
Einleitung:

Die Geschichte beginnt

In der Firma Titanic Software A/S, einem fiktiven dänischen Unternehmen, das auf Navigationssoftware für Schiffe spezialisiert ist, müssen Veränderungen vorgenommen werden. Die Kundschaft beschwert sich über Fehler in der Software, die ihre Schiffe zu falschen Zielen fahren lassen, und die Geschwindigkeit in der Softwareentwicklung ist nicht zufriedenstellend.

Der Chief Technology Officer (CTO) und einige Entwickler*innen beschließen, eine Softwarekonferenz zu besuchen, um zu erfahren, wie andere Unternehmen vorgehen, um der Zeit einen Schritt voraus zu sein. Sie stoßen auf einen neuen Ansatz, die sogenannte agile Softwareentwicklung, die verspricht, die Prozesse zu beschleunigen, Fehler zu reduzieren und die Software besser auf die Bedürfnisse der Kundschaft auszurichten. Mit anderen Worten, dieser neue Ansatz könnte sie dahin bringen, wo sie gerne hin wollen – reich und glücklich sein.

Der CTO schickt alle zu einem Kurs für Scrum Master*innen. Sehen wir uns dazu ein kleines Team an, das aus sechs Personen besteht: Bo, Peter, René, Kim, Sarah und Andrea. Sarah, die zuvor Projektleiterin war, ist diejenige, die am liebsten Scrum Masterin werden möchte, und da Begeisterung eine wichtige Eigenschaft bei der Besetzung dieser Rolle ist, wird sie auch zur Scrum Masterin des Teams ernannt.

Die Berufsbezeichnungen müssen an die agile Arbeitsweise angepasst werden. Peter, der bisher Business-Analyst war, wird zum Product Owner. In ihrer neuen Rolle als Scrum Masterin ist Sarah u. a. dafür verantwortlich, dass das Daily-Standup-Meeting stattfindet. Wenn während des Daily Standup Probleme auftauchen, die ein Gespräch mit anderen im Unternehmen erfordern, kümmert Sarah sich darum, dass dies auch geschieht. Als Scrum Masterin ist sie außerdem Zeremonienmeisterin und sorgt als solche dafür, dass das Team regelmäßig Retrospektiven abhält.

Strukturelle Antipatterns

Das Glücksrad (Wheel of Fortune) ... wenn das Team in der Retrospektive voreilige Schlüsse zieht, indem es Symptome statt Probleme löst, und die moderierende Person die Teammitglieder dazu bringt, sich Zeit zu nehmen für die Ursachensuche hinter den Symptomen.

Die oberste Direktive ignorieren (Prime Directive Ignorance) ... wenn die Teammitglieder die oberste Direktive ignorieren – *»Unabhängig davon, was wir entdecken werden, verstehen und glauben wir aufrichtig, dass in der gegebenen Situation, mit dem verfügbaren Wissen, den Ressourcen und ihren individuellen Fähigkeiten, jede Person ihr Bestes getan hat«* [Kerth 2001] –, weil sie es lächerlich finden, und die moderierende Person sie daran erinnert, wie wichtig diese Denkweise für eine erfolgreiche Retrospektive ist.

Die Ursuppe (In the Soup) ... wenn die Teammitglieder Dinge diskutieren, die sie nicht ändern können, und die Person, die moderiert, ihnen hilft, ihre Energie auf das zu konzentrieren, was sie ändern können, und zu akzeptieren, was sie nicht ändern können.

Die Zeit überziehen (Overtime) ... wenn das Team bei der Retrospektive abschweift und über eine Entwicklung spricht, die für das Team als Ganzes nicht die wichtigste ist, und die moderierende Person dem Team hilft, sich wieder auf das Wesentliche zu konzentrieren.

Teil 1

Smalltalk (Small Talk) ... wenn die Teammitglieder Zeit mit Smalltalk in kleinen Gruppen verbringen, anstatt sich auf den Austausch und das Lernen zu konzentrieren, und die Person, die moderiert, die Aktivitäten ändert, damit sie wieder als Team zusammenarbeiten.

Erschwerende Demokratie (Unfruitful Democracy) ... wenn zur Frustration der Minderheit im Team demokratisch entschieden wird, was zu diskutieren und zu tun ist, und die moderierende Person andere Wege der Entscheidungsfindung definiert, die dann alle zufriedenstellt.

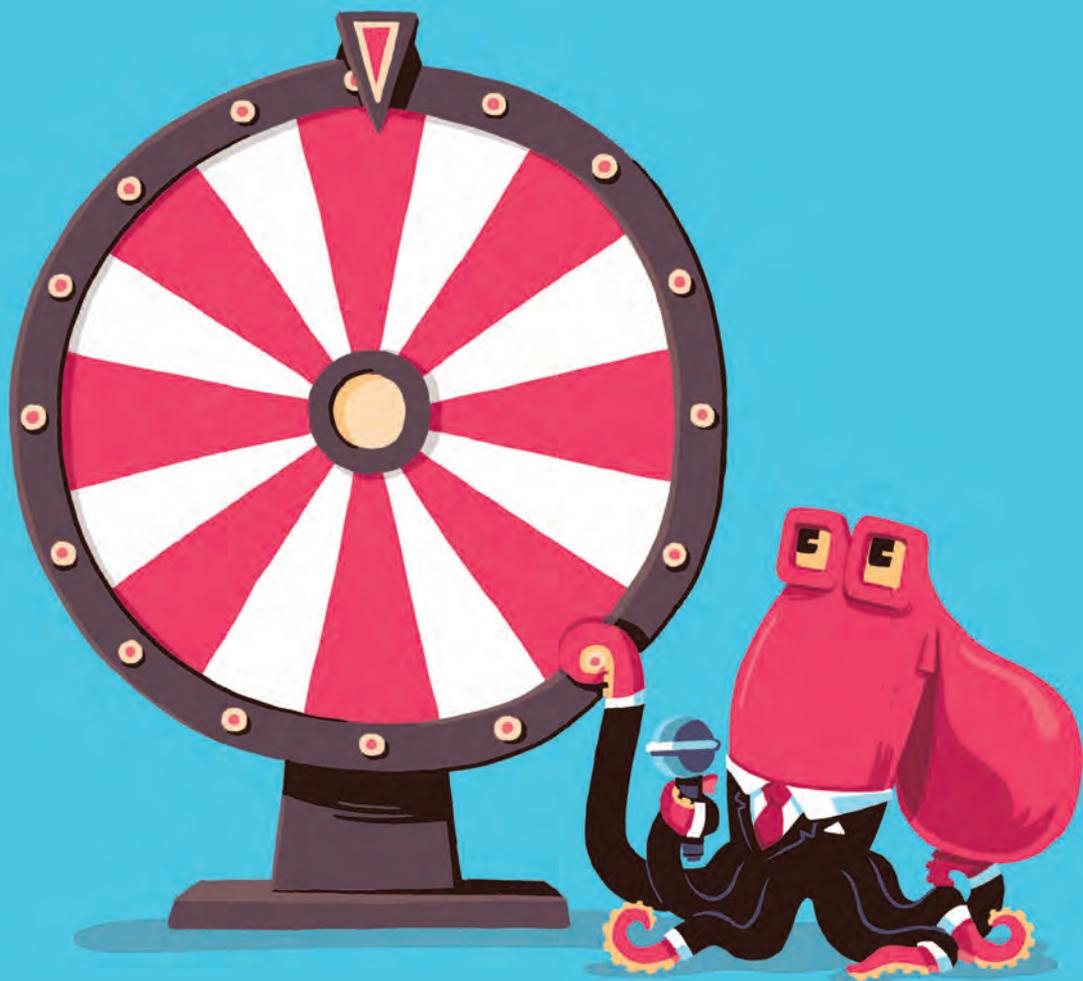
Nichts zu besprechen (Nothing to Talk About) ... wenn das Team glaubt, so gut geworden zu sein, dass es keine Retrospektiven mehr braucht, und die moderierende Person dem Team zeigt, wie es lernen kann, sich weiter zu verbessern.

