam und ein Serviceteam, arbeiten respektvoll und konstruktiv zusammen. Die Erfüllungsquote von 0% stellt einen Widerspruch zwischen der wahrgenommenen Zusammenarbeit und den Ergebnissen dar. Eine Ursachenforschung wird angeordnet.
- Es stellt sich heraus, dass häufige Wechsel im Team, eine initiale Fehleinschätzung der Komplexität des Projekts, eine inkonsequente Umsetzung der Sprints und nicht einsatzbereite Produktinkremente zum desaströsen Ergebnis geführt haben. Auch hier wird weiter geforscht.
- Letztendlich zeigt sich, dass der Wunsch nach einem Big Bang Release seitens des Serviceteams und häufige Product-Owner-Wechsel aufgrund der initialen Fehleinschätzung der Komplexität dazu geführt haben, dass die Erfüllungsquote bei 0% lag.
- Als Empfehlung für künftige Projekte gilt, einsatzbereite Inkremente innerhalb von maximal drei Monaten zu erstellen. Um sicherzustellen, dass das Projekt nicht aus dem Fokus gerät, werden die OKRs bereichsübergreifend priorisiert.

Ein Technologie-Startup hat seinen Markt gefunden. Es wächst und wächst. Das Unternehmen steht vorn in den Ranglisten und wird als potenzielles Unicorn gehandelt.

Die Belegschaft von ca. 500 Mitarbeiter:innen ist jung, international und exzellent ausgebildet. Agile Softwareentwicklung ist der Standard. In den ca. 20 Produktentwicklungsteams gibt es niemanden, dem man die Idee von iterativ-inkrementellem Vorgehen erklären muss. Die meisten Entwickler:innen kennen es nicht anders. Die Produkte sind Everything as a Service (XaaS) Lösungen in einem hoch regulierten Um-

feld. Sie ermöglichen es den B2B-Kund:innen, Dienstleistungen per Softwareschnittstelle in ihre eigenen digitalen Produkte einzubinden, ohne sich mit den regulativen Details auseinandersetzen zu müssen.

Ein Beispiel für diese Art von Produkt ist eine Softwareschnittstelle, um als Betreiber einer Website datenschutzrelevante Datenverarbeitung an eine/-n Dienstleister:in auszulagern. Der/die Dienstleister:in übernimmt die Datenverarbeitung und stellt die Einhaltung als Vorschriften sicher.

Die Bewältigung des Wachstums ist ein großes Thema. Insbesondere die Serviceabteilungen stoßen bei der Bewältigung des exponentiellen Wachstums an ihre Grenzen. Ihre Rolle ist – vergleichbar mit einem Backoffice –, die in Zusammenhang mit der Nutzung der XaaS-Produkte anfallenden Geschäftsvorfälle abzuwickeln. Es ist offensichtlich, dass das steigende Arbeitsaufkommen mit zusätzlichen Mitarbeiter:innen alleine nicht zu bewältigen ist. Der manuelle Aufwand muss erheblich reduziert werden, um die schnell steigenden Volumina mit ungefähr gleichbleibender Mitarbeiter:innen-Anzahl zu stemmen.

Aus diesem Grund wurde ein Projekt zur Automatisierung ausgewählter interner Prozesse initiiert. Die fraglichen Prozesse sind einerseits besonders zeitaufwändig. Andererseits scheinen sie vergleichsweise einfach automatisierbar zu sein. Die Grundidee hinter dem Projekt ist, dass bisher notwendige manuelle Schritte durch neue Funktionalität des digitalen Produkts ersetzt werden.

Bei dieser Art von Automatisierungsprojekten stellen die Serviceteams einen internen Auftraggeber dar und bringen ihre fachliche Expertise ein. Die Produktentwicklungsteams, zusammengesetzt aus Softwareentwickler:innen und Product Ownern, verantworten die konzeptionelle und technische Entwicklung bestimmter XaaS-Produkte.

Die Produktentwicklungsteams sind für jegliche technische Weiterentwicklung der XaaS-Lösungen zuständig. Insbesondere bestehende und neue B2B-Kund:innen haben eine lange Liste mit nachgefragten zusätzlichen Features. Es gibt dementsprechend immer weitaus mehr Nachfrage als Kapazität vorhanden ist. Daher ist ein Platz auf der Roadmap der Produktentwicklungsteams sehr wertvoll. Für die Serviceteams ist diese Situation schwierig, denn sie sind angewiesen auf die Produktentwicklungsteams, konkurrieren aber mit externen, umsatzbringenden Anfragen.

Die OKR Review - ein Desaster

Für die Ausrichtung der Teams an den Unternehmenszielen wurden vor mehreren Jahren "Objectives and Key Results" (OKRs) als Zielmanagement-Methode eingeführt.

Vor diesem Hintergrund findet die OKR Review einer Serviceabteilung statt. Ich begleite die Abteilung seit einigen Monaten als Agile Coach bei Themen der Teamentwicklung und Prozessoptimierung. Eines dieser Themen ist die Ausgestaltung des OKR-Prozesses in der Abteilung.

Aus Agile Coach Perspektive bin ich sehr zufrieden mit dem Verlauf des Termins. Die anwesenden Teamleiter stellen ihre Ergebnisse vor. Die Atmosphäre ist konzentriert, offen und wertschätzend.

Ich horche auf, als bei einem Team 0% Zielerreichung festgestellt wird. Inhalt dieses Key Result war eine Produktoptimierung mit dem Ziel, die Effizienz im Team um wenigstens 50% zu steigern. Die Fertigstellung der geplanten Funktionalität wurde mehrfach verschoben und ist inzwischen mehrere Monate in Verzug. Ein konkreter Liefertermin ist nicht klar.

