7 Teamwork

> Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
>
> Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
>
> – Manifest für Agile Softwareentwicklung[1]

Cross-funktionale, selbstorganisierte Teams sind die grundlegende Ressource jeder agilen Organisation. Doch wer sollte Teil eines agilen Teams sein? Woher wissen sie, woran sie arbeiten sollen? Wie ist es für sie möglich, gut miteinander zu arbeiten?

Dieses Kapitel enthält die erforderlichen Praktiken, um ein großartiges agiles Team zu bilden.

- Abschnitt »Komplettes Team« auf Seite 94 erstellt ein cross-funktionales Team mit allen benötigten Fähigkeiten.
- Abschnitt »Teamraum« auf Seite 113 errichtet einen Raum, ob physisch oder virtuell, in dem das Team effektiv zusammenarbeiten kann.
- Abschnitt »Sicherheit« auf Seite 134 schafft eine Umgebung, die es Teammitgliedern erlaubt, ihre Erfahrungen und Einsichten zu teilen.
- Abschnitt »Zweck« auf Seite 145 hilft den Teammitgliedern zu verstehen, wie ihre Arbeit die übergeordneten Pläne des Unternehmens unterstützt.
- Abschnitt »Kontext« auf Seite 158 erläutert die Stakeholder und die zugesagten Ressourcen des Teams.
- Abschnitt »Ausrichtung« auf Seite 165 legt Standards fest, die es den Teammitgliedern ermöglichen, effektiv zusammenzuarbeiten.
- Abschnitt »Energiegeladene Arbeit« auf Seite 176 ermutigt zu einer Arbeitsweise, die Ihr Team auf Dauer aufrechterhalten kann.

1. *https://agilemanifesto.org/iso/de/principles.html*

Die Ursprünge von Teamwork

Der Kerngedanke cross-funktionaler, selbstorganisierter Teams war von Anfang an Teil der Agilität. Extreme Programming nennt es »Komplettes Team« (Whole Team), ein Begriff, den ich für dieses Buch beibehalten habe. Scrum nennt es das »Scrum-Team«. Der Begriff »Teamraum« (Team Room) ist genauso alt; XP nennt es »Sit Together« und das Original-Scrum-Buch [Schwaber & Beedle 2002] spricht von der Bedeutung einer team-orientierten Arbeitsumgebung.

Auch das Thema *Sicherheit* findet sich seit Jahren in der agilen Konversation. [Beck 2004] beschreibt sie in seiner Diskussion zum kompletten Team. Joshua Kerievsky hat »Sicherheit zur Voraussetzung machen« zu einem Leitprinzip moderner Agilität gemacht. Die Diskussion zu Sicherheit in diesem Buch wird von der Gastautorin Gitte Klitgaard geführt, die über umfangreiche Erfahrung in der Unterstützung von Teams und Organisationen beim Schaffen von psychologischer Sicherheit verfügt.

Die Praktiken zu *Zweck*, *Kontext* und *Ausrichtung* basieren auf dem exzellenten Buch *Liftoff: Start and Sustain Successful Agile Teams* von Diana Larsen und Ainsley Nies [Larsen & Nies 2016]. Zusammen sind sie ein Beispiel für »Chartering«, das ich das erste Mal in einem agilen Kontext in »Industrial XP« (IXP) von Joshua Kerievsky gesehen habe [Kerievsky 2005]. Chartering wurde vom unnachahmlichen III[2], der auch einen großen Einfluss auf die Arbeit von Nies und Larsen hatte, in die IXP-Methode integriert.

Energiegeladene Arbeit (Energized Work) ist eine weitere seit Langem bestehende Idee der Agilität. Der Begriff stammt aus dem XP: Die erste Version von XP benannte es die »40-Stunden-Woche«, doch es wurde in der zweiten Auflage auf »Energiegeladene Arbeit« geändert, was weniger US-zentriert klingt.

7.1 Komplettes Team

Zielgruppe

Coaches

Wir verfügen über alle erforderlichen Fähigkeiten, um exzellente Ergebnisse zu liefern.

Moderne Softwareentwicklung braucht eine Menge Fähigkeiten. Nicht nur Programmierkenntnisse, sondern auch soziale Kompetenz, künstlerische Fähigkeiten, technische Fähigkeiten. Wenn diese Kompetenzen nicht Teil des Teams sind, leidet deren Leistungsfähigkeit. Anstatt sich auf eine neue Funktion zu konzentrieren und diese fertigzustellen, müssen Teammitglieder mehrere Aufgaben unter einen Hut bringen, während sie E-Mails versenden, auf Antworten warten und sich mit Missverständnissen auseinandersetzen.

Um diese Art von Verzögerungen und Fehlern zu vermeiden, sind agile Teams cross-funktionale, *komplette Teams*. Sie setzen sich aus

Verwandtes Thema

Zweck (S. 145)

2. III war sein vollständiger Name, er wird »drei« ausgesprochen.

Personen mit unterschiedlichen Fähigkeiten und Erfahrungen zusammen, die gemeinsam über alle Fähigkeiten verfügen, die das Team benötigt, um seinen Zweck zu erfüllen. Im Großen und Ganzen lassen sich diese Fähigkeiten gruppieren in Fähigkeiten im Umgang mit Kunden, Fähigkeiten in der Entwicklung und Coaching-Fähigkeiten.

Beachten Sie, dass agile Teams *Fähigkeiten* brauchen, keine *Rollen*. Manchmal stellen erfahrene Programmiererinnen mit einer langen Betriebszugehörigkeit die besten Produktmanagerinnen dar. Manchmal verfügen Projektmanager über hervorragende Testfähigkeiten.

> Agile Teams brauchen *Fähigkeiten*, keine Rollen.

Und nicht nur das: Agile Teams lernen und wachsen mit der Zeit. Alle arbeiten daran, die eigenen Fähigkeiten auszubauen, insbesondere kundenbezogene Fähigkeiten.

Wenn ich in diesem Buch von einem »Produktmanager«, einer »Entwicklerin« oder einem anderen Jobtitel spreche, referenziere ich auf eine Person in dem Team, die *über diese Fähigkeiten verfügt*. Agile Teams arbeiten am besten, wenn Personen ihren Beitrag abhängig von ihren Fähigkeiten leisten und nicht aufgrund ihrer Position im Organisationsdiagramm.

Cargo-Kult

Das löchrige Team (hole vs. whole)

»Okay, ihr seid jetzt agil«, sagt Ihr Manager und verschwindet in einer nach Golf duftenden Rauchwolke.

Ihre Vierergruppe schaut sich nervös an. Sie sind ein Team, entwickeln das Frontend und sind sich nicht sicher, was Sie tun sollten. Sie haben Gerüchte zu einer neuen Initiative gehört und Ihr Team wird daran beteiligt sein … irgendwie?

Am nächsten Tag stürmt Claudia herein. »Hi, ich bin euer Scrum Master«, sagt sie. »Entschuldigt, dass ich gestern nicht da war. Ich habe noch vier andere Teams und habe gerade erfahren, dass ich mit euch zusammenarbeiten werde. Ramona wird euer Product Owner sein. Aber sie schafft es heute nicht. Ich habe ein Treffen in der übernächsten Woche vereinbart.«

Claudia macht Sie mit dem Produkt vertraut, an dem Sie arbeiten werden. Ihr Team baut die Benutzungsoberfläche und diverse andere Teams bauen die Microservices im Backend. Das Testen übernimmt, wie immer, die Qualitätssicherungsabteilung und wenn Sie bereit sind auszuliefern, erstellen Sie ein Ticket beim Betrieb.

→

Cargo-Kult

Der Betrieb ist dann für die Überwachung und Verfügbarkeit der Applikation verantwortlich. »Hier ist der Oberflächen-Mock-up, den die Designabteilung erstellt hat«, sagt Claudia und Ramona legt Storys im Tool für die Fehlerverfolgung mit allen Anforderungen an. »Ich schaue jeden Tag für das Standup-Meeting vorbei. Sagt mir einfach, was ihr gemacht habt, und ich aktualisiere die Storys im Nachverfolgungswerkzeug.«

Claudia stürmt hinaus, ihre Stimme verhallt im Flur. »Lasst mich wissen, wenn ihr etwas braucht!« Sie schauen sich an, zucken mit den Schultern und öffnen das Tracking-Werkzeug. Die Storys sind nicht so richtig klar, also versenden Sie ein paar E-Mails. In der Zwischenzeit beginnen Sie mit dem Build.

Die Monate vergehen. Es läuft nicht gut. Sie treffen sich alle paar Wochen mit Ramona und am Ende müssen Sie viel von dem, was sie verlangt, überarbeiten. Die Anforderungen in den Storys sind nicht immer eindeutig, daher müssen Sie per E-Mail nachfragen und in der Zwischenzeit nach bestem Wissen arbeiten.

Selbst wenn Sie denken, etwas sei erledigt, findet die Qualitätssicherungsabteilung Probleme bei Dingen, bei denen Sie sich sicher waren, richtig gehandelt zu haben. Doch die Storys ließen Raum für Interpretation. Sie bitten Ramona, mehr Details zu erfassen, doch es sind nie genug. Die Systeme im Backend funktionieren nie ganz so, wie Sie es erwarten würden, und es dauert Ewigkeiten, bis der Betrieb die Entwicklungsumgebung aktualisiert hat.

Doch dann ist es so weit und Sie liefern aus. Sie wissen nicht, wie gut es ankommen wird, aber das ist zu diesem Zeitpunkt auch nicht wichtig. Sie sind einfach nur bereit, sich mit etwas anderem zu beschäftigen. Von jetzt an ist es immerhin das Problem der Betriebsabteilung.