Politische Abstimmung (Political Vote) ... wenn die Teammitglieder bis zum letzten Moment mit der Abstimmung warten, um das System zu manipulieren, und die moderierende Person einen Weg findet, das Abstimmungssystem fairer zu gestalten.

Kapitel 1

... wenn das Team in der Retrospektive voreilige Schlüsse zieht, indem es Symptome statt Probleme löst, und die moderierende Person die Teammitglieder dazu bringt, sich Zeit zu nehmen für die Ursachensuche hinter den Symptomen.

Das Glücksrad



Kontext

Bei der Planung ihrer ersten Retrospektive hat Sarah nicht viel Zeit. Sie muss eine Anleitung für die Moderation der Retrospektive finden, und ihre einzige Quelle, wie man eine Retrospektive moderiert, stammt aus dem Kurs für Scrum Master*innen. Der Kurs deckte alle Komponenten des Scrum-Frameworks ab, aber aufgrund seines breiten Umfangs war die Zeit, die für die Vermittlung eines einzelnen Themas, einschließlich der Moderation von Retrospektiven, aufgewendet wurde, begrenzt. Sarah entscheidet sich für die Methode »Anfangen-Aufhören-Weitermachen«, bei der das Team überlegt, womit es beginnen, womit es aufhören und womit es fortfahren soll. Sarah verwendet diese Aktivität als Kernstück der Retrospektive, und in der Phase »*Entscheiden, was zu tun ist*« stimmt das Team über die drei verschiedenen Themen ab und findet Dinge, mit denen es beginnen, aufhören und weitermachen möchte.

Anfangen-Aufhören-Weitermachen

Zur Vorbereitung dieser Aktivität erstellt die Person, die moderiert, drei Plakate auf Flipchart-Papier oder Whiteboards, eines mit der Überschrift »Anfangen«, eines mit der Überschrift »Aufhören« und eines mit der Überschrift »Weitermachen«. Die Teammitglieder erhalten Post-it-Zettel und Stifte. Einige ziehen es vor, dass alle Zettel dieselbe Farbe haben; andere bevorzugen Zettel in drei verschiedenen Farben, z.B. in Ampelfarben, damit die Zettel von den Boards entfernt werden können, ohne dass der Bezug zu einer der drei Möglichkeiten verloren geht. Die moderierende Person kann die Teilnehmenden auch auffordern, »Anfangen«, »Aufhören« oder »Weitermachen« oben auf die jeweiligen Zettel zu schreiben. Die Teammitglieder haben etwa 10 Minuten Zeit, um auf die Post-its zu schreiben, was ihrer Meinung nach begonnen, gestoppt und fortgesetzt werden sollte. Anschließend kleben sie die Post-its auf die vorbereiteten Boards. Das Team muss nun auf der Grundlage dieser Notizen seine Aktionspunkte auswählen, und wie dies geschieht, hängt sehr stark von der moderierenden Person, dem Inhalt der Post-it-Zettel und den Teammitgliedern ab.

Bei Sarahs erster Retrospektive hängt an dem »Anfangen«-Board ein Post-it-Zettel mit der Aufschrift »mehr Pair Programming¹«. Das lässt sich leicht verwirklichen: Das Team erstellt einfach einen Zeitplan, der jeden Tag drei Stunden Pair Programming vorsieht. Nach der Retrospektive sind alle zufrieden, aber im nächsten Sprint stellen sie fest, dass der Umfang des Pair Programming nicht wesentlich zugenommen hat. In der ersten Woche haben Bo und Peter etwas Pair Programming gemacht – allerdings nicht die 15 Stunden, die der neue Zeitplan vorsah. Kim und René haben sogar noch weniger gemeinsam im Pair Programming erar-

1. Anm. d. Übers.: Programmieren in Paaren, wobei zwei Entwickler*innen gleichberechtigt an einer Aufgabe arbeiten

beitet. Beide scheinen sich damit unwohl zu fühlen, sodass niemand von ihnen damit beginnen möchte.

Metakontext

Als Moderator* in einer Retrospektive treffen Sie manchmal auf Menschen, die der Meinung sind, dass eine Retrospektive Zeit von der eigentlichen Arbeit abzieht. Dieser Standpunkt könnte Sie dazu verleiten, so wenig Zeit wie möglich für die Retrospektive aufzuwenden, um alle zufriedenzustellen, sodass eine »Alles schnell beheben«-Aktivität die beste Idee zu sein scheint.

Antipattern-Lösung

Am einfachsten lässt sich eine Retrospektive durchführen, indem man drei Plakate mit den Überschriften Anfangen, Aufhören und Weitermachen aufhängt. Der nächste Schritt besteht darin, dass das Team ein Brainstorming durchführt, etwas auf Post-it-Zettel schreibt und die Zettel auf das entsprechende Plakat klebt. Der letzte Schritt ist noch einfacher. Sie beginnen mit den Punkten, die auf dem Anfangen-Plakat aufgelistet sind, hören mit den Punkten auf dem Aufhören-Plakat auf und fahren mit den Punkten auf dem Weitermachen-Plakat fort. Die Retrospektive ist in 15 Minuten beendet, und das Team kann sich wieder seiner »eigentlichen« Arbeit widmen. Selbst wenn Sie ein »Mache mehr davon«-Plakat hinzufügen würden, wären alle in kürzester Zeit wieder an ihren Schreibtischen.

Eigentlich ist dieser Ansatz nicht der einfachste Weg, eine effektive Retrospektive durchzuführen. Ich habe schon von »Retrospektiven« gehört (und ich verwende den Begriff »Retrospektive« hier sehr weit gefasst), bei denen das Team in einen Besprechungsraum ging, die Person mit der Verantwortlichkeit Scrum Master oder Projektleitung fragte, ob etwas geändert werden müsse, und als niemand etwas sagte, gingen alle wieder an die Arbeit.

Konsequenzen

Der Vorteil dieser Antipattern-Lösung besteht darin, dass sie eine sehr schnelle Methode zur Durchführung einer Retrospektive darstellt, aber die negativen Folgen können schwerwiegend sein. Manchmal hat man Glück und die Notizen auf den Post-its zeigen die wirklichen Probleme auf. Um sie zu beheben, hört das Team einfach mit den »Aufhören«-Elementen auf und widmet sich verstärkt den »Mach mehr davon«-Elementen. Oft beschreiben die Post-its jedoch nur die Symptome größerer Probleme, die eine grundlegendere Veränderung erfordern, als es die bloße Behebung der Symptome vermag. Daher sollte verhindert werden, dass die Symptome überhaupt auftreten, indem Sie das zugrunde liegende Problem beseitigen.

In unserem Beispiel von Titanic Software A/S funktioniert dieses Antipattern gut, wenn der einzige Grund dafür, dass nicht mehr im Pair Programming gearbeitet wird, darin besteht, dass die Leute es einfach vergessen und zu früheren Gewohnheiten zurückkehren. Die Schwierigkeiten des Teams können jedoch durchaus ein Symptom für ein interessanteres und herausfordernder zu lösendes Problem sein.