Ohnehin ist das Vertrauen in die Einhaltung derartiger Zusagen nach den Erfahrungen der Vergangenheit gering. Da die Personalplanung des Serviceteams auf den ursprünglichen Terminzusagen und den erwarteten Einsparungen basiert, ist das Team an der Belastungsgrenze. Das Arbeitsaufkommen wächst kontinuierlich und die Personalstärke reicht nicht mehr aus.

Objectives and Key Results

"Objectives and Key Results" (kurz OKRs) ist ein Ansatz zur Formulierung von Zielen, welche in den 1980er Jahren von Andrew Grove bei IBM entwickelt wurde. Populär wurden OKRs durch den Einsatz bei Google.

Eine zentrale Idee von OKRs ist die Zweiteilung in Objectives und Key Results. Mit Objectives werden qualitative Ziele beschrieben. Die zu den Objectives gehörenden Key Results beschreiben Vorhaben, von deren Umsetzung man sich eine Erreichung der Objectives verspricht. Die Key Results müssen dabei so formuliert sein, dass eine objektive und quantitative Bewertung der Umsetzung möglich ist.

OKRs werden in regelmäßigen Zyklen gesetzt und die Ergebnisse reviewt. Ein Zyklus dauert üblicherweise drei Monate. Die Auswertung erfolgt in der Form, dass für jedes Key Result die Erreichung zwischen 0 und 100% festgestellt wird. Dabei entsprechen 70% Zielerreichung den realistischen Erwartungen. Dieses Vorgehen soll die Setzung von ambitionierten Zielen fördern, um so Ergebnisse jenseits der ursprünglichen Erwartungen zu ermöglichen.

Aus demselben Grund soll die Erreichung von OKRs nicht mit Bonuszahlungen oder anderen Anreizsystem verknüpft werden, um ein „Gaming“ zu vermeiden.

Anders als etablierte Zielesysteme kaskadieren OKRs nicht. Ziele werden nicht herunter gebrochen. Stattdessen formulieren selbstorganisierte Teams ihre Ziele selbst und justieren die Zielformulierungen im Dialog miteinander und mit der Geschäftsführung. Durch diesen Dialog werden Unklarheiten und gegenseitige Abhängigkeiten aufgedeckt und die Ausrichtung aller Teams an dem gemeinsamen Unternehmenserfolg kann verbessert werden.

OKRs haben, richtig eingesetzt, den Vorteil, dass sie dezentrale Teamstrukturen ermöglichen. Sie helfen bei der Koordination, ohne einzuengen. OKRs integrieren sich gut mit anderen agilen Vorgehensweisen und ergänzen diese um einen mittelfristigen Planungshorizont.

Der Auftrag: Ursachenforschung zum OKR-Desaster

Nach der OKR Review erhalte ich eine Nachricht des Abteilungsleiters. Er bittet mich darum, herauszufinden, warum das Projekt nach zwölf Monaten immer noch keine Ergebnisse geliefert hat. Ich soll die Resultate meiner Ermittlungen in vier Wochen auf einer bereichsübergreifenden Managementsitzung präsentieren.

Mitarbeiter befragen und relevante Ereignisse zeitlich einordnen

Ich mache mich also ans Werk und beginne mit den Ermittlungen. Das interne Wiki stellt sich als gute Quelle heraus. So finde ich beispielsweise Dokumentationen von vergangenen Teamretrospektiven aus denen sich Hinweise ergeben, wann welche Probleme thematisiert wurden. Nach dieser ersten "Spurensicherung" beginne ich damit, Mitarbeiter:innen zu befragen.

Die Perspektive der Teamleiterin des Serviceteams kenne ich. Sie hat mir in der Vergangenheit wiederholt davon erzählt, dass versprochene Lieferungen verschoben wurden und auch, dass sie wieder einmal eine neue Product Ownerin in die fachliche Thematik einarbeiten muss. Also spreche ich zunächst mit der Product Ownerin des Produktentwicklungsteams und einzelnen Teammitgliedern.

Nächste Stationen sind die involvierten Führungskräfte der nächsthöheren Ebene. Zu dieser Gruppe zählt zunächst der Vice President Product, welcher die fachliche und disziplinarische Führung der Product Owner verantwortet. Sein Gegenpart auf Seiten der Technik ist ein Engineering Manager mit dem ich auch spreche. Den Abschluss bildet der Leiter des Geschäftsbereichs, welcher die kommerzielle Seite verantwortet.