Fähigkeit, im Sinne des Kunden zu handeln

Verwandtes Thema

Echte Kundenbeteiligung (S. 245)

Leute mit der Fähigkeit, Kunden-, Nutzer- und Geschäftsinteressen zu repräsentieren, werden die *Kunden des Teams vor Ort* oder einfach nur »Kunden« genannt. Sie sind dafür verantwortlich, herauszufinden, was gebaut werden soll. Abhängig davon, was für eine Art von Software Sie entwickeln, können Ihre Kunden vor Ort Ihre echten Kunden sein, oder es sind vielleicht Leute, die diese echten Kunden *vertreten*.

Um wirklich erfolgreich zu sein, muss Ihre Software einen Mehrwert für Ihre Kunden, Ihre Benutzer und Ihre Organisation bringen.

Einer der schwierigsten Aspekte beim Aufbau eines agilen Teams ist es, Menschen mit der Fähigkeit zu finden, den Kunden zu vertreten oder seine Sicht einzunehmen. Vernachlässigen Sie diese Fähigkeiten nicht. Sie sind essenziell, um den Wert Ihres zu liefernden Produkts zu steigern. Ein großartiges Team kann auch ohne

Kunden vor Ort technisch hervorragende Arbeit leisten. Doch um wirklich erfolgreich zu sein, muss Ihre Software einen Mehrwert für Ihre Kunden, Ihre Benutzer und Ihre Organisation bringen. Dies erfordert Kundenkompetenz, die in verschiedene Kategorien unterteilt wird:

Produktmanagement (aka Product Ownership oder Produktverantwortung)

Verwandte Themen

Zweck (S. 145),
Kontext (S. 158),
Adaptives Planen (S. 196),
Visuelles Planen (S. 217),
Stakeholder-Demos (S. 351),
Vertrauen der Stakeholder (S. 342)

Agile Teams konzentrieren sich darauf, Wert zu schaffen, doch woher wissen sie, was wertvoll ist? Das ist der Punkt, an dem das Produktmanagement ins Spiel kommt. Teammitglieder mit Fähigkeiten im Produktmanagement arbeiten mit Stakeholdern zusammen, um herauszufinden, *woran* das Team arbeiten sollte, *warum* das wichtig ist und *wen* sie zufriedenstellen sollten. Sie führen Demos durch, holen Feedback ein und werben innerhalb der Organisation für die Arbeit des Teams. In den meisten Teams handelt es sich dabei um eine Vollzeitaufgabe.

Produktmanager müssen außerdem über die organisationale Autorität verfügen, Kompromissentscheidungen darüber zu treffen, welche Funktionen in das Produkt aufgenommen werden und welche draußen bleiben. Es braucht politisches Geschick, um die Interessen der verschiedenen Stakeholder in Einklang zu bringen, sie mit den Zielen des Teams zu konsolidieren und Wünsche, die nicht erfüllt werden können, entschlossen abzulehnen.

Leute mit solch einem Kaliber an Fähigkeiten und Einfluss bekommen viele Anfragen während ihrer Arbeitszeit. Es kann schwierig sein, ihre Aufmerksamkeit zu erhalten. Bleiben Sie hartnäckig. Das Produktmanagement ist eine der wichtigsten Fähigkeiten im Team. Ist die Software nicht ausreichend wertschöpfend, um die Zeit von jemandem mit guten Fähigkeiten im Produktmanagement zu rechtfertigen, einer Person, die den Unterschied zwischen Erfolg und Misserfolg ausmachen könnte, dann hat die Software vielleicht von Beginn an nicht den Wert, entwickelt zu werden.

Viele Unternehmen weisen ihren Produktmanagern zu viele Teams zu, sodass wenig Zeit für die einzelnen Teams bleibt. Das kann bei langsam arbeitenden Teams mit vorhersehbaren Aufgaben funktionieren, führt aber in der Regel dazu, dass die Teams Zeit damit verschwenden, die falschen Dinge zu entwickeln. [Rooney 2006] erlebte dieses Problem mit bedauerlichen Ergebnissen:

> Wir waren uns nicht sicher, was unsere Prioritäten waren. Uns war nicht klar, woran wir als Nächstes arbeiten sollten. Wir zogen Storys von der allgemeinen Liste, aber der Kunde [Produktmanager] gab uns nur wenige Hinweise, woran wir arbeiten sollten. So ging es ein paar Monate.

Dann fanden wir heraus, dass der Hauptsponsor sauer war, so richtig sauer. Wir hatten nicht an dem gearbeitet, was wir seiner Meinung nach hätten tun sollen.

Begehen Sie nicht den Fehler, das Produktmanagement zu vernachlässigen. Vergessen Sie aber nicht: Teams benötigen Leute mit *Produktmanagementfähigkeiten*, nicht Personen mit dem *Jobtitel* Produktmanager. Erfahrene Entwicklerinnen können mit entsprechender Ausbildung exzellente Produktmanagerinnen sein, insbesondere wenn sie schon länger an dem Produkt oder im Unternehmen arbeiten. Zum Beispiel hat bei Toyota der Chefingenieur eines Fahrzeugs die vollumfängliche Verantwortung für alles: vom Konzept bis zum wirtschaftlichen Ergebnis.

Begehen Sie nicht den Fehler, das Produktmanagement zu vernachlässigen.

Domänenkompetenz (aka Fachkenntnisse)

Die meisten Softwaresysteme werden in einer bestimmten Branche eingesetzt, wie beispielsweise dem Finanzwesen, mit jeweils eigenen speziellen Geschäftsregeln. Um in dieser Branche erfolgreich zu sein, muss die vom Team entwickelte Software diese Regeln getreu und genau implementieren. Diese Regeln werden *Domänenregeln* genannt. Die Kenntnis über diese Regeln ist das *Domänenwissen*.

Das Team braucht Personen mit Fachkenntnissen, die die Verantwortung dafür übernehmen, diese Details herauszufinden, Widersprüche aufzulösen und die Antworten parat zu haben. Das sind Menschen mit umfassender Erfahrung. Ein agiles Team, mit dem ich gearbeitet habe, hat eine Software für die chemische Analyse entwickelt und hatte jemanden im Team mit einem Masterabschluss in Analytischer Chemie. Ein anderes Team baute eine Software für das Sicherheitsmanagement von Bank zu Bank und verfügte daher über zwei Finanzexpertinnen. Ein drittes Team entwickelte eine Software für Lebensversicherungen und sein Fachexperte war Versicherungsmathematiker.

Selbst wenn Ihre Software keine komplizierten Domänenregeln hat, brauchen Sie trotzdem Personen, die herausfinden können, was Ihre Software im Einzelnen tun soll. In einigen Teams ist das vielleicht der Produktmanager, ein User Experience Designer oder eine Business-Analystin.

Im Unterschied zum Produktmanagement, bei dem Sie eine Menge Zeit mit Stakeholdern verbringen, erfordern Fachkenntnisse, viel Zeit mit dem Team zu verbringen. Die meiste Zeit vergeht dabei, die Details der bevorstehenden Arbeit herauszufinden, Beispiele komplizierter Regeln zu erstellen und Fragen zur Fachlichkeit zu beantworten.

Verwandte Themen

Inkrementelle Anforderungen (S. 252),
Kundenbeispiele (S. 326),
Ubiquitous Language (S. 467)

User Experience Design (aka Interaktionsdesign)

Die Benutzungsoberfläche (User Interface, kurz UI) Ihrer Software ist das Gesicht Ihres Produkts. Für viele Benutzerinnen ist die UI das Produkt. Sie bewerten das Produkt allein anhand ihrer Wahrnehmung der UI.

Leute mit UX[3]-Fähigkeiten gestalten die Benutzeroberfläche. Diese Fähigkeiten konzentrieren sich darauf, Benutzerinnen und ihre Bedürfnisse zu verstehen und nachzuvollziehen, wie sie mit dem Produkt interagieren. Die Aufgaben umfassen Interviews mit Nutzern, das Erstellen von Personas, mit Nutzern gemeinsam Prototypen zu bewerten, die Benutzung der aktuellen Software zu beobachten und diese Informationen in spezifische Entwürfe und Bilder zu übertragen.

Die schnelllebige, iterative, Feedback-orientierte Natur agiler Entwicklung führt zu einer anderen Umgebung, als es UX-Designerinnen gewohnt sind. Das UX-Design wird nicht in einer vorgelagerten Phase des Benutzungsverhaltens festgelegt, sondern iterativ, parallel zur iterativen Verfeinerung der Software selbst. Agile Teams liefern Software alle ein bis zwei Wochen, das bietet Designerinnen die Möglichkeit, mithilfe echter Software Muster im Benutzerverhalten zu beobachten und dieses Feedback für die Planung des Teams zu nutzen.

Entwicklungsfähigkeiten

Ein großes Ziel erfordert eine solide Ausführung. Wenn es bei der Fähigkeit, die Sicht des Kunden einzunehmen, darum geht, herauszufinden, was das Team tun soll, dann geht es bei den Entwicklungsfähigkeiten darum, sich darüber klar zu werden, wie das Team etwas umsetzt. Personen mit Entwicklungsfähigkeiten tragen die Verantwortung dafür, den effektivsten Weg zu finden, die Software auszuliefern.

Anmerkung

Einige Leute bezeichnen Entwicklungsfähigkeiten als »technische Fähigkeiten«, doch das klingt für mich abwertend. Es ist nicht so, als seien analytische Chemikerinnen und Versicherungsmathematiker nicht technisch orientiert. In Anbetracht eines besseren Begriffs bezeichne ich Menschen, die dabei helfen, die Software zu bauen, zu testen und auszuliefern, als Personen mit »Entwicklungsfähigkeiten«.

3. UX = User Experience

Programmierung, Entwurf und Architektur

Programmierkenntnisse werden von jedem Team benötigt, das Software entwickelt. In einem *Delivering*-Team arbeitet jede Person, die programmiert, auch am Entwurf und an der Architektur mit, und vice versa. Das Team verwendet testgetriebene Entwicklung für die Kombination von Architektur, Entwurf, Test und Programmierung für eine für sich stehende, fortlaufende Aktivität.