Hätten die Teammitglieder während der Retrospektive Zeit damit verbracht, die Ursache des Problems zu erforschen, hätten sie vielleicht die Gründe dafür herausgefunden, warum sie weniger im Pair Programming gearbeitet haben, als gemeinsam beschlossen wurde. Vielleicht konnten die Entwickler*innen den Nutzen nicht erkennen und wurden nie gefragt, ob sie Pair Programming machen wollten. Oder sie sahen zwar den Nutzen, aber die meisten von ihnen waren introvertiert und brauchten daher Stunden für sich allein, um über Informationen nachzudenken, bevor sie sie mit ihren Kolleg*innen teilten. Ein anderer Grund könnte sein, dass die Teammitglieder nicht wissen, wie Pair Programming funktioniert, also müssen sie erst lernen, wie das geht, bevor sie damit anfangen können. In unserem kleinen Team, in dem René eher ein *Großmaul* und Kim eine *Schweigsame* ist, könnte auch ein Mangel an psychologischer Sicherheit vorliegen. Keines dieser Probleme lässt sich allein dadurch lösen, dass man Menschen zur Zusammenarbeit zwingt.

Symptome

Wenn Sie Dinge hören wie »Warum reden wir immer wieder in der Retrospektive darüber?« und »Die Retrospektiven funktionieren nicht – nichts ändert sich außer der Art und Weise der Problembeschreibung«, dann sollten Sie sich fragen, ob Sie sich in einem *Glücksrad* befinden.

Wie bei einem echten Glücksrad erzielen Sie manchmal einen Gewinn, wenn Sie das Rad drehen – vorausgesetzt, das Rad landet auf dem eigentlichen Problem und nicht auf den Symptomen des Problems. Und die Wahrscheinlichkeit, dass das passiert, ist ungefähr so groß wie beim Drehen eines Glücksrads auf einem Jahrmakkt. Daraus resultiert als weiteres Symptom dieses Antipatterns, dass man bei den Retrospektiven immer wieder dieselben Themen diskutiert, und zwar aus dem einfachen Grund, dass die Probleme nie gelöst, sondern nur Pflaster auf die Symptome geklebt werden.

Refactoring-Lösung

Es ist verlockend, direkt von den Problemen zu den Lösungen überzugehen, und die meisten Entwickler*innen sind darauf trainiert, genau das zu tun. Leider werden die Probleme, die bei den Retrospektiven auftauchen, manchmal nicht sofort verstanden, sodass man sie untersuchen muss, bevor mit der Arbeit an den Lösungen begonnen werden kann. Nachdem die Daten gesammelt wurden, müssen Sie die Ursachen für die Probleme analysieren. Diese Phase wird, wie in der Einleitung beschrieben, als »*Erkenntnisse gewinnen*« bezeichnet und kann nicht übersprungen werden.

In dieser Phase können mehrere Aktivitäten durchgeführt werden. Eine einfache Möglichkeit besteht darin, nach der Geschichte hinter der Notiz auf dem Post-it zu fragen, um zu erfahren, was zu diesem Problem geführt hat. Andere Aktivitäten sind das Fishbone-Diagramm² (siehe Abb. 1–1 inkl. Beschreibung) und die »5 How«. Früher habe ich für die Ursachenanalyse die »5 Why³«-Methode verwendet, aber aufgrund der Erkenntnisse, die ich durch John Allspaw [Allspaw 2014] gewonnen habe, verwende ich jetzt »5 How«.

»5 How«

»5 How« ist eine iterative Fragetechnik, die dazu dient, die einem bestimmten Problem zugrunde liegenden Ursache-Wirkungs-Beziehungen zu untersuchen. Das Hauptziel dieser Technik besteht darin, die Ursache eines Fehlers oder Problems zu ermitteln, indem wiederholt gefragt wird: Wie ist es dazu gekommen? Jede Antwort bildet die Grundlage für die nächste Frage. Die Fünf im Namen leitet sich von einer anekdotischen Beobachtung über die Anzahl der Iterationen ab, die zur Lösung des Problems erforderlich sind.

Nur wenige Probleme haben eine einzige Ursache. Um mehrere Grundursachen aufzudecken, kann die Methode wiederholt werden, wobei jedes Mal eine andere Abfolge von Fragen gestellt wird.

Die Methode enthält keine festen Regeln darüber, wie die Reihenfolge der Fragen formuliert werden sollen oder wie lange die Suche nach weiteren Ursachen fortgesetzt werden soll. Selbst wenn die Methode genau befolgt wird, hängt das Ergebnis also immer noch vom Wissen und der Beharrlichkeit der beteiligten Personen ab.

John Allspaw erklärt, dass die Frage nach dem Warum dazu führen kann, dass man versucht, eine einzige Ursache für ein Problem zu finden, und sogar in Schuldzuweisungen enden kann [Allspaw 2014]. Die Frage nach dem »Wie« könnte stattdessen zu einer Geschichte und einer Reihe von Ursachen führen.

2. Anm. d. Übers.: Fishbone (engl.) bedeutet Gräte; die Bezeichnung Fishbone- oder Ishikawa-Diagramm wird im Deutschen analog zu Ursache-Wirkungs-Diagramm verwendet, siehe auch https://en.wikipedia.org/wiki/Ishikawa_diagram oder <https://de.wikipedia.org/wiki/Ursache-Wirkungs-Diagramm>.
3. Anm. d. Übers.: Bei dieser Methode werden fünf Warum-Fragen gestellt, um eine Ursache für ein Problem zu identifizieren.

Eine Schuldzuweisung ist natürlich nicht erwünscht, und die Suche nach einer einzigen Ursache in einem so komplexen Bereich wie der Softwareentwicklung und der Teamarbeit ist optimistisch und grenzt an Leichtgläubigkeit. Sie könnten dies sogar auf einer persönlichen Ebene tun: Versuchen Sie selbst das nächste Mal, wenn etwas nicht so läuft, wie Sie es sich vorgestellt haben, nach dem »Wie« und nicht nach dem »Warum« zu fragen.

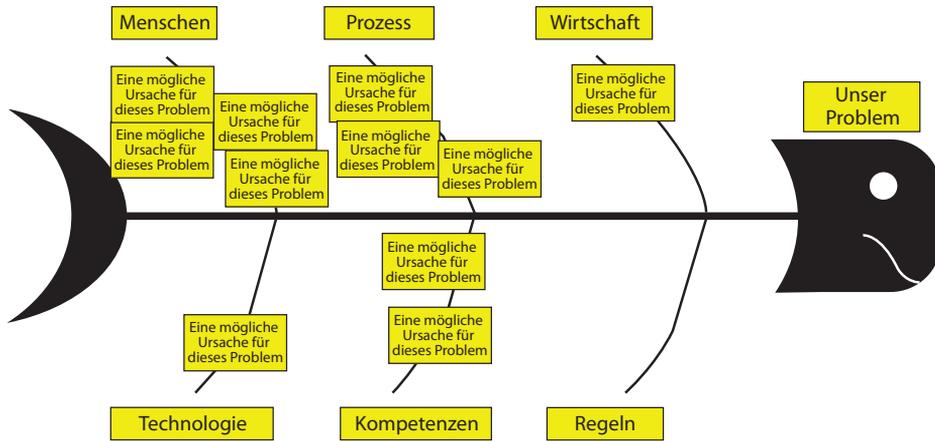


Abb. 1-1 Fishbone-Diagramm zur Ermittlung mehrerer Ursachen

Fishbone

Ein *Fishbone* ist eine Aktivität zur Ursachenanalyse, bei der in der Moderation ein Fischskelett mit verschiedenen Themenbereichen wie Menschen, Prozesse, Wirtschaft und Technologie als Fischgräten gezeichnet wird, um die verschiedenen Arten von Ursachen für ein Problem zu visualisieren. Dann bittet die Person, die moderiert, die Teammitglieder auf Post-it-Zettel die verschiedenen Ursachen aufzuschreiben, die ihnen einfallen, und sie auf der Fischgräte zu platzieren. Das Ergebnis ist ein Fisch, der viele verschiedene Ideen über die mögliche Ursache des Problems zeigt. Vielleicht stellen Sie fest, dass sich die meisten Ideen in einer bestimmten Kategorie bündeln, z.B. Prozess oder Technologie, und das gibt Ihnen Hinweise, an welcher Ursache Sie arbeiten sollten. Diese Technik wird im Allgemeinen in der Phase »Erkenntnisse gewinnen« einer Retrospektive eingesetzt. Sie wird auch als *Ishikawa-Diagramm* oder *Ursache-Wirkungs-Diagramm* bezeichnet.