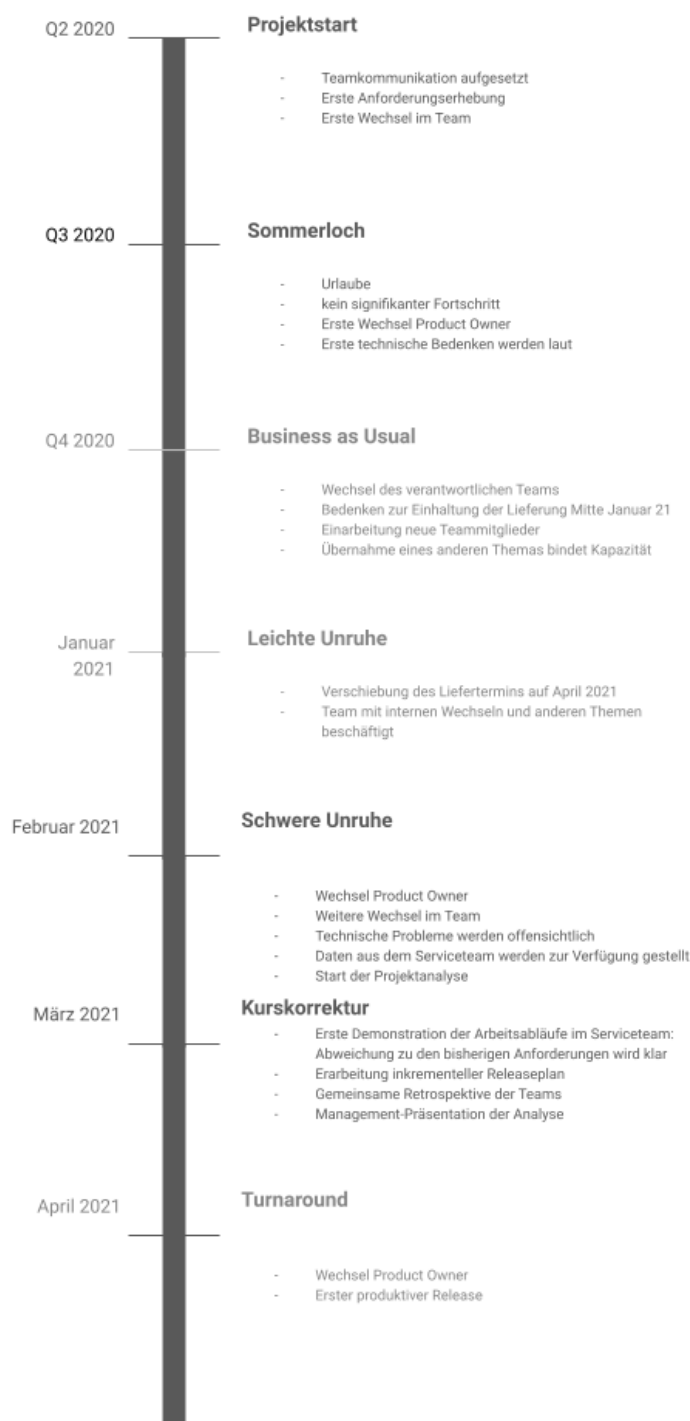


Bild 1: Zeitstrahl mit relevanten Ereignissen für das Projektziel

Bei der Auswahl der Gesprächspartner:innen geht es mir darum, eine gute Mischung von fachlichen Hintergründen und Hierarchieebenen zu erhalten.

Insbesondere das Gespräch mit der Product Ownerin ist ergiebig. Sie verschafft mir Zugriff auf die Dokumentation von Retrospektiven, die bereits auf Teamebene erfolgt sind. Besonders wertvoll ist eine Darstellung des zeitlichen Verlaufs relevanter Ereignisse, welche die Product Ownerin aus dem Teamchat gewinnt. Ich lerne auch, welche Maßnahmen die Product Ownerin ergriffen hat, seit sie zwei Monate zuvor die Verantwortung für das Projekt übertragen bekommen hat (siehe Bild 1).

Als Ergänzung führe ich eine teamübergreifende Retrospektive mit Vertreter:innen des Produktentwicklungsteams und des Serviceteams durch.

Ich lerne, dass die Product Ownerin bereits Maßnahmen ergriffen und umgesetzt hat, um das Projekt wieder auf Kurs zu bringen. Die Retrospektive zeigt mir, dass diese Maßnahmen von beiden Teams als sinnvoll angesehen und voll unterstützt werden.

Vorbildliche Projektteams und trotzdem schlechte Ergebnisse

Aus den Gesprächen und den Dokumenten ergibt sich ein zunehmend klares Bild des bisherigen Verlaufs und der Probleme.

Wenn Projekte in Verzug geraten, wird die Schuld oft zuerst beim Projektteam und den Projektverantwortlichen gesucht. Die Daten, Gespräche und ergriffenen Maßnahmen zeichnen ein anderes Bild:

Das Team strahlt ein bewundernswertes Maß an Verantwortung und Initiative aus. Das Team hat im Angesicht zahlreicher Hindernisse, Planänderungen und erheblichen Drucks nie aufgehört, konstruktiv, engagiert und lösungsorientiert zusammenzuarbeiten. Der Umgang ist offen und wertschätzend. Die fachliche Kompetenz wird von niemandem meiner Gesprächspartner:innen infrage gestellt.

Auch die Zusammenarbeit zwischen Produktentwicklungsteam und Fachteam ist offen und konstruktiv. Und das obwohl der erhebliche Verzug und die daraus resultierenden Probleme durchaus zu gegenseitigen Schuldzuweisungen verleiten könnten.

Die Qualität der Zusammenarbeit ist keineswegs selbstverständlich und ich habe mehr als einmal das genaue Gegenteil erlebt – unter weniger herausfordernden Umständen.

Dieser überraschend positive Eindruck ändert natürlich nichts daran, dass die mit dem Projekt verfolgten Ziele erheblich verfehlt wurden. Im Gegenteil, die Situation erscheint als krasser Widerspruch zwischen der Qualität der Zusammenarbeit und den Resultaten.

Wie ist es also dazu gekommen?

Das Ergebnis: 8 Faktoren führten zur OKR-Nullrunde

Die gewonnenen Daten zeigen, dass die Verkettung von mehreren Faktoren zur Schiefelage des Projekts geführt hat:

1. Unterschätzung der fachlichen Komplexität zu Beginn

2. Übergabe des Projekts an einen unerfahrenen Product Owner (ausgehend von der Fehleinschätzung)
3. Wechsel der Projektverantwortung zwischen Teams
4. Sehr häufige Wechsel der Teamzusammensetzung und insbesondere der Product Owner
5. Unterbrechung der Arbeit am Projekt aufgrund anderer, zeitkritischer Themen – ohne dass ein produktiver Zwischenstand fertig gestellt wurde
6. Spätes Infragestellen der ursprünglichen (und fehlerhaften) Grundannahmen
7. Ineffektive, zu oberflächliche Abstimmung des fachlichen Konzepts zwischen Produktentwicklung und Serviceteam
8. Langes Festhalten an der Forderung nach der Umsetzung aller Anforderungen seitens des Serviceteams

1. Unterschätzung der fachlichen Komplexität zu Beginn

Bei vielen technischen Themen steckt der Teufel im Detail. In diesem Fall ergaben sich unerwartete Herausforderungen aus den Begrenzungen einer relevanten Drittsoftware, Spezialfälle aufgrund rechtlicher Anforderungen und der notwendigen Verzahnung mit den verbleibenden manuellen Prozessen, welche auch nach dem Wegfall der automatisierten Prozesse bestehen bleiben.