Menschen mit Fachkenntnissen in den Bereichen Entwurf und Architektur werden weiter benötigt. Sie unterstützen, indem sie das Team bei den Entwurfs- und Architekturbestrebungen anleiten und anderen Teammitgliedern Wege zur Vereinfachung komplexer Entwürfe aufzeigen. Sie agieren als respektierte Kolleginnen, die eher anleiten als vorschreiben.

Programmierkenntnisse werden auch für die Planung und das Vermeiden von Fehlern benötigt. Ebenso, um sicherzustellen, dass die Software in der Produktion einfach zu implementieren und zu betreiben ist.

Verwandte Themen

Testgetriebene Entwicklung (S. 501),
Einfacher Entwurf (S. 563),
Inkrementeller Entwurf (S. 550),
Reflektierender Entwurf (S. 574),
Evolutionäre Systemarchitektur (S. 614),
Das Planning Game (S. 234),
Keine Fehler (S. 626),
Für den Betrieb bauen (S. 589)

Testen

In einem *Delivering*-Team unterstützen Menschen mit Testfähigkeiten das Team von Beginn an bei der Lieferung hochqualitativer Ergebnisse. Sie setzen ihre Fähigkeit zum kritischen Denken dabei ein, um Kundinnen zu helfen, alle Möglichkeiten in Betracht zu ziehen, wenn sie sich das Produkt vorstellen. Sie helfen außerdem als technische Ermittler beim Erkennen blinder Flecken und beim Sammeln von Informationen über nicht funktionale Eigenschaften, wie Performance und Sicherheit.

Im Gegensatz zu den meisten anderen Teams geht es beim Testen in einem *Delivering*-Team nicht darum, ausgiebig nach Fehlern zu suchen. Es wird stattdessen davon ausgegangen, dass das restliche Team von sich aus nahezu fehlerfreien Code produziert. Sollte doch mal ein Fehler durchrutschen, passt das Team seine Gewohnheiten an, um diese Art von Fehlern in der Zukunft zu vermeiden.

Verwandte Themen

Inkrementelle Anforderungen (S. 252),
Kundenbeispiele (S. 326),
Aufdecken blinder Flecken (S. 636),
Keine Fehler (S. 626)

Betrieb

Delivering-Teams benötigen Teammitglieder mit operativen Fähigkeiten. Sie unterstützen das Team bei der Bereitstellung, Überwachung und dem sicheren Produktionsbetrieb der Software. In kleineren Organisationen sind sie vielleicht auch verantwortlich für die Bereitstellung und Verwaltung der Hardware. In größeren Organisationen koordinieren sie sich mit einer zentralen Betriebsgruppe.

Verwandte Themen

Continuous Deployment (S. 607), Für den Betrieb bauen (S. 589), Vorfallanalyse (S. 644)

Zu den operativen Fähigkeiten gehört es auch, dem Team zu helfen, den Überblick über ihr Produktionssystem zu behalten und Produktionsanforderungen einzuplanen. Das betrifft die Themen Sicherheit, Performance, Skalierung, Überwachung und Verwaltung. Außerdem ist das Erstellen einer gerechten Rotation für Bereitschaften (falls erforderlich) zu nennen und die Unterstützung, wenn das Team Fehler im Produktionssystem analysieren oder verhindern möchte.

Coaching-Fähigkeiten

Teams, die erst seit Kurzem agile Methoden verwenden, haben viel zu lernen: Sie müssen lernen, wie sie agile Praktiken einsetzen und wie sie effektiv als selbstorganisierte Teams zusammenarbeiten können.

Auch ihre Organisationen haben viel darüber zu lernen, wie sie ihre Teams unterstützen können. Der Großteil der Unterstützung erfolgt über die in Kapitel 4 beschriebenen Investitionen, doch es sind immer auch zusätzliche Anpassungen erforderlich. Und obwohl die notwendigen Investitionen im besten Fall im Voraus getätigt wurden, müssen Teams häufig für diese eintreten, nachdem sie bereits mit der Arbeit begonnen haben.

Personen mit Coaching-Fähigkeiten unterstützen Teams darin, zu lernen, wie sie effektive agile Teams werden können. Sie schulen Praktiken, leiten die Selbstorganisation und die Teamentwicklung an und zeigen dem Team, wie es mit Führungspersonen und anderen Stakeholdern zusammenarbeitet, damit es die benötigten Investitionen erhält.

> Die Aufgabe eines Coaches ist es, dem Team dabei zu helfen, eigenständig zu routinemäßigen Handlungen zu kommen.

Teams, die neu mit agilen Methoden arbeiten, haben typischerweise eine, manchmal zwei Personen, die explizit als Coach des Teams benannt sind. Die Aufgabe dieser Coaches ist es, dem Team dabei zu helfen, eigenständig zu routinemäßigen Handlungen zu kommen, sodass die Teammitglieder in der Lage sind, die benötigten Fähigkeiten ohne die Unterstützung eines Coaches einzusetzen. Das muss nicht bedeuten, dass die Coaches das Team verlassen, aber sie könnten es verlassen. Verbleiben sie im Team, werden sie allmählich zu festen Teammitgliedern.

Anmerkung

Selbst nachdem ein Team eigenständig routinemäßige Handlungen ausführen kann, ist es hilfreich, wenn ein erfahrener Coach hin und wieder hinzukommt. Zum Beispiel einmal im Jahr, um dem Team Hilfe beim Ausprobieren neuer Dinge anzubieten und es an vergessene Praktiken zu erinnern.

Je nachdem, welche *Fluency*-Zonen Teams anstreben, benötigen sie in bis zu vier Kategorien Coaching. Dies kann mehr als einen Coach erfordern:

- Teamentwicklung, Selbstorganisation und Moderation (alle Teams)
- *Focusing*-Zone – Planung und Praktiken der Teamzusammenarbeit
- *Delivering*-Zone – Entwicklungspraktiken
- *Optimizing*-Zone – Praktiken der Geschäftsentwicklung

Ein Teil der Arbeit des Coaches besteht darin, dem Team beizubringen, selbstständig zu agieren. Teammitglieder müssen in der Lage sein, ihre eigenen Diskussionen zu moderieren, die eigenen Teamdynamiken und -praktiken zu verbessern, herauszufinden, welche Investitionen sie effektiver machen, und mit Stakeholdern zusammenzuarbeiten, um diese Investitionen zu erhalten. Wie bei allen Fähigkeiten im Team müssen nicht alle dazu in der Lage sein, aber je mehr Teammitglieder dazu fähig sind, desto belastbarer wird das Team sein.

Manche Coaches tappen in die Falle, diese Dinge für das Team zu tun, anstatt die Teammitglieder darin zu schulen, es selbst zu tun. Stellen Sie sicher, dass dies in Ihrem Team nicht der Fall ist.

Verwandte Themen

Retrospektiven (S. 398),
Teamdynamik (S. 408),
Beseitigung von Hindernissen (S. 425)

Praxis-Coaches

Mein bevorzugter Typ eines agilen Coaches ist der *Praxis-Coach*: eine Person mit echtem Fachwissen in der Anwendung agiler Praktiken, die mit gutem Beispiel vorangehen kann. Ihr Schwerpunkt liegt weiterhin darauf, dem Team und der Organisation beim Lernen zu helfen, anstatt selbst etwas zu liefern. Aber sie hat die benötigten Fähigkeiten und den Hintergrund, eher *vorzuleben* als zu *beschreiben*, und das beinhaltet oft, Teammitglieder bei ihrer Arbeit zu unterstützen. Erfahrene Praxis-Coaches können mit ein oder zwei Teams zeitgleich arbeiten oder »Mitspiel-Coaches« für mehrere Teams anleiten.

Mitspiel-Coaches

Eine Variante des Praxis-Coaches ist der *Mitspiel-Coach*. Er hat Erfahrung mit agilen Praktiken und verfügt über einige Coaching-Fähigkeiten. Er konzentriert sich mehr darauf, für das Team Dinge zu liefern, als es beim Lernen zu unterstützen. Intern ausgebildete Coaches und die meisten erfahrenen agilen Entwicklerinnen fallen häufig in diese Kategorie. Sie haben vielleicht den Titel des »Technical Lead« oder »Senior Engineer«.

Mitspiel-Coaches können sehr effektiv dabei sein, wenn es darum geht, Teams beim Anwenden agiler Praktiken zu unterstützen, sogar bis zu dem Punkt, an dem sie diese routiniert anwenden können. Doch sie sind in der Regel weniger erfolgreich, wenn es darum geht, Teams dabei zu helfen, dies eigenständig zu erreichen. Außerdem kann es für sie schwierig sein, zu verstehen, wie und wann sie den organisatorischen Wandel beeinflussen können. Sie sollten einem einzelnen Team zugeordnet sein.

Moderator-Coaches

Eine der am meistverbreiteten Arten von Coaches ist der *Moderator-Coach*, häufig auch Scrum Master[4] genannt. Er führt von der Seitenlinie aus, indem er Besprechungen moderiert und organisatorische Hindernisse auflöst. Er vermittelt meistens die Praktiken der *Focusing*-Zone und hilft Teams dabei, selbstständig zu werden. Er ist bei Teams nützlich, die mit vielen Hindernissen zu tun haben, denn er kann sich dafür einsetzen, dass die erforderlichen Investitionen getätigt werden. Ein Mitspiel-Coach und Moderator-Coach können mit ihren komplementären Stärken und Schwächen ein gutes Team darstellen.