Alle Aktivitäten der Phase »Erkenntnisse gewinnen« zur Ursachensuche beinhalten eine Vertiefung dessen, was auf den Post-it-Zetteln steht, um die wahren Geschichten dahinter zu finden.

Jedes Meeting, auch eine Retrospektive, hat einen Lebenszyklus von Divergenz und Konvergenz⁴ (siehe Abb. 1–2), wie im Buch *Facilitator's Guide to Participatory Decision-Making* von Sam Kaner [Kaner 2007] beschrieben. Wir beginnen hoffentlich in einem kleinen Raum, in dem wir uns über den Zweck des Treffens und die erwarteten Ergebnisse im Klaren sind; zumindest über das Thema des Treffens sollte man sich einig sein. Dann kommen wir in die Divergenzphase, in der die Diskussionen zu Meinungsverschiedenheiten und Meinungskonflikten führen. Als Nächstes kommt die Diskussionsphase, in der die Teilnehmenden weiter diskutieren, nicht um eine Einigung zu erzielen, sondern um mehr über das Thema zu erfahren. Nach einer gewissen Zeit für die Diskussionsphase muss das Treffen in eine Konvergenzphase übergehen, in der wir versuchen, den Umfang einzugrenzen und/oder eine Einigung zu erzielen. Wenn es zu keiner Einigung kommt, kann sich die Gruppe zumindest darauf einigen, nicht einer Meinung zu sein, und die Bedingungen und Gründe für die Uneinigkeit festlegen.

Zu Beginn ähnelt die Einigung auf das Thema der Phase »*Den Rahmen schaffen*« der Retrospektive. Der Divergenzteil einer Retrospektive ist die Phase des Sammelns von Daten und der Gewinnung von Erkenntnissen. Dies ist kein einfacher, aber ein sehr wichtiger Teil einer Retrospektive. Die Diskussionsphase kann mit dem letzten Teil der Phase »*Erkenntnisse gewinnen*« und dem Beginn der Phase »*Entscheiden, was zu tun ist*« zusammenhängen. Die Konvergenz ist der Kern der Phase »*Entscheiden, was zu tun ist*«, und wenn Sie am Ende der Retrospektive eine Entscheidung getroffen haben, schließen Sie die Retrospektive ab. Achten Sie darauf, alle fünf Phasen einer Retrospektive zu durchlaufen und Erkenntnisse zu gewinnen, anstatt voreilige Schlüsse zu ziehen und so zu einer verfrühten Konvergenz zu gelangen, nur um schnell zum Punkt zu kommen.

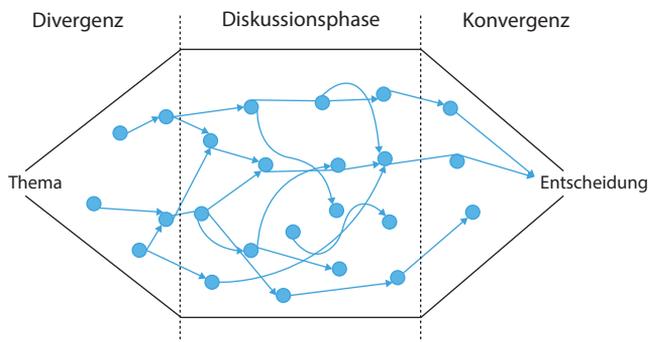


Abb. 1–2 Divergenz und Konvergenz in einem Meeting

4. Anm. d. Übers.: Divergenz = Unterschiedlichkeit/Widersprüchlichkeit, Konvergenz = Annäherung, Übereinstimmung, siehe auch das »Diamant-Modell« von Sam Kaner

Online-Aspekt

Wenn die Retrospektive online durchgeführt wird, können Sie ein Online-Tool verwenden, das das Team dazu zwingt, bei der Phase »Erkenntnisse gewinnen« zu beginnen und dort zu bleiben, bevor es zur Problemlösung übergeht. Wenn Sie Ihr eigenes Online-Dokument erstellen, können Sie jederzeit visualisieren, in welcher Phase der Retrospektive Sie sich befinden. Ich verwende oft ein mit »Google Drawings« erstelltes Diagramm für die Phase »Erkenntnisse gewinnen« und streiche das Feld für Vorschläge durch. Auf diese Weise werden die Teammitglieder visuell daran erinnert, keinen Vorschlag für ein Experiment hinzuzufügen, bevor wir die Phase »Entscheiden, was zu tun ist« erreichen (siehe Abb. 1–3).

Sie können die Phasen der Retrospektive auch visualisieren, indem Sie für jede Phase einen anderen Hut aufsetzen. Wenn die Teilnehmenden weiterhin vergessen, dass sie keine voreiligen Schlüsse ziehen sollen, können Sie lächelnd auf Ihren Hut zeigen.

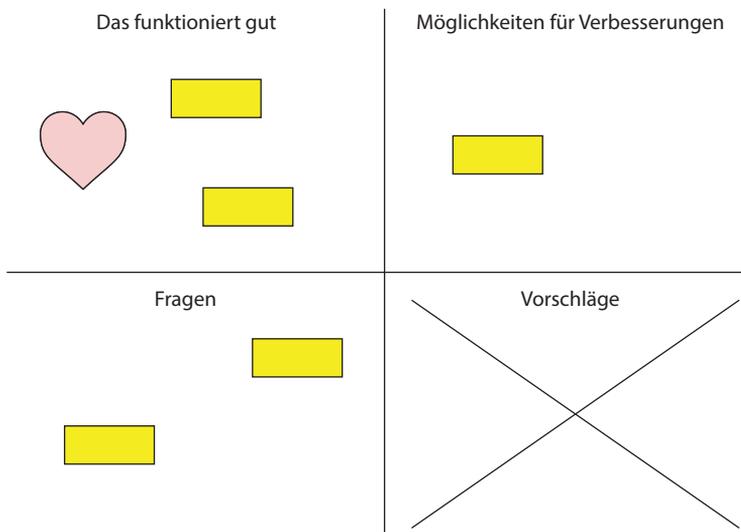


Abb. 1–3 Ein Online-Retrospektiven-Dokument mit durchgestrichenem Feld für Vorschläge

Persönliche Anekdote

Ich leitete ein Team in einem Unternehmen in Dänemark, das von einer großen ausländischen Organisation übernommen worden war. Das Unternehmen hatte seine eigene Art, wie es Dinge erledigte, und mochte es auch so. Leider beschloss die neue Organisation, das ursprüngliche Unternehmen in drei Teams aufzuteilen. Bei einer der Retrospektiven, die nicht lange nach der Übernahme stattfand, wurde unter anderem »Weniger Arbeit in Silos« auf die Tagesordnung gesetzt.

Wir alle verstanden, was damit gemeint war: Die Teammitglieder wollten zusammenarbeiten, anstatt als drei separate Teams parallel zu arbeiten, ohne sich gegenseitig zu helfen.

Da wir das Problem verstanden hatten, wurde ohne große Diskussion beschlossen, diesen Aktionspunkt bei dieser Retrospektive umzusetzen. Und bei der folgenden Retrospektive und der danach. Schließlich beschlossen wir, die Ursache für das Silo-Arbeiten zu untersuchen. Es stellte sich heraus, dass die für jedes Team festgelegten Leistungsindikatoren der Zusammenarbeit im Wege standen. Da jedes Team seine eigenen Ziele verfolgte, wurde es zu einem Nullsummenspiel, bei dem der Gewinn des einen Teams auf dem Verlust des anderen Teams beruhte. Auch die Planungsmeetings für die Teams fanden mit verschiedenen Personen in der neuen Organisation statt. Es wäre gut gewesen, dies von Anfang an erkannt zu haben, um zu verstehen, ob dies etwas war, das sie ändern konnten, oder etwas, mit dem sie lernen mussten, umzugehen (siehe auch Kap. 3 »Die Ursuppe«).