Vermeidung bzw. Gegenmaßnahme

Umfassende Spezifikationsphasen scheinen naheliegend, sind aber nur scheinbar eine Lösung. Bei komplexen Rahmenbedingungen zeigen sich viele Probleme erst bei der Umsetzung. Und Spezifikationen liefern keinen Wert. Wenn die Zeit ohnehin knapp ist, besteht bei einer langen Analysephase das Risiko, dass das Projekt wegen wechselnder Prioritäten abgebrochen wird, bevor überhaupt etwas entwickelt wurde.

Ein besseres Vorgehen ist, in Etappen vorzugehen und dabei gleich zu Anfang einen horizontalen Schnitt durch die Anforderungen zu machen. Mit „horizontalem Schnitt“ ist gemeint, dass alle Bereiche einer Anwendung zunächst oberflächlich umgesetzt werden. Demgegenüber meint ein „vertikaler Schnitt“ die Umsetzung eines Teilbereichs in großer Tiefe – wobei alle anderen Aspekte zunächst vernachlässigt werden. Ein horizontaler Schnitt hilft, das Zusammenspiel der Komponenten zu verstehen und Integrationsprobleme aufzudecken. Im besten Fall steht nach dem ersten Sprint eine rudimentäre aber funktionsfähige Fassung der gesamten Anwendung, welche dann schrittweise um Funktionalität angereichert wird. Der große Vorteil dieses Vorgehens ist, dass immer eine lauffähige Fassung der Anwendung vorliegt, mit einem stetig steigenden Funktionsumfang.

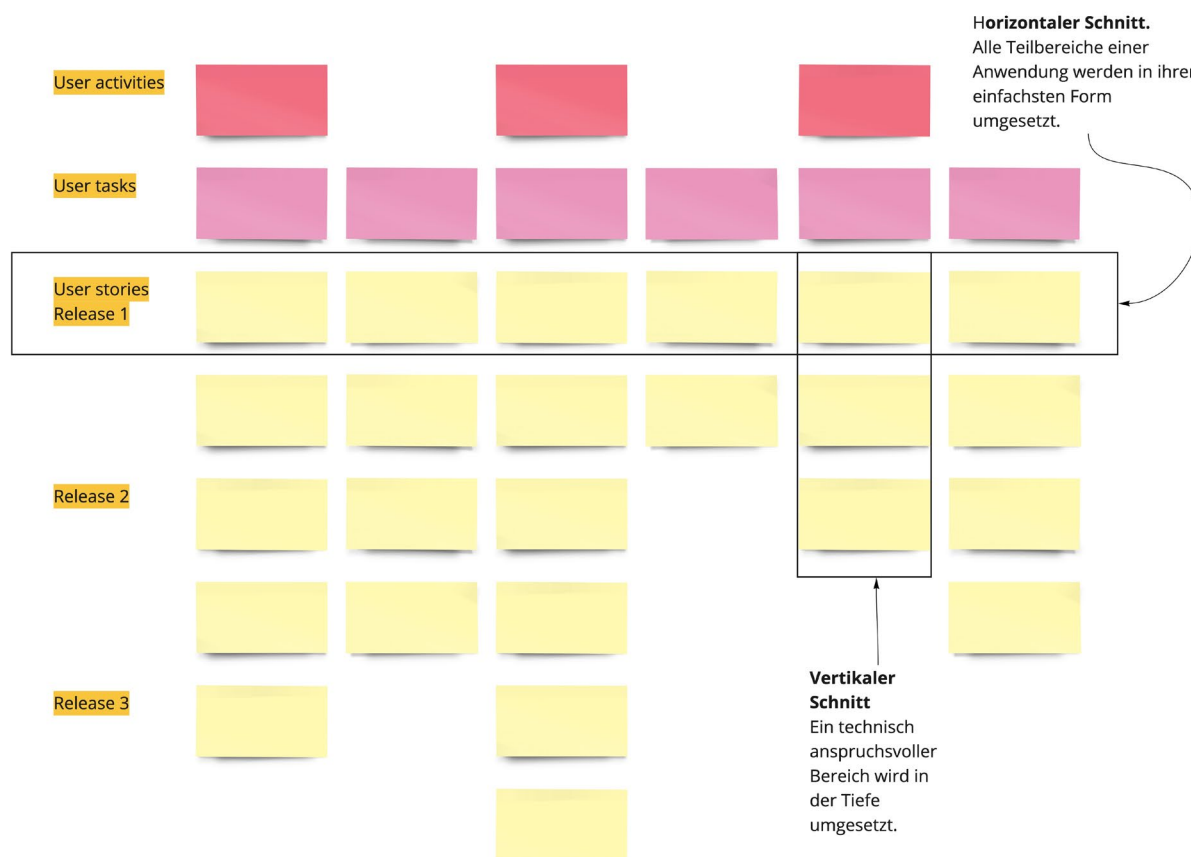


Bild 2: Unterschied zwischen horizontalem und vertikalem Schnitt

Bei Bedarf kann in einem zweiten Schritt ein vertikaler Schnitt erfolgen, um besonders komplexe und damit für den Gesamterfolg kritische Aspekte der Anwendung zu untersuchen. Es ist besser, solche Risiken möglichst früh zu adressieren, um nicht kurz vor Ende eines Projekts auf Probleme zu stoßen, die dann den Zeitplan sprengen.

Unabhängig davon hätte eine Visualisierung des Gesamtprozesses zusammen mit allen Projektbeteiligten einige der Probleme frühzeitig und ohne großen Zeitaufwand aufgedeckt (beispielsweise mit einem Virtual Whiteboard). Durch eine solche Visualisierung lassen sich Abläufe und vor allem das Zusammenspiel unterschiedlicher Prozesse viel besser erfassen. Auf diese Weise lässt sich z.B. erkennen, wenn ein Prozess auf Daten von einem anderen Prozess angewiesen ist, die dieser aber gar nicht liefern kann – oder in einem anderen Format.