Erfahrene Moderator-Coaches können mit einem oder zwei Teams gleichzeitig arbeiten. Ein Nachteil bei Moderator-Coaches ist, dass sie nicht viel zur täglichen Entwicklung beitragen. Das kann dazu führen, dass Unternehmen sie als zu wenig ausgelastet betrachten und sie daraufhin zu vielen Teams zuweisen. Doch dann sind sie oft nicht anwesend, um die Herausforderungen des Teams zu erkennen und darauf zu reagieren. Coaches, die sich in dieser Situation befinden, enden häufig als glorifizierte Organisatoren von Meetings, was ihren Fähigkeiten nicht gerecht wird.

Generalisierte Spezialistinnen

Agile Teams funktionieren am besten, wenn sie aus *generalisierten Spezialistinnen*, auch *T-Shaped*-Personen genannt, zusammengesetzt sind. Eine generalisierte Spe-

4. Die Bezeichnung »Scrum Master« hat ihren Ursprung in der Scrum-Methode. Der Name ist irreführend; er soll jemanden bezeichnen, der Scrum meisterhaft verstanden hat, nicht jemand, der Autorität oder Kontrolle über das Team besitzt.

zialistin verfügt in mehreren Bereichen über fundierte Fachkenntnisse. Das ist die vertikale Linie des T. Sie verfügt aber auch über die Fähigkeit, breit gefächert ihren Beitrag zu anderen vom Team benötigten Fähigkeiten zu leisten. Damit ist die horizontale Linie des T gemeint. (Einige Leute verwenden den Begriff »*M-Shaped*-Personen«, damit möchten sie verdeutlichen, dass generalisierte Spezialistinnen in mehreren Bereichen Fachexpertise entwickeln können.)

Agile Teams benötigen generalisierte Spezialistinnen, um Engpässe zu vermeiden. Nicht agile Organisationen unternehmen komplexe »Ressourcenplanungen«, um sicherzustellen, dass jedes Team mit den richtigen Spezialistinnen zur richtigen Zeit besetzt ist. Diese Planungen gehen nie ganz auf, weil sich die Arbeit in der Softwareentwicklung nicht so vorhersagen lässt, wie die Ressourcenplanung es erfordert. So kommt es dazu, dass Teams auf Personen warten, die sich verspätet haben, oder sich damit beeilen, Arbeit für Leute zu finden, falls diese bereits fertig sind, bevor das Teams bereit ist. Dadurch kommt es häufig zu einer bruchstückhaften Zuteilung, bei der das Management versucht, alles unter einen Hut zu bringen. Das führt zu einer Menge Verschwendung.

In einer agilen Organisation stellen Teams die Ressource dar, nicht Individuen. Das macht die Ressourcenplanung viel einfacher. Alles oder nichts. Entweder bearbeitet das Team ein Feature oder nicht. Entweder ist eine Person dem Team zugeordnet oder sie ist es nicht.

Doch es kann weiterhin zu Engpässen in Ihrem Team kommen. Stellen Sie sich ein Team mit zwei Frontend-Entwicklern und zwei Backend-Entwicklerinnen vor. Manchmal wird es mehr Arbeit im Frontend geben und die Backend-Entwicklerinnen drehen daraufhin Däumchen. (Oder sie arbeiten vor, was letzten Endes zu verschwenderischer Nacharbeit führt, wie ich im Abschnitt »Reduzieren Sie angefangene Arbeit« auf Seite 201 beschreibe.) Zu anderen Zeiten wird es genau umgekehrt sein.

Teams, die über generalisierte Spezialistinnen verfügen, vermeiden diese Art von Engpässen. Wenn es eine Menge Arbeit im Frontend zu tun gibt, übernehmen die Spezialisten im Frontend zwar die Führung, doch die Backend-Spezialistinnen helfen aus. Wenn es viel Arbeit im Backend zu tun gibt, übernehmen die Backend-Spezialistinnen die Führung. Und das beschränkt sich nicht nur auf die Programmierung. Unabhängig davon, welchen Engpässen das Team auch begegnet, sollten Teammitglieder bereit sein, einzuspringen und auszuhelfen.

Beim ersten Zusammenstellen des Teams muss es sich nicht bei allen Personen um generalisierte Spezialistinnen handeln. Es geht dabei eher um eine Frage der Einstellung als um Fähigkeiten. Jede Spezialistin eines Fachgebiets kann sich dazu entscheiden, einen Beitrag zu Bereichen zu leisten, die an das eigene Fachgebiet grenzen. Achten Sie bei der Auswahl der Teammitglieder darauf, dass sie bereit sind, auch bei Aufgaben außerhalb ihres Fachgebiets zu helfen.

Besetzung des Teams

Die exakten Rollen und Jobtitel in Ihrem Team sind nicht wichtig, solange das Team über alle benötigten Fähigkeiten verfügt. Die Jobtitel in Ihrem Team werden mehr mit der Tradition der Organisation zu tun haben als mit irgendetwas anderem.

Anmerkung

Für neue agile Teams ist es hilfreich, die Rollen der Produktmanagerin und des Coaches explizit zu benennen. Für erfahrenere agile Teams können zugewiesene Rollen ein Hindernis darstellen. Doch Leute, für die Agilität neu ist, wissen es zu schätzen, an wen sie sich bei Fragen wenden können.

Product Owner in Teilzeit werden auf Dauer nicht mithalten können.

Es ist möglich, dass Sie Schwierigkeiten dabei bekommen, jemanden mit Fähigkeiten im Produktmanagement oder mit Fachwissen in Ihrem Team aufzunehmen. In vielen Unternehmen werden Personen mit diesen Fähigkeiten beauftragt, sich in Teilzeit als »Product Owner« (oder Produktverantwortlicher) mit dem Team abzustimmen. Verstehen Sie das als ein Signal, dass das Team seine eigenen Fähigkeiten in den Bereichen Produktmanagement und domänenspezifischer Expertise entwickeln sollte. Auch wenn der externe Product Owner beim Start helfen kann, wird er auf Dauer nicht mithalten können. Die besten agilen Teams verfügen selbst über tiefgreifende Fähigkeiten, die Sicht ihrer Kunden einzunehmen. Bas Vodde hat es gut ausgedrückt:[5]

> Die meisten Teams, mit denen ich zusammenarbeite, befinden sich auf einer Reise, die sie von »Programmierern« zu »Produktentwicklern« führt. Das bedeutet, das Team (das ganze Team!) verfügt über ein genaues Verständnis seiner Kunden und Kundendomäne, anstatt von jemanden abhängig zu sein, der für sie alles klärt und das Ergebnis dieser Klärung an sie weitergibt. Ja, ich schätze es sehr, Leute mit fundierten Kundenkenntnissen in meinem Team zu haben. Doch das Ganze verfolgt hauptsächlich das Ziel, das gesamte Team zu verbessern.

Auch bei einem hochqualifizierten Team werden einige Entscheidungen von Leuten außerhalb des Teams getroffen werden. Das kann der Fall sein, wenn Ihr Team an einem größeren Produkt mitarbeitet. Entscheidungen, die die Architektur betreffen, werden dann vielleicht außerhalb Ihres Teams getroffen. Es ist in Ordnung, wenn solche Entscheidungen im Hintergrund erfolgen. Doch wenn Sie

5. Aus einer persönlichen Kommunikation.

immer wieder auf Entscheidungen oder auf Zuarbeit von Leuten außerhalb des Teams warten, haben Sie kein »komplettes Team«. Dieses Wissen und diese Abhängigkeiten sollten in das Team geholt werden oder der Teil des Produkts, der die Abhängigkeiten benötigt, sollte aus dem Team herausgenommen werden. Kapitel 6 geht beim Thema »Koordination zwischen Teams« in die Tiefe.

Vollständig zugewiesene Teammitglieder

Jedes permanente Teammitglied sollte einzig diesem Team zugeordnet sein. *Anteilige Zuordnung*, also einzelne Personen mehreren Teams zuzuordnen, führt zu furchtbaren Ergebnissen. Anteilig zugeordnete Mitarbeitende gehen mit ihrem Team keine Bindung ein, sind häufig nicht anwesend, um Unterhaltungen mitzubekommen und Fragen zu beantworten, und sie müssen zwischen Aufgaben hin- und herwechseln. Das führt wiederum zu erheblichen versteckten Einbußen. »Die minimale Einbuße beträgt 15 Prozent … Anteilig zugeordnete Wissensarbeiter wirken vielleicht beschäftigt, doch ein großer Teil ihrer Arbeit ist bloß unnötiger Wirbel« [DeMarco 2002, Kap. 3].

Es wird bestimmt einige Fähigkeiten geben, die Ihr Team nur gelegentlich benötigt. Dafür können Sie Leute, die über diese Fähigkeiten verfügen, bitten, temporär Teil Ihres Teams zu werden. Nehmen wir das folgende Beispiel an: Ihr Team baut ein komplexes, serverseitiges Produkt mit einer minimalistischen Benutzungsoberfläche. Für die wenigen Wochen, an denen Ihr Team daran arbeitet, können Sie einen User Experience Designer um Unterstützung bitten.

Selbst wenn jemand nur temporär Teil des Teams ist, sollten Sie sicherstellen, dass die Person dem Team während dieser Zeit vollständig zugewiesen ist. Es ist besser, wenn sich jemand eine Woche lang voll und ganz Ihrem Team widmet und dann für eine weitere Woche einem anderen Team, als dass dieselbe Person zwei Wochen lang in beiden Teams arbeitet.

Stabile Teams

Es kann mehrere Monate dauern, bis Teams gelernt haben, wie sie effektiv zusammenarbeiten können. In einigen Organisationen werden Teams für jedes neue Projekt gebildet und wieder aufgelöst. Es ist jedoch weniger verschwenderisch, Teams zusammenzuhalten. Selbst wenn der Zweck des Teams erreicht oder das Produkt fertiggestellt ist, sollten Sie das Team nicht auflösen. Geben Sie ihm stattdessen eine neue Aufgabe.