Nun zu etwas ganz anderem: Ich arbeite hauptsächlich mit IT-Leuten und finde es immer schwierig, die Menschen dazu zu bringen, in der offenen Diskussion oder in der Diskussionsphase zu bleiben. Es fühlt sich schon fast körperlich schmerzhaft an, diese Phase der Retrospektive offenzuhalten, bevor ich ihnen erlaube, sich in die Lösungsphase zu stürzen. Meiner Erfahrung nach ist jedoch jede investierte Minute in der offenen Diskussionsphase wertvoll. Hier kommen oft neue Einsichten und Ideen zum Vorschein. Dave Snowden hat oft über verfrühte Konvergenz als Teil der Komplexitätstheorie geschrieben, und er erläutert, was uns dabei entgehen könnte: »Dinge offenzuhalten, ihnen erlauben, in feingranularere Objekte zu verfallen, und dann zu sehen, wie sie sich wieder zusammenfügen und gemeinsam entwickeln« [Snowden 2015].

Als ich schon viel Erfahrung mit IT-Teams hatte, wurde ich gebeten, die Mitarbeitenden eines Kunstmuseums dabei zu unterstützen, Ideen zu generieren. Ich war nicht auf das vorbereitet, was dann passierte. Es war extrem einfach, die Teilnehmenden in die offene Diskussionsphase zu bringen, in der alles möglich ist und in der sie sich nicht auf eine Lösung oder einen Weg festlegen sollten. Ich hatte das Gefühl, dass es sehr gut lief und dass es mir gelungen war, sie an diese Arbeitsweise heranzuführen.

Aber dann begannen meine Probleme. Es war fast unmöglich, sie dazu zu bringen, Entscheidungen zu treffen, Ideen einzuschränken und Möglichkeiten auszuschließen. Ich erappte mich dabei, wie ich von Gruppe zu Gruppe lief und Post-it-Zettel vom Board nahm, die sie in dem Moment, in dem ich mich einer anderen Gruppe zuwandte, wieder anbrachten. Als ich jemanden sagen hörte: »Benötigen wir überhaupt ein Kunstmuseum?«, verspürte ich leichte Panik, denn wir waren hier, um ein Brainstorming für eine App durchzuführen und nicht, um eine existierende App zu verbessern.

tenzielle Diskussion darüber zu führen, ob wir ein Kunstmuseum brauchen. Ich musste mich auf dieses neue Publikum einstellen, und zwar schnell.

Für mich fühlte es sich fast so an, als würde ich mit einer anderen Spezies zusammenarbeiten. Ich musste viel direkter sein als bei anderen Gruppen, mit denen ich bisher gearbeitet hatte, und ich musste mehr Energie aufwenden, um diese Gruppe dazu zu bringen, in die letzte Phase eines Meetings zu gehen und zu einer Art Konvergenz zu kommen. Am Ende gelang es uns nur durch das Versprechen, dass die anderen Ideen für die Zukunft aufbewahrt werden, uns auf einen Schwerpunkt für die nächste Phase zu einigen. Es ist interessant, wie kulturell abhängig dieses Verhalten oder Denkmuster ist. Dies wird auch in Kapitel 23 »*Unterschiedliche Kulturen*« beschrieben. In manchen Kulturen versucht man, Konflikte oder Meinungsverschiedenheiten zu vermeiden. Menschen aus einer solchen Kultur (und das kann innerhalb eines bestimmten Landes, Unternehmens oder Teams der Fall sein) werden so schnell wie möglich einen Konsens anstreben und damit vielleicht die Chance verpassen, dass aus ihrer Interaktion neue Ideen entstehen.

Kapitel 7

*... wenn das Team glaubt, so gut geworden zu sein,
dass es keine Retrospektiven mehr braucht, und die
moderierende Person dem Team zeigt, wie es lernen
kann, sich weiter zu verbessern.*

Nichts zu besprechen



Kontext

Das Team macht seine Sache wirklich gut. Den Teammitgliedern gefällt die agile Arbeitsweise, und ihr Arbeitsprodukt bringt den Nutzenden durchweg einen Mehrwert. In den letzten beiden Retrospektiven gab es keine Probleme zu lösen, und alle Post-it-Zettel waren grün, bis auf ein paar über schlechtes Wetter (es ist schließlich Dänemark). Sarah hört zufällig, wie sich René und Andrea darüber unterhalten, ob Sarah von ihnen erwartet, dass sie Probleme erfinden.

Es scheint, dass die Retrospektiven nicht mehr nötig sind, da alles gut läuft. Sarah fordert das Team auf, gründlicher nachzudenken, ob es Probleme gibt, die angegangen werden müssen, aber ohne Erfolg. Obwohl sie frustriert ist, muss Sarah dem Rest des Teams zustimmen, dass die Retrospektiven vielleicht nicht mehr erforderlich sind.

Metakontext

Ich habe dieses Muster des Lebenszyklus von Retrospektiven in fast jeder Organisation und in jedem Team, in dem ich Retrospektiven eingeführt habe, beobachtet: Zunächst könnte das Team die Retrospektiven als Zeitverschwendung empfinden – bis es lernt, wie hilfreich und effizient (und unterhaltsam) sie sein können. Dann kommt eine Phase, in der die Teammitglieder viele Dinge finden, über die sie während der Retrospektiven sprechen können. Es fühlt sich gut an, etwas gegen die Probleme in der Teamarbeit, der Kommunikation, der Technologie oder dem System, an dem gearbeitet wird, zu unternehmen. Doch nach einiger Zeit gehen dem Team »die Probleme aus«. Die Diskussionen drehen sich um die gleichen Themen, und die Retrospektiven werden schal und nutzlos. Es stellt sich eine Routine ein, und die Retrospektiven werden zu einem weiteren Meeting, das man einfach hinter sich bringen möchte.

Antipattern-Lösung

Die offensichtliche Lösung besteht darin, jedes Mal, wenn eine Retrospektive geplant ist, die Teammitglieder zu fragen, ob sie etwas Neues zu besprechen haben, und wenn die Antwort nein lautet, wird die Retrospektive ausgesetzt.

Konsequenzen

Um zu verstehen, welche Folgen es hat, wenn auf eine Retrospektive verzichtet wird, sollte man sich zunächst die erwarteten Vorteile vor Augen führen: eine gemeinsame Geschichte über das, was geschehen ist oder geschieht, eine Möglichkeit, Dampf abzulassen, wenn etwas nicht gut läuft, und eine gemeinsame Ent-

scheidungsfindung über das weitere Vorgehen. Auch die persönlichen Faktoren sind bei Retrospektiven wichtig. Hier kommen oft alle zusammen, um zu lachen, Gefühle zu verschiedenen Themen auszutauschen und zu lernen, wie man sich gegenseitig oder sich selbst helfen kann. Richtig moderiert, können Retrospektiven dazu beitragen, als Team zu lernen und zu wachsen. Oft ist dies die einzige Gelegenheit für ein Team, über das Offensichtliche hinauszugehen und sich mit den tieferen Problemen zu befassen. Wie J. P. Morgan sagte, gibt es zwei Gründe für jede Entscheidung und jede Handlung: den offiziellen Grund und den wahren Grund.

Symptome

Man hört im Team: »Wir brauchen keine Retrospektiven mehr«, »Es gibt nichts zu besprechen« oder »Wir sind jetzt ein gutes Team – wir möchten keine Zeit mehr damit verschwenden, Probleme zu erfinden«.