Eine geeignete Methode für kollaborative Visualisierung wäre z.B. Event Storming. Event Storming ist eine Methode der Anforderungsplanung. Sie stammt aus dem Domain-driven Design, einer Philosophie zur Entwicklung komplexer Softwaresysteme.

2. Übergabe des Projekts an einen unerfahrenen Product Owner

Das Projekt wurde zu Beginn an einen Junior Product Owner übergeben, der außerdem gerade erst eingestellt worden war. Aufgrund der unterschätzten Komplexität war dieses Vorgehen zumindest plausibel.

Dem Product Owner ist es in der Folge nicht gelungen, die fachliche Komplexität zu durchdringen, dabei die Fehleinschätzung seiner erfahrenen Kollege:innen zu hinterfragen und das Projekt angemessen zu strukturieren.

Daneben war es diesem Product Owner nicht möglich, den fordernden Stakeholdern mit der notwendigen Autorität gegenüberzutreten und statt des geforderten „Big Bang“ ein iterativ-inkrementelles Vorgehen durchzusetzen.

Vermeidung bzw. Gegenmaßnahme

Es muss das Vorgehen hinterfragt werden, einem neu eingestellten, unerfahrenen Product Owner ein solches Projekt zu übergeben – ohne ihn dabei intensiv zu begleiten. Eine bessere Einarbeitungsstrategie wäre sicherlich gewesen, den neuen Kollegen für einige Zeit im Tandem mit einem erfahrenen Product Owner einzusetzen.

Ein konsequent iteratives Vorgehen hätte die Folgen der Fehleinschätzungen früher sichtbar gemacht.

3. Wechsel der Projektverantwortung zwischen Teams

In Folge der initialen Fehleinschätzungen wurde das Projekt an ein Team übergeben, welches gerade Kapazität hatte. Als dann im Verlauf festgestellt wurde, dass für das Projekt erhebliches Domänenwissen erforderlich ist, musste das Projekt an ein anderes Team übergeben werden. Diese Übergabe kostete Zeit und es ging Erfahrung mit dem Thema verloren, die nicht vollständig übertragen werden konnte und neu aufgebaut werden musste.

Vermeidung bzw. Gegenmaßnahme

Für diesen Aspekt ist es schwierig, eine generelle Empfehlung zu geben. Im konkreten Fall war die Entscheidung unter den vorliegenden Informationen plausibel. Ein besseres Verständnis der Anforderungen hätte zu einer anderen Entscheidung geführt. Aber das ist gewissermaßen ein Folgefehler der anfänglichen, zu wenig gründlichen Analyse.

4. Sehr häufige Wechsel innerhalb des Teams und insbesondere der Product Owner

Im Verlauf des Projekts waren insgesamt vier Product Owner im Entwicklungsteam mit dem Thema befasst. Die technische Leitung im Entwicklungsteam wechselte ebenfalls – insgesamt sechsmal. Insgesamt waren zwölf Softwareentwickler mit dem Thema befasst, wovon fünf im Verlauf des Projekts ausgewechselt wurden.

Diese Wechsel hatten unterschiedliche Gründe. In der Summe ist die Diskontinuität dramatisch – nicht nur wegen des Wissensverlusts, sondern auch wegen der Auswirkungen auf das Teamgefüge.

Vermeidung bzw. Gegenmaßnahmen

Die einzelnen Wechsel im Team waren nicht zu vermeiden. Eine längerfristige Einsatzplanung hätte die Auswirkungen auf das Team abmildern oder ganz vermeiden können.

Insgesamt hat das Team durch die Anwendung von standardisierten Entwicklungspraktiken und effektiver Dokumentation den Effekt der häufigen Wechsel deutlich abgemildert.

5. Unterbrechung der Arbeit am Projekt aufgrund anderer Themen

Wie bereits erwähnt, ist die Kapazität der Produktentwicklungsteams sehr begehrt und eine Vielzahl von Themen stehen in Konkurrenz um einen Platz auf der Roadmap.

Das Produktentwicklungsteam musste die Arbeit an dem Projekt für mehrere Monate unterbrechen, weil ein anderes Thema höher priorisiert wurde. Diese Unterbrechung führte nicht nur zu Verzug, sondern auch zu Verlust von Wissen. Zudem konnte kein Inkrement fertiggestellt werden.

Vermeidung bzw. Gegenmaßnahme

In einem dynamischen Umfeld ändern sich Prioritäten. Es ist eines der zentralen Argumente für agiles Vorgehen, auf Veränderungen reagieren zu können.

Ein effektives, inkrementelles Vorgehen von Beginn an hätte dazu geführt, dass zum Zeitpunkt, wo die Prioritäten verschoben wurden, bereits Zwischenergebnisse geliefert worden wären und die damit realisierten Effizienzgewinne den Druck auf das Team reduziert hätten.

Hier zeigt sich, dass es sehr leicht möglich ist, der Täuschung zu unterliegen „Wir arbeiten in Sprints, also sind wir agil“. Wenn es am Ende eines Sprints nicht möglich ist, zusammen mit Stakeholdern konkrete Zwischenergebnisse dahingehend zu betrachten, ob die Richtung stimmt, dann ist es kein effektives, inkrementelles Vorgehen.

Es ist im Eifer des Gefechts sehr leicht, in diese Falle zu tappen, ohne es zu merken. Gute Scrum Master oder Agile Coaches sind unter anderem dazu da, aus ihrer Außenperspektive derartige Fehler zu erkennen und zum Thema zu machen.