Dasselbe gilt auch für die Änderung von Teamzusammensetzungen. Wenn zu einem existierenden Team eine Person hinzukommt, fügt sich diese in die bestehende Teamkultur und die Teamnormen ein. Wenn Sie stattdessen dem Team viele Leute hinzufügen, fängt das Team praktisch wieder bei null an. Meine Faustregel lautet, pro Monat nur eine Person hinzuzufügen oder zu entfernen.

Verwandtes Thema

Teamdynamik (S. 408)

Teamgröße

Verwandte Themen

Pair Programming (S. 448),
Mob Programming (S. 461)

Die Anleitungen in diesem Buch sind für Teams von drei bis 20 Personen geeignet. Für neue Teams sind vier bis acht Leute ein guter Startpunkt. Sobald mehr als 12 Personen im Team sind, werden Sie Störungen in der Kommunikation wahrnehmen. Seien Sie also bei der Bildung großer Teams vorsichtig. Sind Ihre Teams hingegen sehr klein, überlegen Sie, diese zusammenzulegen. So reduzieren Sie Ihren Overhead und werden weniger anfällig für Fluktuation sein.

Für die meisten Teams wird die Programmierung der erste Engpass sein. Berücksichtigen Sie daher bei Ihren Überlegungen zu Teamgrößen, wie viel Anteil der Arbeitszeit in die Programmierung fließt. Der Einfachheit halber spreche ich im Folgenden von einer Programmiererin und bezeichne damit eine Person, die sich ausschließlich der Programmierung widmet. Das soll aber nicht bedeuten, dass Ihr Team aus strikt definierten Rollen bestehen muss.

- Teams, die nicht Pair oder Mob Programming einsetzen, sollten über drei bis fünf Programmiererinnen verfügen. Je größer sie werden, desto schwieriger wird die Koordination.
- Teams, die Pair Programming einsetzen, sollten über vier bis zehn Programmiererinnen verfügen. Sechs ist das Optimum. Teams, denen die Praktiken der *Delivering*-Zone neu sind, sollten es vermeiden, über sechs bis sieben Programmiererinnen hinaus zu wachsen, bis sie über mehr Erfahrung verfügen.
- Wenn Teams für ihre Arbeit Mob Programming einsetzen, sollten sie über drei bis fünf Programmiererinnen verfügen. Sie können Mob Programming mit deutlich größeren Gruppen verwenden, es ist eine großartige Technik, aber irgendwann ist ein Punkt erreicht, an dem der Nutzen abnimmt.

Typischerweise besetzen Sie den Rest des Teams proportional zum Umfang der Programmieraufgaben. Das Verhältnis sollte dem Umfang der zu erledigenden Arbeit entsprechen, damit es nicht zu Engpässen kommt. Generalisierte Spezialistinnen geben etwas Freiraum, um Schwankungen der Arbeitslast zu bewältigen. Planen Sie im Allgemeinen wie folgt:

- *Kunde*: Ein bis zwei »Vor-Ort-Kunden« (Vertreter des Kunden) im Team für jeweils drei Programmiererinnen. Ein Viertel bis eine Hälfte ihrer Arbeitszeit wird auf das Produktmanagement entfallen. Der Rest fällt in die Kategorien Fachwissen sowie UX- und UI-Design, abhängig davon, um welche Art von Software es sich handelt.

- *Test*: Eine Testerin auf jeweils zwei bis vier Programmiererinnen, wenn das Team nicht wie im Schlaf über die Fähigkeiten der *Delivering*-Praktiken verfügt. Führt das Team diese Handlungen routiniert aus, dann reicht eine Testerin auf vier bis acht Programmiererinnen.
- *Betrieb*: Keine bis zwei Personen aus dem Betrieb, abhängig von der Produktionsumgebung.
- *Coaching*: Ein oder zwei Coaches, die eventuell ihre Zeit auf ein weiteres Team aufteilen.

Noch einmal: Es geht hier nicht darum, strikte Rollen zuzuweisen. Sie könnten zum Beispiel ein Team aus sechs Programmiererinnen einschließlich eines Mitspiel-Coaches haben, die gemeinsam die Hälfte ihrer Zeit mit Programmierung verbringen, ein Sechstel ihrer Zeit damit, mit Kunden zu arbeiten, ein Sechstel mit Tests und ein Sechstel mit Betriebsthemen.

Warum so viele »Vor-Ort-Kunden«?

Zwei Kunden vor Ort für jeweils drei Programmiererinnen klingt noch viel, finden Sie nicht? Ich habe mit einem deutlich kleineren Verhältnis begonnen. Gleichzeitig habe ich häufig beobachtet, wie Kundenvertreterinnen Schwierigkeiten haben, mit den Programmiererinnen Schritt zu halten. Letztendlich kam ich auf das Verhältnis von zwei zu drei, nachdem ich verschiedene Verhältnisse bei mehreren erfolgreichen Teams ausprobiert hatte.

Diese Teams verfügten alle über Fluency in den *Delivering*-Praktiken und hatten das Produktmanagement im Team integriert. Die meisten von ihnen hatten mit komplexen Problembereichen zu tun. Wenn Ihre Software leicht zu verstehen ist, Sie keine *Delivering* Fluency anstreben oder Sie kein Produktmanagement im Team haben, werden Sie vermutlich auch weniger Kundenvertreterinnen benötigen.

Doch bedenken Sie, dass die Arbeit mit Kunden umfangreich ist. Sie müssen herausfinden, was den höchsten Wert liefert, die geeigneten Prioritäten für die Arbeit setzen, die Details kennen, nach denen die Entwicklerinnen fragen werden, und Zeit dafür finden, um gemeinsam mit Kunden Reviews und Demos durchzuführen. Sie sollten all das tun und gleichzeitig immer dem Entwicklungsteam einen Schritt voraus sein. Wobei das Entwicklungsteam, insbesondere wenn es sich um ein *Delivering*-Team handelt, direkt hinter ihnen wie ein Güterzug durch die Arbeit rauscht. Es ist eine große Aufgabe. Unterschätzen Sie sie nicht.

Ein Team von Individuen auf Augenhöhe

Keine Person im Team ist einer anderen Person vorgesetzt. Das bedeutet nicht, dass jede Entscheidung auch zur Diskussion steht. Vielmehr wird das letzte Wort

von denen gesprochen, die über die jeweilige Fachexpertise verfügen. Zum Beispiel können sich Entwicklerinnen nicht über die Produktpriorisierungen der Kunden hinwegsetzen. Gleichzeitig können diese nicht Entscheidungen der Entwicklerinnen zu technischen Notwendigkeiten überstimmen. Ebenso werden Sie Teammitglieder mit viel Erfahrung haben, die bei wichtigen Entscheidungen die Führung übernehmen. Doch sie werden keine Vorgesetztenstruktur im Team finden. Selbst Leute mit schicken Bezeichnungen wie »Produktmanagerin« managen nicht das *Team*.

Dies ist ein wichtiger Bestandteil eines selbstorganisierten Teams. Ein selbstorganisiertes Team entscheidet selbst, wer für welche Aufgabe die Führung übernimmt. Dabei handelt es sich nicht um eine schwierige Entscheidung. Bei Teams, die sich untereinander gut kennen, ergibt sich die Entscheidung, wer die Führung übernimmt, gewöhnlich automatisch. Es wird die Person sein, die am meisten über die Aufgabe weiß, die das größte Interesse und Lernbereitschaft daran gezeigt hat, die als Nächstes an der Reihe ist oder jede sonstige Variante.

Es ist schwer zu vermitteln, wie viel Spaß es bereitet, in einem agilen High-Performance-Team zu arbeiten. Brian Marick, einer der Autoren des Agilen Manifests, pflegte zu sagen, dass »Freude« ein weiterer agiler Wert sei.[6] Er hat recht, großartige agile Teams haben das gewisse Etwas. Sie sind optimistisch, enthusiastisch und arbeiten wirklich gerne gemeinsam. Gibt es zum Beispiel eine Aufgabe, die niemand erledigen möchte, verläuft die Unterhaltung dazu, wer sie übernimmt, spielerisch und lustig. Und die Entscheidung wird schnell getroffen. Effektive agile Teams tun sich leicht damit, Entscheidungen zu treffen.

Es braucht Zeit – und Arbeit –, um so effektiv zu werden. Die Übung zu »Teamdynamiken« zeigt, wie es geht.

Verwandtes Thema

Teamdynamik (S. 408)

Schlüsselidee

Selbstorganisierte Teams

Agile Teams treffen selbstständig Entscheidungen, woran sie arbeiten, wer daran arbeitet und wie die Arbeit erledigt wird. Es ist einer der Grundsätze von Agilität: *Die Menschen, die die Arbeit erledigen, sind diejenigen, die am ehesten dazu qualifiziert sind, zu entscheiden, wie sie getan wird.*

→

6. Marick identifizierte vier Werte, die er bei frühen agilen Teams entdeckte: Fertigkeit, (Selbst-) Disziplin, Leichtigkeit und Freude [Marick 2007a].

Schlüsselidee

Das ist der Grund, weshalb dieses Buch so viele Praktiken zu Planungen, zur Zusammenarbeit und zur Arbeit mit Stakeholdern enthält. Das sind alles Themen, die beim Team liegen, nicht beim Management. Es wird von ihnen erwartet, gemeinsam zu arbeiten, die Verantwortung zu übernehmen und ihr Ziel zu erreichen.

Das bedeutet nicht, dass das Management auf der Ebene von Teams nichts zu tun hat. Tatsächlich bietet das Delegieren von Details an das Team für die Manager die Möglichkeit, sich auf andere Dinge zu konzentrieren – die für sie mehr Wirkung haben. Ihre Aufgabe besteht darin, ihren Teams zum Erfolg zu verhelfen, indem sie das größere System, in dem sich die Teams bewegen, managen. Mehr Details gibt es im Abschnitt »Management« auf Seite 382.