Refactoring-Lösung

Wie der Untertitel des Buches von Diana Larsen und Esther Derby, *Making Good Teams Great* [Larsen & Derby 2006], andeutet, sind Retrospektiven für alle Teams nützlich, nicht nur für Teams, die in irgendeiner Weise dysfunktional sind. Wir können immer besser werden in dem, was wir tun: Wir können uns von gut zu großartig entwickeln.

Weltklasse-Skifahrer*innen versuchen immer noch, ihr Skifahren zu verbessern: Sie überprüfen ständig, was sie tun, und passen ihre Technik, ihren Schlaf oder ihre Ernährungsgewohnheiten entsprechend den Ergebnissen an. Auch wenn Sie in eine Zahnarztpraxis gehen, um Ihre Zähne reinigen zu lassen, oder in eine Werkstatt, um die Räder Ihres Autos auswuchten zu lassen, müssen Sie das regelmäßig wiederholen. Das sage ich manchmal auch zu einem Team, das sich in der Phase seiner Retrospektivenreise befindet, wo es glaubt, keine Retrospektiven mehr zu brauchen.

Und dann führe ich etwas Neues in ihre Retrospektiven ein. Ich probiere eine neue Art von Retrospektive aus, z. B. eine *positive Retrospektive* oder eine *Teamradar-Retrospektive* (siehe Abb. 7-1), oder ich mache eine *Zukunftsperspektive*, die aufzeigt, was sie befürchten oder hoffen, das passieren wird. Wie im Antipattem »*Negatives Team*« (Kap. 21) beschrieben, kann das Team nicht nur aus den positiven Aspekten eines Problems lernen, sondern auch aus den negativen – und welche guten Praktiken noch besser gemacht werden können.

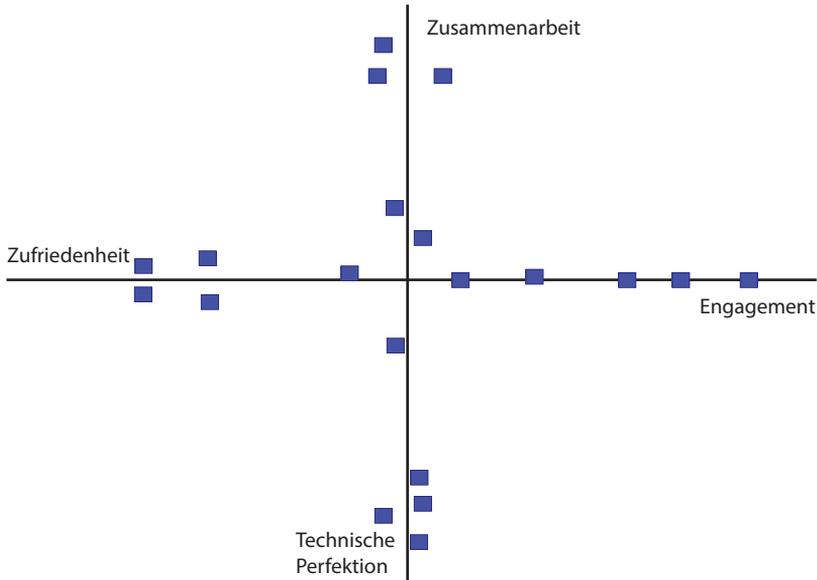


Abb. 7-1 Teamradar-Retrospektive mit vier Themen

Zukunftsperspektive

Eine *Zukunftsperspektive* ist wie eine Retrospektive, aber der Fokus liegt auf der Zukunft, nicht auf der Vergangenheit. Für eine Zukunftsperspektive erstelle ich an dem Board einen Zeitstrahl, der heute (am Tag der Zukunftsperspektive) beginnt und in die Zukunft reicht, vielleicht drei Monate, vielleicht ein Jahr oder vielleicht bis zum nächsten Release. Dann bitte ich die Teammitglieder, sich vorzustellen, sie befänden sich in der Zukunft, am Ende des Zeitstrahls. Wenn ich mich traue, bitte ich sie, die Augen zu schließen und zu versuchen, sich vorzustellen, wie die Welt »jetzt« aussieht. Dann öffnen sie die Augen und kleben Post-it-Zettel an das Board, auf denen sie alle positiven und negativen Ereignisse beschreiben, die vom Anfang bis zum Ende des Zeitstrahls stattgefunden haben.

Ein typischer Einwand gegen diese Übung ist, dass die Menschen nicht in die Zukunft sehen können. Als moderierende Person können Sie zugeben, dass sie die Zukunft nicht vorhersagen können, aber erklären, dass sie ihre Hoffnungen und Befürchtungen in Bezug auf das, was geschehen wird, mitteilen werden.

Wenn die Teilnehmenden ihre erwarteten positiven und negativen Ereignisse eingetragen haben, gehen Sie diese durch und besprechen, was zu diesen Ereignissen geführt hat – die Ursachen oder die Geschichten dahinter. Das Ergebnis einer Zukunftsperspektive kann sein, dass die Teammitglieder Richtlinien beschließen, die es ihnen ermöglichen, so zu arbeiten, wie es für sie am besten funktioniert, z.B. keine morgendlichen Treffen abhalten. Es könnte auch ein Hinweis an das Management sein, was es braucht, damit dieses Team erfolgreich ist – zum Beispiel sollte das Team zusammenbleiben und nicht von anderen Projekten im Unternehmen in seiner Arbeit behindert werden. Außerdem



könnten die Teammitglieder, wie bei einer normalen Retrospektive, Experimente identifizieren: Dinge, die sie ein paar Wochen lang tun können, und sich die Auswirkungen ansehen, wie z.B. ein Peer-Review aller Arbeiten oder Mob Programming³ einmal im Monat.

Für mich ist ein zusätzlicher Vorteil, dass ich viel über die Vergangenheit der Teammitglieder erfahre. In ihren Befürchtungen beschreiben sie, welche negativen Dinge sie zuvor erlebt haben könnten. In ihren Hoffnungen beschreiben sie, wie sie im Wesentlichen arbeiten möchten. Daher führe ich oft zu Beginn der Zusammenarbeit eine Zukunftsperspektive mit einem neuen Team durch.

- a. Wie beim Pair Programming, nur dass das ganze Team gemeinsam programmiert und dabei die von Woody Zuill aufgestellten Regeln befolgt (Anm. d. Übers.: siehe z.B. <https://woodyzuill.com/>).

Eine weitere nützliche Übung ist »Die Schiff-Retrospektive« (Abb. 7–2), die auch eine gute Möglichkeit ist, in einer Retrospektive etwas Neues auszuprobieren.