6. Spätes Infragestellen der fehlerhaften Grundannahmen

Die hier beschriebenen Erkenntnisse wurden erst durch die dritte Product Ownerin erzielt, die nach erheblichem Verzug und ca. vier Wochen vor dem Beginn meiner Untersuchung mit dem Projekt betraut wurde. Bis zu diesem Zeitpunkt wurde nicht erkannt, dass fundamentale Grundannahmen nicht zutreffend waren.

Vermeidung bzw. Gegenmaßnahmen

Ein effektives iterativ-inkrementelles Vorgehen und insbesondere eine Planung in Inkrementen und eine Inspektion dieser Inkremente zusammen mit den Stakeholdern aus dem Serviceteam hätte zwangsläufig aufgedeckt, dass wesentliche Aspekte nicht berücksichtigt wurden.

7. Ineffektive und zu oberflächliche Abstimmung des fachlichen Konzepts

Die Product Owner und die Leitung des Serviceteams standen in regelmäßigem Kontakt. Bei diesen Gesprächen gelang es nicht, ein hinreichend tiefes gemeinsames Verständnis der fachlichen Hintergründe aufzubauen. Mit anderen Worten: Die Product Owner haben nicht erkannt, dass sie mit ihrem Verständnis immer noch an der Oberfläche kratzen. Begünstigt wurde dies sicherlich durch die häufigen, personellen Wechsel.

Vermeidung bzw. Gegenmaßnahmen

Die Verwendung von Visualisierungstechniken, wie das bereits erwähnte Event Storming, ist eine gute Möglichkeit, schnell und effektiv eine gemeinsame Sprache zu finden und auf dieser Grundlage ein umfassendes, gemeinsames Verständnis zu schaffen.

8. Langes Festhalten an der Forderung nach Umsetzung aller Anforderungen durch das Serviceteam

Wie bereits erwähnt, sind die Serviceteams bei tiefergreifenden Automatisierungsthemen vollständig auf die Produktentwicklungsteams angewiesen. Wenn es darum geht, manuelle Prozesse durch Veränderungen am Produkt vollständig entfallen zu lassen, ist dies mit oft erheblichem konzeptionellem und technischem Aufwand verbunden.

Gleichzeitig sind die Produktentwicklungsteams für die Umsetzung von umsatzsteigernden Anforderungen verantwortlich. In diesem Konflikt zwischen mehr Umsatz und internen Effizienzsteigerungen ist es schwierig, die Zustimmung für die Umsetzung interner Automatisierungsthemen zu bekommen.

Deshalb ist es nachvollziehbar, so viel wie nur irgend möglich umgesetzt bekommen zu wollen, wenn man schon einmal „das große Los gezogen hat“. Damit einher geht ein bestenfalls rudimentäres Verständnis von Softwareentwicklung und den damit verbundenen Herausforderungen auf Seiten des Serviceteams. Das Argument, mit iterativ-inkrementellem Vorgehen das Risiko zu reduzieren und früher Wert zu liefern, klingt nach einer Ausrede des Product Owners, um das Thema klein zu halten und mehr Platz für umsatzbringende Themen in der Roadmap zu haben. Die (berechtigte) Vermutung auf Seiten des Serviceteams ist, mit dem ersten Inkrement abgespeist zu werden, welches nur einen Bruchteil der benötigten Funktionalität liefert, weil sich durch den Druck von B2B-Kund:innen, neue Features zu bekommen, die Prioritäten verschieben und an anderen Themen gearbeitet wird.

Aus dieser Sichtweise heraus ist es vernünftig, auf den maximalen Funktionsumfang zu bestehen, sobald man einmal den „Fuß in der Tür hat“.

Vermeidung bzw. Gegenmaßnahmen

Die Vermutung, nicht alle Wünsche erfüllt zu bekommen, ist absolut berechtigt. Nicht zuletzt ist die Möglichkeit, den Fokus zu wechseln, eines der Argumente für agile Softwareentwicklung.

Entscheidend ist eine Vereinbarung gemeinsamer Ziele, die sich auf den Business Value beziehen. Wenn es also darum geht, mit einer Automatisierung beispielsweise 50% weniger manuellen Aufwand pro Geschäftsvorfall zu haben, so sollte das gemeinsame Ziel dies ausdrücken: „Optimierung von Produkt X, so dass die manuellen Aufwände für Prozess Y um 50% reduziert werden (gemessen am ...)“.

Mit einer Verpflichtung aller beteiligten Bereiche auf dieses Ziel ist klar, dass die Teams so lange mit diesem Thema beschäftigt sind, bis die Erreichung des Ziels gemessen wird. Alle Beteiligten haben dabei ein Interesse, dieses Ziel mit so wenig Aufwand wie möglich zu erreichen, auch wenn dies gegebenenfalls eine Änderung der ursprünglichen Anforderungen bedeutet. Die Frage der Entwicklungsstrategie – ob es jetzt einen großen Release gibt oder mehrere kleine – tritt in den Hintergrund.

Bei der Formulierung des Ziels sind die Rahmenbedingungen zu beachten. Auch hier besteht die Gefahr, dass das Ziel zu groß gesetzt wird. Dauert es zu lange, das Ziel zu erreichen, wird der Druck beispielsweise durch eingehende Kundenanforderungen irgendwann so groß, dass das Management sich gezwungen sieht einzugreifen, die Arbeit am aktuellen Thema abubrechen und das Team auf andere Themen anzusetzen. Der Eintritt einer solchen Situation ist für alle Beteiligten unerfreulich, erscheint jedoch mitunter unvermeidlich.

Für die Festlegung der richtigen Größe gibt es keine eindeutige Formel. Eine Analyse vergangener Projekte kann aber helfen, zu erkennen, ab welcher Größe typischerweise Probleme anfangen.

Mit der Liste dieser gemachten Fehler wird klar, wie das Projekt in den aktuellen Zustand geraten ist. Dabei ist zu beachten, dass im Nachhinein vieles klarer scheint, als es in dem Moment war, wo die Fehler passiert sind, Stichwort *hindsight bias*.