Rückblick auf das »Löchrige Team«

Werfen Sie einen zweiten Blick auf die Geschichte im Abschnitt »Das löchrige Team (hole vs. whole)« auf Seite 95. Was ist schiefgelaufen?

- Das Teammanagement überraschte das Team und ließ es im Stich, statt ihm zum Erfolg zu verhelfen.
- Die Coachin, Claudia, hat dem Team nicht dabei geholfen zu lernen.
- Die Produktmanagerin, Ramona, nahm sich keine Zeit für das Team.
- Das Team hatte niemanden, der den Kunden vertreten könnte.
- Das Team war nicht in der Lage, selbstständig zu liefern. Es musste sich mit der außenstehenden Qualitätssicherungsabteilung, dem Betrieb und anderen Backend-Teams abstimmen.

So hätte es ablaufen sollen:

»Okay, heute geht es mit der Agilität los«, sagt Ihr Manager. Sie hatten alle vor Wochen dem Vorhaben zugestimmt, ein agiles Vorgehen auszuprobieren. Daher ist niemand überrascht. »Wie wir besprochen haben, bilden wir ein neues Team, das an diesem Produkt arbeitet. Warum stellt ihr euch alle nicht noch einmal vor?«

Die Vorstellung verläuft reihum. Es gibt drei Frontend-Programmiererinnen (eine mit Full-Stack-Erfahrung), einen Backend-Programmierer, eine Testerin und einen UX Designer. Sie haben auch schon Ihre Coachin, Claudia, kennengelernt. Sie stellt wiederum die Produktmanagerin, Ramona, vor.

Claudia tritt vor. »Ich weiß, dass ihr alle gespannt seid, wie ein agiles Vorgehen funktioniert, daher starten wir direkt. Wir beginnen mit einer Aktivität, die ›Chartering‹ genannt wird. Ramona hat mit unseren wichtigsten Stakeholdern zusammengearbeitet, um zu verstehen, was wir bauen, warum wir es bauen und für wen wir es bauen. Wir treffen uns in ein paar Minuten mit ihnen und werden

uns dann auch Zeit nehmen, um herauszufinden, wie wir am besten zusammenarbeiten können.«

Über die nächsten Tage überlegen Sie sich, wie Sie die Arbeit angehen wollen, anschließend beginnen Sie die Kerntechnologien zusammenzufügen. Ramona ist nicht Teil Ihres Teams, aber Michael, Ihr UX Designer, arbeitet eng mit ihr zusammen, um die Pläne des Teams zu konkretisieren.

Die Wochen vergehen. Ramona schaut regelmäßig rein und Michael ist mittlerweile in der Lage, sie zu vertreten, sollte sie nicht verfügbar sein.

Mit Frontend, Backend, Test und Betrieb gemeinsam in einem Team sind Sie in der Lage, schnell voranzukommen. Ihre erste Stakeholder-Demo findet nach einem Monat statt und die Resonanz ist ermutigend. Es ist ein gutes Team und Sie sind gespannt, was als Nächstes passiert.

Fragen

Was passiert, wenn nicht ausreichend Personal vorhanden ist, um jedes Team mit den benötigten Fähigkeiten auszustatten, oder ein Team eine bestimmte Fähigkeit nicht die ganze Zeit benötigt?

Prüfen Sie als Erstes, ob Ihr Unternehmen überdurchschnittlich viele Programmierinnen einstellt, im Vergleich zu den anderen Fähigkeiten, die ein Entwicklungsteam braucht. Das ist ein bekannter Fehler. Wenn das der Fall ist, sollten Sie Ihre Priorisierung bei Personaleinstellungen überprüfen.

Wenn Sie verschiedene Teams haben, die am selben Produkt arbeiten, erwägen Sie eine vertikale Skalierung, um die Anstrengungen zu bündeln – wie im Abschnitt »Vertikale Skalierung« auf Seite 77 beschrieben.

Wenn diese Optionen nicht funktionieren, kann Ihr Unternehmen ein »Enabling Team[7]« gründen, das für die Bereitstellung von Leitlinien, Standards und Schulungen für andere Teams verantwortlich ist (siehe hierzu Abschnitt »Horizontale Skalierung« auf Seite 82). Das könnte zum Beispiel ein zentrales UX-Team sein, das einen Styleguide erstellt und Leuten beibringt, wie sie diesen Styleguide für ihre Software verwenden können. Das versetzt Teams in die Lage, ihre eigenen Probleme zu lösen, ohne über allumfassendes Fachwissen zu verfügen.

Wenn mehr Fachwissen gebraucht wird, können Sie darum bitten, dass jemand mit diesen Fähigkeiten Ihrem Team vorübergehend zugeteilt wird. Ist diese Person im Team, kann sie gleichzeitig andere Teammitglieder trainieren. Versuchen Sie die Arbeit, die solche Fähigkeiten erfordert, aufzuschieben, bis jemand mit diesen Fähigkeiten zur Verfügung steht. Auf diese Weise haben Sie keine halbfertige Arbeit, was zu Verschwendung führt, wie im Abschnitt »Reduzieren Sie angefangene Arbeit« auf Seite 201 beschrieben wird.

7. Anm. d. Übers.: Ein Enabling Team hilft einem anderen Team dabei, Hindernisse zu überwinden, und erkennt in diesem Zuge auch fehlende Fähigkeiten, siehe *https://teamtopologies.com/key-concepts* (zuletzt besucht August 2022).

Sind unerfahrene Teammitglieder mit allen anderen gleichgestellt?

Die Teammitglieder sind nicht unbedingt gleichberechtigt, alle haben verschiedene Fähigkeiten und Erfahrungen, aber sie sind gleichgestellt. Es ist klug, wenn unerfahrene Teammitglieder sich für Beratung und Förderung an erfahrenere wenden. Genauso klug ist es, wenn erfahrene Teammitglieder jeden mit Respekt behandeln, kollegiale Beziehungen aufbauen und Neulingen helfen, sich weiterzuentwickeln, indem sie sich zurückhalten. Sie bieten ihnen dadurch die Gelegenheit, Führung zu übernehmen.

Wie können Teammitglieder spezielle Fähigkeiten entwickeln, ohne sich in einem Team zu befinden, das sich mit diesem Spezialwissen beschäftigt?

Viele agile Organisationen bilden »Communities of Practice« rund um funktionale Spezialgebiete. Diese werden vielleicht von einer Managerin, einem zentralen Supportteam oder einfach von interessierten Freiwilligen geleitet. Meistens finden in ihnen eine Reihe von Veranstaltungen statt, in denen Trainings, Kontaktpflege und andere Möglichkeiten zur Erweiterung der Fähigkeiten angeboten werden.

Voraussetzungen

Um ein *komplettes Team* aufzubauen, braucht es die Zustimmung und Unterstützung des Managements und die Zustimmung der Teammitglieder, gemeinsam als agiles Team zu arbeiten. Lesen Sie in Kapitel 5 mehr dazu, wie Sie so eine Zustimmung erhalten.

Indikatoren

Wenn Sie ein *komplettes Team* haben:

- ist Ihr Team in der Lage, Probleme zu lösen und seinen Zweck zu erfüllen, ohne dabei auf Leute außerhalb des Teams zu warten;
- arbeiten Personen im Team außerhalb ihrer speziellen Fähigkeiten, um zu verhindern, dass Engpässe das Team ausbremsen;
- trifft Ihr Team Entscheidungen reibungslos und wirkungsvoll;
- wechseln Teammitglieder, je nach Aufgabe, nahtlos ihre Führungsrolle.

Alternativen und Experimente

Die Theorie, die hinter dieser Praktik steht, ist eindeutig. Es sollen Verzögerungen und Kommunikationsprobleme vermieden werden:

1. Finden Sie alle Personen, die Sie für das Erreichen Ihrer Ziele brauchen.
2. Setzen Sie sie im selben Team ein.
3. Lassen Sie sie gemeinsam auf diese Ziele hinarbeiten.

Das ist ein Kerngedanke der Agilität und es gibt nicht wirklich Alternativen, die der agilen Philosophie entsprechen. Doch es besteht die Möglichkeit, an vielen Details Anpassungen vorzunehmen. Sobald Sie ein paar Monate Erfahrung in den Praktiken gesammelt haben, sollten Sie einige Experimente durchführen.

Wie könnte Ihr Team zum Beispiel damit experimentieren, wie es Entscheidungen trifft? Funktioniert es besser, wenn jemand die Diskussionen explizit moderiert? Oder sollten Sie die Diskussionen ganz natürlich ablaufen lassen? Sollten bestimmte Entscheidungen auch bestimmten Personen zugewiesen werden? Oder sollte die Führungsverantwortung eher rotieren?

Auf diese Fragen gibt es keine eindeutigen Antworten. Probieren Sie es aus. Versuchen Sie es, schauen Sie, ob es funktioniert, und probieren Sie etwas Neues aus. Hören Sie niemals damit auf, zu experimentieren. Das ist der Weg, die Kunst der Entwicklung zu meistern.

Weiterführende Literatur

The Wisdom of Teams: Creating the High Performance Organization [Katzenback & Smith 2015] ist ein Klassiker zum Thema High-Performance-Teams.

Agile Conversations [Squirrel & Fredrick 2020] ist eine exzellente Wissensquelle für Coaches, die ihren Teams und Organisationen dabei helfen, eine agile Kultur zu etablieren.

7.2 Teamraum

Wir arbeiten schnell und effektiv zusammen.