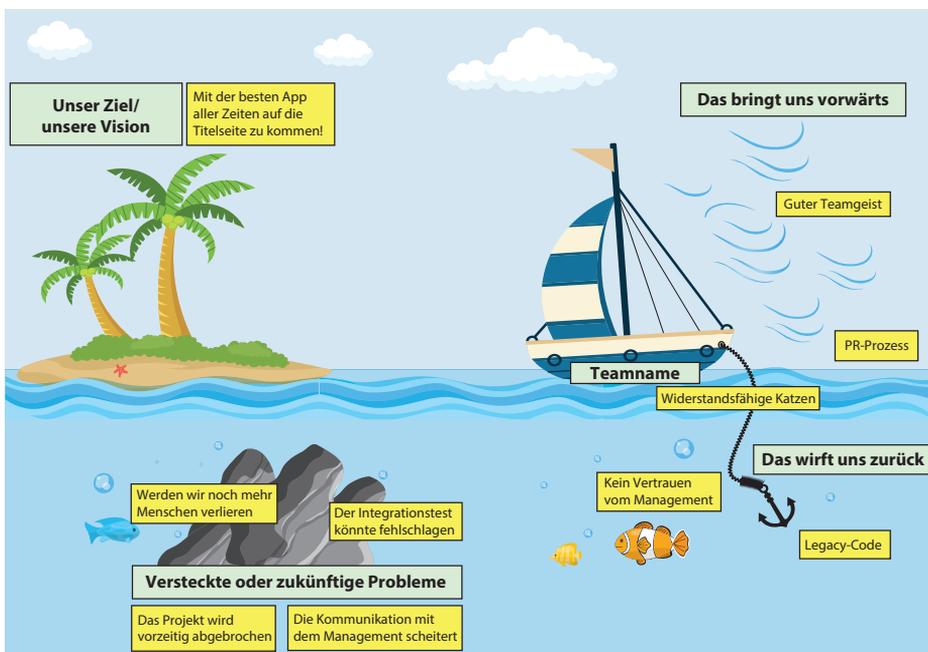


Abb. 7–2 Die Schiff-Retrospektive

Die Schiff-Retrospektive

Die *Schiff-Retrospektive* ist um das Bild eines Schiffes vor einer unbewohnten Insel, mit Wind in den Segeln, einem Anker und einigen unter Wasser versteckten Klippen herum aufgebaut. Das Team muss sich zunächst darauf einigen, was sich auf der Insel befindet, d. h., was das Traumszenario, ihre Vision, ist. Dann sollte dem Schiff ein Name gegeben werden, denn die Entscheidung über den Namen für das Schiff macht deutlich, wie Menschen im selben Team sehr unterschiedliche Ansichten über den bestimmenden Faktor dieses Teams haben können. Ich überspringe diesen Schritt manchmal aus Zeitgründen. Nun schreiben die Teammitglieder einzelne Post-it-Zettel für die positiven Dinge und platzieren sie auf dem Wind, der die Segel in Bewegung setzt, damit das Schiff die Insel erreicht. Der nächste Schritt ist der Anker, der für die Dinge steht, die uns von der Insel fernhalten und das Vorankommen erschweren. Der letzte Schritt sind die Klippen, d.h. die Dinge, bei denen wir uns nicht sicher sind, aber befürchten, dass sie passieren könnten. Dann ist die Phase »Daten sammeln« abgeschlossen, und das Team kann mit den Phasen »Erkenntnisse gewinnen« und »Entscheiden, was zu tun ist« beginnen. Die Schiff-Retrospektive wird unter anderem in dem Buch *Getting Value Out of Agile Retrospectives: A Toolbox of Retrospective Exercises* [Gonçalves & Linders 2014] beschrieben.

Eine andere Möglichkeit besteht darin, die Teammitglieder zu fragen, was das Worst-Case-Szenario ist oder wie sie arbeiten würden, wenn sie weniger Zeit, mehr Zeit, weniger Leute, mehr Leute usw. hätten. Versuchen Sie, ihnen ein anderes Weltbild zu vermitteln, und beobachten Sie, was ihnen dazu einfällt. Wenn sie nicht erkennen können, was sie ändern müssen, um sich zu verbessern, liegt das in der Regel daran, dass sie sich so sehr an den jetzigen Zustand gewöhnt haben, dass es ihnen schwerfällt, sich eine andere Situation für das Team vorzustellen.

Manche Teams haben gelernt, sich in der Retrospektive nur mit ihrer Kommunikation, ihrer Zusammenarbeit und ihren Prozessen zu befassen. Obwohl diese Überlegungen wichtig sind, kann es sich auch lohnen, im Teil »Daten sammeln« der Retrospektive harte Daten zu betrachten wie z.B. Burndown-Charts, Übersichten über Regressionstests, Peer-Review-Statistiken, Feedback von Personen, die die Software nutzen.

Die Wahl eines Themas für die Retrospektive kann auch eine Möglichkeit sein, sich auf ein bestimmtes Thema zu konzentrieren und es zu vertiefen. Das Thema kann ein bestimmtes Release, die Teststrategie, die Art und Weise, wie wir als Team und individuell lernen, die Architektur oder was auch immer für dieses Team interessant sein könnte, sein.

Manchmal können Sie sich erlauben, in der Retrospektive eine etwas aktivere Rolle einzunehmen. Wenn ich zum Beispiel Fragen und Kommentare höre wie: »Warum werden wir nicht einfach besser darin, Zeitpläne für Aufgaben zu schätzen? Dann brauchen wir nicht so viel zu reden, sondern melden uns einfach,

wenn die Arbeit fertig ist« oder: »Müssen wir besprechen, was gut läuft? Können wir uns nicht einfach auf die Dinge konzentrieren, die schlecht laufen?«, dann stelle ich manchmal gerne Fragen dazu, wie die Teammitglieder wirklich arbeiten wollen. Glauben sie an Agilität? Und an Wertschätzung? Und welche Ziele verfolgen sie mit diesem Team?

Sie können auch eines der Modelle zum Ermitteln der Wirksamkeit, wie z.B. »Agile Fluency«, verwenden, um dem Team bewusst zu machen, wie es in Bezug auf agile Methoden, Standards für die Codequalität oder Teamzufriedenheit arbeitet. Dann können Sie eine Diskussion darüber beginnen, wo es hinwill oder wo es dachte, auf welcher Stufe es sich befindet.

Agile Fluency

Agiles Arbeiten zu lernen ist wie das Erlernen einer neuen Sprache. Man lernt zwar die Sprache, aber wenn etwas Unerwartetes passiert, wechselt man wieder zurück zur gewohnten Sprache. Ich kann zum Beispiel Deutsch sprechen, aber wenn ich mir den Zeh an einem Tischbein anstoße, fluche ich auf Dänisch. Wenn ich eine Sprache nicht fließend spreche, wechsele ich zu einer Sprache, die ich fließend beherrsche, wenn ich überrascht, wütend oder ängstlich bin. Das Gleiche gilt für die agile Arbeitsweisen in Teams. Die Teammitglieder haben vielleicht einige Kurse besucht und gelernt, wie man mit Scrum arbeitet, aber wenn etwas Unerwartetes passiert, wechseln sie zum Wasserfallmodell, weil sie das kennen und verstehen. Das Ziel von Agile Fluency ist es, zu ermitteln und zu visualisieren, in welchem Maße ein Team mit agilen Methoden vertraut ist.

Diana Larsen und James Shore haben das »Agile Fluency Project« mitbegründet, mit dem Ziel, Coaches dabei zu unterstützen, Teams in die Lage zu versetzen, auf der Stufe von Agile Fluency zu arbeiten, die den Bedürfnissen ihres Unternehmens am besten entspricht.

Der Sinn dieser überarbeiteten Lösung besteht darin, Teammitglieder dabei zu unterstützen, Wege zu finden, um jeden Aspekt des Teams und seiner Arbeit zu verbessern, und sei es auch nur in kleinen Schritten.

Online-Aspekt

Wenn die Retrospektive online durchgeführt wird, gelten die gleichen Punkte. Der einzige Unterschied besteht darin, dass es schwieriger ist, die Agenda einer Online-Retrospektive zu ändern, sodass Sie dieses Antipattern vielleicht nicht ändern können, wenn Sie darauf stoßen, aber Sie können es für die nächste Retrospektive einplanen.

Persönliche Anekdote

Wie bereits erwähnt, habe ich diesen speziellen Teil des Lebenszyklus von Retrospektiven bei zahlreichen Teams erlebt. Bei einem bestimmten Team war es auch für mich schwer zu erkennen, wie es sich verbessern könnte. Es schien alles genau richtig zu machen und die Teammitglieder waren dabei sehr nett zueinander.