Hindsight Bias

Hindsight Bias beschreibt einen psychologischen Trugschluss. Es geht um die menschliche Tendenz, vergangene Entscheidungen aus dem Blickwinkel jetzt verfügbarer Informationen zu beurteilen – inklusive des Wissens, welchen Verlauf die Ereignisse genommen haben.

Es stimmt häufig, dass zum Zeitpunkt einer möglicherweise fatalen Entscheidung Informationen vorlagen, die es, rückblickend betrachtet, möglich gemacht hätten, den Fehler zu erkennen und zu vermeiden. Zum tatsächlichen Zeitpunkt der Entscheidung waren diese Informationen jedoch nur ein kleiner Teil in der Flut von Daten und in ihrer (späteren) Signifikanz nicht erkennbar.

Zudem wirken rückblickend betrachtet Ereignisverläufe zwangsläufig, weil sie eben genau so eingetreten sind und nicht anders. In der Regel ist dies jedoch ein Trugschluss. Bereits minimal andere, zufällige Details hätten zu einem anderen Ausgang geführt.

Das Bewusstsein dieser Effekte ist oft nicht vorhanden und trägt dazu bei, vergangene Entscheidungen unangemessen zu beurteilen.

Wie konnte es so weit kommen?

Eine Betrachtung des Gesamtbildes offenbart, dass sich die Situation dadurch erklären lässt, dass entscheidende Grundsätze agiler Entwicklungsarbeit verletzt wurden.

Die Umsetzung und Lieferung waren wegen des Widerstands der Serviceabteilung nicht inkrementell angelegt. Dadurch war der tatsächliche Fortschritt bzw. Nicht-Fortschritt nicht erkennbar.

Der Arbeitsfortschritt war nicht transparent und die Arbeit erfolgte nicht iterativ, auch wenn sie formal in Sprints organisiert war. Zwar wurden Tickets als erledigt markiert und in Sprint Reviews vorgestellt. Aber mit diesen Tickets wurden jeweils nur technische Aspekte erfasst und kein für das Serviceteam greifbarer Wert. Die Entwicklung erfolgte unter der Annahme, dass die ursprünglichen Anforderungen korrekt und vollständig sind. Eine Überprüfung am laufenden System und insbesondere eine Integration mit den Arbeitsprozessen des Serviceteams konnte nicht erfolgen, weil es keine inkrementellen Zwischenergebnisse gab, die dies ermöglicht hätten.

Eine derartige Integration hätte die Defizite zwangsläufig und frühzeitig offenbart. Iterativ-inkrementelle Softwareentwicklung (mit Scrum) besteht eben nicht darin, eine vorher definierte Menge von Anforderungen über mehrere Sprints zu verteilen, sondern jeden Sprint ein in sich abgeschlossenes Inkrement zu liefern, das eine Beurteilung des inhaltlichen Fortschritts durch Fachabteilungen ermöglicht.

Die Planung von Produktinkrementen ist bei komplexen Softwareprodukten äußerst anspruchsvoll und nicht immer 100% möglich. Wie in diesem Fallbeispiel beschrieben, kann der investierte Aufwand jedoch den Unterschied zwischen effektivem agilen Arbeiten und der frühzeitigen Korrektur von Problemen auf der einen Seite *und* erheblicher Verschwendung von Zeit und Ressourcen mit dramatischen Konsequenzen auf der anderen Seite bedeuten.

Diese Schlussfolgerungen sind erschreckend einfach. Scrum wurde als Vorgehensmodell eingesetzt und beinhaltet mit der Sprint Review ein Element, das genau dazu dient, mit den Stakeholdern zusammen Produktinkremente zu inspizieren und aus dieser Analyse Schlussfolgerungen über den Projektverlauf und angemessenes weiteres Vorgehen abzuleiten.

In der Form, wie die Sprint Review gelebt wurde, wurde sie dieser Rolle nicht gerecht.

Kaum macht man es richtig, schon geht es

Ich habe weiter oben erwähnt, dass die Product Ownerin bereits Maßnahmen getroffen hatte, um das Projekt wieder auf Kurs zu bringen. Diese Maßnahmen waren:

- Erarbeitung einer gemeinsamen visuellen Repräsentation der entscheidenden Prozesse zwischen Produktentwicklungsteam und Fachteam als Grundlage aller Entscheidungen
- Durchsetzung einer inkrementellen Releasestrategie
- Regelmäßige Inspektion der Fortschritte mit den fachlichen Stakeholdern

Diese Maßnahmen adressieren genau die zuvor benannten Defizite, mit unmittelbar sichtbaren, positiven Resultaten.

Der Bericht

Den Abschluss der Untersuchung bildete die Präsentation der Ergebnisse beim Senior Management.

Die Präsentation erfolgte durch die hauptverantwortlichen Personen zusammen mit mir als Organisator der Analyse. Konkret präsentierten die aktuelle Product Ownerin, die Leiterin des Serviceteams, der zuständige Bereichsleiter des Produktmanagements und ich. Im Gremium vertreten waren die Leiter:innen aller Geschäftsbereiche und Funktionen.

Mir war es sehr wichtig, die direkt beteiligten Personen für die Präsentation der Analyse und Schlussfolgerungen zu gewinnen. Dieses Vorgehen erschien mir wesentlich besser geeignet, eine Rechtfertigungssituation zu vermeiden, als wenn ich gewissermaßen als Ankläger die gefundenen Defizite präsentiert hätte. Auf die gewählte Weise konnte eine Konfrontation vermieden werden. Es wurde ermöglicht, dass sich alle Beteiligten darauf konzentrieren, welche Verbesserungsmaßnahmen sich aus der Analyse ergeben und wie diese umgesetzt werden.