Zielgruppe
Komplettes Team, Coaches

Cargo-Kult

Der Rest der Geschichte

Sie entwickeln gerade eine Story für Ihr Team und benötigen eine Klarstellung zu einer der Anforderungen. Sie schicken eine E-Mail an Ihre Fachexpertin Lynn, anschließend machen Sie für ein paar Dehnübungen eine Pause und holen sich einen Kaffee. Als Sie wiederkommen, hat Lynn immer noch nicht geantwortet, daher schauen Sie sich ein paar Softwareentwicklungs-Blogs an, die Sie schon länger lesen wollten.

→

Cargo-Kult

Eine halbe Stunde später klingelt es in Ihrem Posteingang. Lynn hat geantwortet.

Hoppla – es sieht so aus, als hätte Lynn Ihre Nachricht missverstanden und die falsche Frage beantwortet. Sie senden eine weitere Anfrage, aber eigentlich können Sie nicht länger warten. Sie beantworten sich die Frage so gut es geht selbst und machen sich wieder an die Arbeit.

Einen Tag und einige E-Mails später haben Sie die korrekte Antwort mit Lynn herausgefunden. Es war nicht genau das, was sie angenommen hatten, aber Sie waren ziemlich nah dran. Sie gehen zurück an die Stelle und korrigieren den Quellcode. Während Sie daran sitzen, bemerken Sie, dass es einen Sonderfall gibt, der noch nicht berücksichtigt wurde.

Sie könnten Lynn nach der Antwort fragen, aber es handelt sich dabei um einen ungewöhnlichen Fall, der vermutlich niemals in der Realität auftritt. Außerdem ist Lynn ziemlich beschäftigt und Sie hatten versprochen, die Story gestern fertig zu haben. (Tatsächlich waren Sie gestern fertig, bis auf diese lästigen kleinen Details.) Sie gehen von der wahrscheinlichsten Antwort aus, ohne jemals zu merken, dass Ihre Antwort falsch war.

Wenn Menschen nicht direkt miteinander kommunizieren können, schmälert das die Effektivität ihrer Kommunikation, wie im Abschnitt »Persönliche Gespräche« auf Seite 115 beschrieben. Es kommt zu Missverständnissen und Verzögerungen schleichen sich ein. Die Personen fangen an zu raten, um sich die Mühe zu ersparen, auf eine Antwort zu warten. Fehler treten auf und es entwickelt sich langsam eine Wir-gegen-sie-Haltung.

Um dieses Problem zu bekämpfen, versuchen viele Teams, den Bedarf an direkter Kommunikation zu verringern. Das ist eine vernünftige Reaktion. Wenn Fragen zu Verzögerungen und Fehlern führen, reduzieren Sie die Notwendigkeit, Fragen zu stellen! Sie verbringen mehr Zeit damit, im Vorfeld Anforderungen zu ermitteln und jeden Bedarf zu dokumentieren. Später, so die Theorie, müssen die Programmiererinnen nicht mehr mit einem Experten sprechen. Sie können einfach alle Antworten im Dokument nachschlagen.

Das Geschriebene wird zu leicht missverstanden.

Das klingt nachvollziehbar, aber es funktioniert in der Praxis nicht gut. Es ist zu schwierig, jede Frage vorherzusehen, und das Geschriebene kann leicht missverstanden werden. Es zieht außerdem den Entwicklungsprozess in die Länge: Bevor die Arbeit beginnt, müssen die Personen Zeit damit verbringen, Dokumente zu schreiben, zu übergeben und zu lesen.

Stattdessen verwenden agile Teams einen *Teamraum*, um direkt zu kommunizieren. Das ist ein Ort, egal ob physisch oder virtuell, an dem das Team gemeinsam miteinander arbeiten kann. Statt dass jemand mit Domänenexperten spricht und ein Dokument schreibt, das die Programmiererinnen später lesen können, integrieren agile Teams Domänenexperten und andere Vor-Ort-Kunden ins Team. Wenn die Programmiererinnen verstehen müssen, was zu tun ist, sprechen sie direkt mit dem Kunden vor Ort.

Verwandtes Thema

Komplettes Team (S. 94)

Gemeinsam in einem Raum zu arbeiten, hat enorme Vorteile. In einer Feldstudie mit sechs sich am selben Ort befindenden Teams fand [Teasley et al. 2002] heraus, dass sich die Produktivität durch gemeinsames Arbeiten verdoppelte und die Markteinführungszeit sich auf fast ein Drittel im Vergleich zum Ausgangswert verkürzte.

Verwandtes Thema

Inkrementelle Anforderungen (S. 252)

Solche Ergebnisse sind es wert, wiederholt zu werden: Die Teams lieferten Software in *einem Drittel* ihrer normalen Zeit. Nach der Pilotstudie erreichten elf weitere Teams das gleiche Ergebnis.

Schlüsselidee

Persönliche Gespräche

Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im persönlichen Gespräch.
–Manifest für Agile Softwareentwicklung

Trotz des technischen Fortschritts bleibt das gemeinsame Gespräch von Angesicht zu Angesicht die effektivste Form der Kommunikation. Um es mit den Worten von [Cockburn 2006, Kap. 3] zu sagen: »Es gibt mehrere Achsen für die Wirksamkeit der Kommunikation. Jede Achse, die Sie entfernen, macht die Kommunikation weniger effektiv.«

- *Kollaboration ist besser als Konversation*. Mit »Kollaboration« meine ich gemeinsam zu arbeiten, mit einer geteilten Visualisierung oder einem anderen Artefakt. Sie sorgt dafür, Ideen Wirklichkeit werden zu lassen, und deckt Annahmen und Bedeutungsunterschiede auf, die mit Worten allein verdeckt geblieben wären.
- *Persönlich ist besser als virtuell*. Bei einem persönlichem Gespräch nehmen die Teilnehmenden winzige Hinweise wahr, wie die Bewegungen der Augenmuskeln und subtile Bewegungen der Körpersprache. Sie bewegen sich, kommunizieren durch ihre Haltung sowie Berührungen (z. B. eine Hand auf einer Schulter) und synchronisieren sich unbewusst. Ohne dass sie es merken, helfen diese Hinweise den Personen dabei, einander besser zu verstehen.

→

Schlüsselidee

- *Video ist besser als nur Audio*; *nur Audio ist besser als Text*. Bei Audiogesprächen setzen die Teilnehmenden Intonation und Pausen ein, um Humor, Bedenken und wichtige Punkte mitzuteilen. Bei Gesprächen per Video kommunizieren sie zusätzlich über Mimik und Gestik.
- *In Echtzeit ist besser als asynchron*. In einer Unterhaltung in Echtzeit können die Personen unterbrechen, etwas klären und das Gespräch in eine Richtung lenken.
- *Interaktiv ist besser als einseitig*. In einer interaktiven Unterhaltung fragen Leute nach, um verwirrende Punkte zu klären.

Entfernen Sie sie alle und Sie landen bei schriftlichen Dokumenten – keine Feinheiten, keine Interaktivität und die maximale Chance für Missverständnisse.

Geheimnisse der Zusammenarbeit

Um das Beste aus Ihrem Teamraum herauszuholen, stellen Sie sicher, dass Sie ein komplettes Team haben. Sie werden die Vorteile einer cross-funktionalen Zusammenarbeit nicht nutzen können, wenn die Menschen, mit denen Sie sprechen müssen, nicht Teil Ihres Teams sind. Stellen Sie für die Leute, die aufgrund ihrer Arbeit häufig den Teamraum verlassen müssen, sicher, dass eine Vertretung verfügbar ist. Produktmanager fallen tendenziell in diese Kategorie.

Selbst wenn Sie nicht über ein komplettes Team verfügen, bietet Ihnen die gemeinsame Arbeit in einem Teamraum neue Möglichkeiten, Ihre Zusammenarbeit zu optimieren. Hier sind einige meiner favorisierten Techniken:

Verwandtes Thema

Komplettes Team (S. 94)

Immer fragen, immer helfen

Wenn Sie bei einem Problem nicht weiterkommen und jemand anderes aus dem Team die Antwort kennt, bitten Sie diese Person um Hilfe. Es ergibt keinen Sinn, mit seinem Kopf gegen eine Wand zu rennen. Um das zu unterstützen, haben viele Teams eine Arbeitsvereinbarung: »Wir helfen immer, wenn ein Teammitglied fragt.«

Einige Leute hören diese Regel und befürchten, dass sie nicht mehr so produktiv sein werden. Zu einem gewissen Grad haben sie recht. Wenn Sie viel Zeit damit verbringen, Fragen zu beantworten, sind *Sie* vielleicht nicht mehr so produktiv.

Agilität konzentriert sich auf die Leistung von Teams, nicht auf individuelle Leistungen.

Aber Agilität dreht sich um das *Team*. Selbst wenn Sie mehr Zeit aufwenden, als sie einsparen, wird das Team im großen Ganzen produktiver sein.

Wie ist es mit Programmierung und anderen Arbeiten, die eine hohe Konzentration erfordern? Nach [DeMarco 2013] braucht eine Programmiererin nach einer Unterbrechung 15 Minuten oder mehr, um wieder in den Arbeitsfluss zu kommen. Würde eine Kultur, in der nach Hilfe gefragt wird, die allgemeine Programmierproduktivität nicht beeinträchtigen?

Das kann passieren. Gleichzeitig ist es aber eine der besten Möglichkeiten, die Teamleistung zu verbessern, wenn Programmierprobleme durch Fragen kurzfristig gelöst werden. Statt also Unterbrechungen zu vermeiden, sollten Sie Wege finden, zu verhindern, dass Unterbrechungen ablenken.