Ich beschloss, eine Retrospektive rund um das Wort »mutig« vorzubereiten. Die Phase »*Den Rahmen schaffen*« beinhaltete eine Frage dazu, wie mutig jedes Teammitglied im letzten Monat gewesen war. Ich habe nur nach einem Beispiel gefragt, das sowohl bei der Arbeit als auch im Privatleben vorkommen konnte. (Natürlich sind wir auch die Aktionspunkte der letzten Retrospektive durchgegangen und haben besprochen, ob die Experimente erfolgreich waren, was sie gelernt hatten und was sie weiterhin tun werden.)

In der Phase »*Daten sammeln*« habe ich die Teammitglieder gebeten, sich drei Dinge zu überlegen, die sie tun würden, wenn sie eines Morgens aufwachen und zehnmal mutiger als sonst wären. Mindestens zwei dieser Dinge sollten einen Bezug zu ihrer Arbeit haben. Man könnte einwenden, dass alle drei Dinge mit ihrer Arbeit zusammenhängen sollten, aber ich habe mich aus zwei Gründen dafür entschieden, einen möglichen privaten Rahmen einzubeziehen.

Erstens kann diese Frage zu einigen lustigen Fakten übereinander führen, zu Dingen, die sie schon immer einmal ausprobieren wollten, sich aber nie getraut haben. Spaßige Fakten können das Team zum Lachen bringen und den Zusammenhalt der Gruppe stärken. Manchmal finden die Teammitglieder sogar heraus, dass sie sich gegenseitig helfen können, ein »mutiges Ziel« zu erreichen. Als zum Beispiel jemand sagte, er wolle fliegen lernen, lud ihn eine Teamkollegin ein, bei ihrem nächsten Flug ihr Co-Pilot zu sein.

Zweitens: Indem Sie in der Frage Raum für Gedanken jenseits der Arbeit lassen, ermutigen Sie die Teilnehmenden vielleicht dazu, über einen wichtigen persönlichen Wunsch zu sprechen, über den sie bei der Arbeit normalerweise nicht reden würden, der sie aber dennoch in Gedanken beschäftigt. Wenn sie ihre wichtigen Träume oder Ziele mitteilen, anstatt zu versuchen, ihre Gedanken zu unterdrücken, sind sie weniger abgelenkt und können sich besser auf ihre Arbeit konzentrieren.

Das Ergebnis dieser Übung waren so unterschiedliche Antworten wie »direkt mit den künftigen Nutzenden des Systems im nächsten Einkaufszentrum sprechen«, »Mob Programmierung ausprobieren«, »um Hilfe bitten, wenn ich auf etwas stoße, das ich nicht verstehe« und »mich für die Backend-Entwicklung umschulen zu lassen«. In der nächsten Phase, »*Erkenntnisse gewinnen*«, ging es um die Geschichten hinter ihren Antworten. Wenn die Personen wollten, konnten sie die Erfahrungen mitteilen, die zu ihrer Liste der Dinge führten, die sie tun würden,

wenn sie zehnmal mutiger wären. Hatten sie es bereits versucht und waren gescheitert? Was ist passiert, als sie es versucht haben? Was hat sie dazu bewogen, es zu versuchen, und was haben sie sich davon versprochen?

In der Phase »*Entscheiden, was zu tun ist*« wählte jede Person eine Idee aus, auf die sie sich konzentrieren wollte oder jemand anderem dabei helfen wollte, sich darauf zu konzentrieren. Und dann unterstützte ich sie dabei, herauszufinden, was ein weniger beängstigender erster Schritt sein könnte. Die Retrospektive brachte einige neue Themen an die Oberfläche, da die Teammitglieder gezwungen waren, auf eine andere Art zu denken. Einigen fiel es leichter als anderen, aber alle Anwesenden machten sich Gedanken über ihr normales Verhalten und darüber, ob etwas auf persönlicher Ebene verbessert werden könnte.

Ein weiterer Aspekt dieses Antipatterns ist, dass manche Teams von vornherein denken, dass Retrospektiven aus einem bestimmten Grund keinen Wert für sie hätten. Kürzlich hat mir jemand gesagt, dass »Agile Coaches und Scrum Master*innen nervös werden, weil überall in ihrem Umfeld DevOps-Methoden verwendet werden. Wenn Scrum Master*innen und Agile Coaches über DevOps Bescheid wüssten, würde sich ihre gesamte Sichtweise auf Retrospektiven ändern. Bei DevOps-Methoden geht es darum, Ergebnisse zu messen. In diesem Fall sollte sich die Retrospektive auf diese Ergebnisse konzentrieren, und viele DevOps-Ergebnisse sind technischer Natur: Fehlerrate in der Produktion, Feature-Zykluszeit, mittlere Zeit bis zur Wiederherstellung des Service. Man kann keine Diskussionen über diese Dinge führen, ohne etwas über DevOps zu wissen.«

Ich mag Aussagen wie diese sehr, weil sie mir die Möglichkeit geben, zu erklären, warum ich Retrospektiven für wertvoll halte. Zumindest wird die Person, die diese Aussage gemacht hat, Retrospektiven nicht nur um der Retrospektiven willen einführen. Ich habe auf diesen Kommentar geantwortet, dass erstens, auch wenn die Leute DevOps-Methoden einsetzen, sie immer noch Menschen sind und die gleichen Probleme haben wie alle anderen auch.

Zweitens, auch wenn ich weiß, dass die meisten Leute bei einer Retrospektive beim Sammeln von Daten an Daten zu Vorfällen und Gefühlen denken, sehe ich oft auch Burndown-Charts, Regressionstestergebnisse, Wiederherstellungszeiten und Antwortzeiten. Manchmal leite ich Retrospektiven an, bei denen das Team beschlossen hat, sich nur mit technischen Themen zu befassen, was sehr gut funktionieren kann. Oft sprechen wir aber auch über zwischenmenschliche Themen, weil wir uns Fragen stellen wie: »Wie kam es zu dieser Antwortzeit?« oder: »Warum haben wir dieses Testergebnis erhalten?«

Drittens weiß ich, dass ich von meinem technischen Hintergrund profitiere, und wenn Leute in technische Diskussionen eintauchen, kann ich sie manchmal abkürzen, weil ich verstehe, was sie sagen, und weiß, wann sie anfangen, sich zu wiederholen. Aber viele moderierende Personen haben diesen Hintergrund nicht

und sind trotzdem perfekt in der Lage, Retrospektiven zu moderieren. Wir als Moderator*innen sind dazu angehalten, unsichtbar zu sein und die Diskussionen zu unterstützen. Mit dem richtigen Verständnis für Körpersprache und Teamdynamik ist es dennoch möglich, ein Meeting zu moderieren, bei dem es um etwas geht, von dem man thematisch nichts weiß.

Viertens, und das ist spezifisch für DevOps-Methoden, moderiere ich derzeit Retrospektiven für zwei Teams, ein Betriebsteam und ein Entwicklungsteam, die gerade zu einem DevOps-Team zusammengelegt werden. Sie haben spezifische Probleme im Zusammenhang mit der Zusammenführung ihrer separaten Teams zu einem einzigen. Die Traumvorstellung ist natürlich, dass sie als ein Team arbeiten und dass die Entwickler*innen nicht nur Sympathie für den Betrieb haben, sondern auch Empathie, weil sie lernen, was im Betrieb gemacht wird – und vice versa. Das bringt einige Herausforderungen mit sich, wenn es darum geht, das Wissen und die Fähigkeiten der anderen zu respektieren und eine Lernumgebung zu schaffen, in der sie voneinander lernen, damit sie Daten sammeln und auf diese Daten reagieren können. Es gibt viele Spannungen zwischen diesen beiden Teams, und die Retrospektiven helfen ihnen, sich gegenseitig zu respektieren, wenn auch langsam.

Diese Leseprobe haben Sie beim

 edv-buchversand.de heruntergeladen.

Das Buch können Sie online in unserem
Shop bestellen.

[Hier zum Shop](#)