Die gemeinsame Präsentation durch Vertreter:innen des Produktentwicklungs- und Serviceteams war trotz aller Defizite konstruktiv. Damit konnte auch vermieden werden, dass der Eindruck einer einseitigen Sichtweise und der Versuch einer Schuldzuweisung entsteht.

Der Bericht enthielt die hier vorgestellten Analyseergebnisse und einen Überblick über den geplanten weiteren Verlauf, die Auswirkungen auf die Auslastung des Serviceteams und die verbleibenden Risiken. Des Weiteren wurden konkrete Empfehlungen formuliert, um ähnliche Situationen zu vermeiden.

Empfehlungen für künftige Projekte

Projektgröße im Auge behalten

Eine zentrale Empfehlung war, **Projekte über einer bestimmten Größe unbedingt zu vermeiden**. Der Verlauf dieses Projektes hatte anschaulich illustriert, dass mit zunehmender Größe die Gefahr steigt, dass Defizite im Vorgehen unerkant bleiben und sich potenzieren.

Ein weiterer Gesichtspunkt hinter dieser Empfehlung ist die Erkenntnis, dass das Umfeld so dynamisch ist und sich Prioritäten so häufig ändern, dass jenseits dieser Projektgröße der Druck, andere Themen zu bearbeiten, so groß wird, dass es zwangsläufig zu Unterbrechungen kommt.

Das bedeutet, dass für alle Projekte, die diese Größenordnung erreichen oder überschreiten, die inhaltliche Planung in einer Weise erfolgen muss, dass innerhalb dieser Zeit Zwischenergebnisse produziert werden, die getestet werden können.

Die konkrete Obergrenze ist abhängig von den Rahmenbedingungen, so dass ich keinen allgemeingültigen Wert nennen kann.

Im Fall dieses Unternehmens habe ich als Obergrenze für einzelne Projekte eine Dauer von drei Monaten empfohlen. Dass sich die Projekte auf diese Weise elegant in den ebenfalls quartalsweisen OKR Prozess einfügen, war dabei ein wichtiger Gesichtspunkt.

Priorisierung mit bereichsübergreifenden OKRs

Die zweite Empfehlung war die **Priorisierung solcher Projekte in Form von bereichsübergreifenden OKRs**. Auf diese Weise wird sichergestellt, dass der gesamten Organisation die Bedeutung des Projekts bewusst ist und für alle notwendigen Teams die Beteiligung an dem Projekt Priorität hat.

Durch die Formulierung von gemeinsamen Zielen wird sichergestellt, dass eine einheitliche Vorstellung vom Zweck des Projekts herrscht.

Die Ziele, konkreter die Key Results im OKR-Kontext, müssen dabei den Business Value in messbarer Form beinhalten. Im konkreten Fall hätte das beispielsweise so aussehen können: „Mit der Automatisierung des Prozesses X entfällt für Y Geschäftsvorfälle pro Monat die manuelle Bearbeitung (von Z Minuten pro Vorfall). Dadurch werden ZZ Mitarbeiterkapazitäten eingespart.“ So wird sichergestellt, dass das Projekt immer den angestrebten Zweck im Blick behält und überprüft werden kann, inwieweit das Projekt tatsächlich die angestrebten Effekte hat.

Die empfohlene Projektobergrenze von drei Monaten ergänzt sich dabei praktisch mit dem ebenfalls dreimonatigen OKR-Zyklus. Mit der Einhaltung dieser beiden Empfehlungen ist es unter Nutzung des OKR-Prozesses ohne zusätzlichen Aufwand möglich, die Ergebnisse des Projekts im Rahmen der OKR Review zu evaluieren und bei Bedarf eine weitere Iteration einzuplanen oder den Fokus auf ein anderes Thema zu legen.

Feedback des Vorstands

Die Präsentation wurde sehr positiv aufgenommen und die Schlussfolgerungen bestätigt.

Besonders gefreut habe ich mich über die Reflexion des für den Service zuständigen Vorstands zu dem bisherigen Drängen auf „Alles oder Nichts“. Es scheint, dass wir den Zusammenhang von diesem Vorgehen und den entstandenen Problemen nachvollziehbar illustriert haben. Er hat zugestimmt, künftig auch von seiner Seite immer auf eine inkrementelle Planung zu achten.

Überraschend und sehr erfreulich war auch die Anmerkung eines anderen Vorstandsmitglieds im Hinblick auf die vielen Veränderungen im Team, welche zu den Problemen beigetragen haben. Er merkte an, dass bei ihnen auf der Senior-Managementebene bei Ressourcenplanungen manchmal entgeht, dass hinter den Boxen auf den Powerpoint-Folien konkrete Personen stehen und deren Hin- und Herbewegung negative Konsequenzen auf die Teams nach sich ziehen.

Fazit

In diesem Artikel habe ich gezeigt, wie auch bei scheinbar etabliertem Einsatz agiler Methoden, die Nichtbeachtung fundamentaler agiler Prinzipien zu genau denjenigen Problemen führt, zu deren Vermeidung agile Methoden ursprünglich einmal entwickelt wurden. Isolierte, zum Zeitpunkt ihres Entstehens auch für erfahrene Praktiker:innen schwer zu erkennende Fehler können sich leicht potenzieren und ein Projekt aus der Spur bringen. Und das obwohl alle Beteiligten kompetent sind und mit bester Absicht handeln.

Es kommt darauf an, die Intention der eingesetzten Methoden zu verstehen und ihre Anwendung konstant zu hinterfragen.

Die gute Nachricht: Eine Korrektur ist möglich. Im konkreten Fall brauchte es eine engagierte und erfahrene Product Ownerin, Mut, guten Willen und ein paar intensive Workshops, um das Projekt wieder auf Spur zu bringen und die bisherigen Fehler im Vorgehen zu korrigieren.

Hat Ihnen dieser Artikel gefallen?

Bewerten und kommentieren Sie den Artikel auf projektmagazin.de!

[> zum Artikel](#)