Verwandte Themen

Pair Programming (S. 448),
Mob Programming (S. 461)

Der beste Weg, zu verhindern, dass Fragen ablenkend wirken, ist Pair Programming oder Mob Programming einzusetzen. Bei Pair Programming beantwortet eine Person die Fragen, während die andere Person über das eigentliche Problem nachdenkt. Wenn die Unterbrechung vorbei ist, sorgt ein schnelles »Wo waren wir?« dafür, dass es weitergeht. Bei Mob Programming sind Ablenkungen noch weniger ein Thema; in der Regel entstehen gar keine Ablenkungen, weil alle gemeinsam arbeiten. Wenn doch eine Ablenkung aufkommt, geht die Person, die unterbrochen wird, einfach weg, während der Rest der Gruppe weiterarbeitet.

Verwandtes Thema

Ausrichtung (S. 165)

Wenn Pair oder Mob Programming keine Option ist, wird Ihr Team Arbeitsvereinbarungen festlegen müssen, die das Thema Unterbrechungen von Arbeiten, die hohe Konzentration erfordern, betreffen. Ein Ansatz besteht darin, einen Indikator zu verwenden, z. B. einen Kopfhörer aufzusetzen, wenn Sie vor Ort sind, oder eine Statuseinstellung vorzunehmen bei virtueller Arbeit, um zu vermitteln »Bitte weniger Unterbrechungen«. Aber erinnern Sie sich bitte daran, dass das Ziel weiterhin ist, die Leistungsfähigkeit des *Teams* zu maximieren, nicht die des Individuums.

Einsteigen und Aussteigen

Ein Teamraum ermöglicht es Ihnen, weniger Zeit in Meetings zu verbringen. Wenn Sie ein Problem mit anderen Teammitgliedern besprechen müssen, planen Sie kein Meeting; sagen Sie im Teamraum einfach, was Sie besprechen möchten. Stehen Sie entweder auf und sagen Sie etwas (wenn Sie sich an einem Ort befinden) oder senden Sie eine Nachricht in den Gruppenchat (wenn Sie remote arbeiten). Und fangen Sie dann einfach an, miteinander zu reden. Jede Unterhaltung sollte nur die betroffenen Personen einschließen und beendet werden, sobald das Problem behoben ist. Stellt sich währenddessen heraus, dass noch eine weitere Person aus dem Team gebraucht wird, bitten Sie sie teilzunehmen.

Wenn jemand eine Unterhaltung startet, müssen Sie sich nicht beteiligen. Hören Sie sich das vorgeschlagene Thema an und entscheiden dann, ob es sich um etwas handelt, das Ihre Teilnahme erfordert. Ebenso müssen Sie auch nicht dabei bleiben, wenn sich herausstellt, dass die Diskussion für Sie nicht relevant ist. Dann können Sie zu Ihrer Arbeit zurückkehren. Das wird das »Gesetz der zwei Füße« genannt: »Wenn Sie zu irgendeinem Zeitpunkt weder lernen noch einen Beitrag leisten können, gehen Sie dorthin, wo sie das können.«[8] Und andersherum! Wenn sich herausstellt, dass ein Gespräch für Sie relevant ist, nehmen Sie daran teil.

In einem physischen Teamraum ist es höflich, Personen, die sich konzentrieren, nicht durch Unterhaltungen zu stören. Die meisten Teamräume haben zu diesem Zweck einen separaten Bereich für Besprechungen. Er liegt nah genug, sodass das restliche Team mithören und sich ggf. einmischen könnte, ist aber auch ausreichend abgetrennt, sodass die Besprechungen nicht ablenken.

Virtuelle Teams haben das umgekehrte Problem: Es ist nicht möglich, die Gespräche der anderen mitzuhören. Wenn eine Unterhaltung startet, ist es normalerweise am effektivsten, diese aus dem Gruppenchat in eine Videokonferenz zu verlagern. Doch dann kann niemand mithören, was besprochen wird. Überlegen Sie daher, gelegentlich Zwischenstände im Gruppenchat zu teilen. So können andere Leute für sich entscheiden, ob sie bei der Besprechung vorbeischauen.

Visualisierungen erstellen

Wenn Menschen miteinander sprechen, haben sie jeweils ihr eigenes Modell von der Welt im Kopf. Wenn diese mentalen Modelle sich zu sehr unterscheiden, kommt es zu Missverständnissen.

Wandeln Sie Ihr internes Modell in ein externes Modell, um Missverständnissen vorzubeugen. Erstellen Sie eine Visualisierung, die jede Person sehen, sie mit dem eigenen mentalen Modell vergleichen und verändern kann. Zeichnungen auf einem Whiteboard funktionieren, aber Modelle, die die Zusammenarbeit fördern, sind noch besser. Karteikarten und Haftnotizen oder ihr virtuelles Pendant funktionieren am besten. Schreiben Sie Ihre Ideen auf Karten und bewegen Sie diese dann umher, um die Beziehungen zu visualisieren.

Sie finden Beispiele dieser Visualisierungen überall in diesem Buch, beispielsweise in den Praktiken zum visuellen Planen oder der Aufgabenplanung. Beschränken Sie sich aber nicht auf die Visualisierungen, die in diesem Buch beschrieben sind. Wann immer Sie eine Diskussion führen, insbesondere wenn da-

Verwandte Themen

Visuelles Planen (S. 217),
Aufgabenplanung (S. 265)

8. Das »Gesetz der zwei Füße« stammt aus Harrison Owens »Open Space Technology«, eine hervorragende Methode, um große Gruppen für produktive Diskussionen zu organisieren.

bei Verständnisprobleme oder Probleme der Übereinkunft bestehen, erstellen Sie ein Modell, das die Beteiligten bearbeiten können.

Gleichzeitiges Arbeiten

Beim gemeinsamen Arbeiten sollten sich die zu leistenden Beiträge nicht bei einer einzelnen Person aufstauen. Stellen Sie sicher, dass alle gleichzeitig einen Beitrag leisten können. Verhindern Sie beispielsweise beim Planning, dass eine Person am Computer sitzt und alles in ein elektronisches Planungswerkzeug einträgt.

Visualisieren Sie stattdessen den Plan mit Karteikarten oder deren virtuellem Äquivalent. Auf diese Weise können mehrere Personen zur selben Zeit neue Karten schreiben und verschiedene Personen können den Plan ändern – und ihre Änderungen diskutieren –, indem sie die Karten verschieben und auf sie zeigen.

> Die zu leistenden Beiträge sollten sich nicht bei einer einzelnen Person aufstauen.

Diese Art der gleichzeitigen Zusammenarbeit ist enorm effektiv. Es erfordert zwar, dass die Person, die normalerweise an der Tastatur sitzt, die Kontrolle abgibt. Doch sobald sie das tut, werden Ihre Diskussionen viel schneller verlaufen. Teams, die vor Ort arbeiten, werden sich ganz natürlich in kleine Gruppen aufteilen, um Themen von Interesse zu diskutieren. Sie schaffen damit zwei bis dreimal so viel Arbeit in derselben Zeit. Virtuelle Teams werden keinen so großen **Vorteil** daraus ziehen, denn es ist schwieriger, kleine Gruppenbesprechungen durchzuführen. Doch sie werden trotzdem effektiv sein.

Eine meiner Lieblingsmethoden für simultanes Arbeiten ist das *gleichzeitige Brainstorming*. Dabei fordert jemand die Gruppe auf, Ideen zu einem bestimmten Thema zu entwickeln, genau wie beim normalen Brainstorming. Wenn jemandem eine Idee einfällt, spricht er diese laut aus, schreibt sie auf eine Karteikarte und hängt sie dort hin, wo jeder sie sehen kann. (Eine Idee pro Karte, damit sie später leichter sortiert werden können.) Das laute Aussprechen der Ideen inspiriert die anderen zu neuen Ideen und wenn Sie die Ideen selbst aufschreiben, verhindert dies, dass die Gruppe durch eine Person, die mitschreibt, ins Stocken gerät.

Anmerkung

Denken Sie daran, Ideen während des Brainstormings nicht zu kritisieren. Brainstorming funktioniert am besten in zwei Phasen: Zuerst freie Ideenfindung, bei der alles möglich ist; als Zweites werden die Ideen verfeinert und gefiltert.

Manchmal lasse ich die Gruppe nach dem gleichzeitigen Brainstorming ein Affinitätsdiagramm (engl. *Affinity Mapping*) erstellen. Dafür nehmen Sie alle Karten, die Ihre Gruppe geschrieben hat, und verteilen diese zufällig auf einem Tisch oder

einem virtuellen Whiteboard. Anschließend verschieben Sie die Karten so, dass ähnliche Ideen nah beieinander liegen. Alle arbeiten zur selben Zeit und bewegen die Karten so, wie es aus ihrer Sicht passt. Am Ende sollten die Karten zusammenhängende Gruppen bilden, die Sie betiteln können.

Eine Variante des Affinity Mapping ist das »Mute Mapping«. Der Ablauf ist derselbe wie beim Affinity Mapping, außer dass niemand beim Bewegen der Karten sprechen darf. Es eignet sich gut, um Streitereien darüber zu verhindern, wo die Karten hinkommen sollen, und kann außerdem zu einigen lustigen pantomimischen Interaktionen führen.

Eine weitere Möglichkeit, Ihre Ideen nach dem Brainstorming zu filtern, ist die Punktabstimmung (engl. *Dot Voting*). Dabei erhält jede Person eine bestimmte Anzahl von Punkten. (Ich multipliziere die Anzahl der Auswahlmöglichkeiten mit drei und teile das Ergebnis dann durch die Anzahl der teilnehmenden Personen.) Alle stimmen zeitgleich ab, indem sie Punkte auf die Optionen setzen, die sie bevorzugen. Es ist in Ordnung, mehrmals für eine Option zu stimmen. Sie könnten bei vier Punkten zum Beispiel alle vier Punkte auf verschiedene Optionen verteilen, alle Punkte auf eine Option setzen oder irgendetwas dazwischen. Die Optionen mit den meisten Punkten gewinnen.

Zustimmung einholen

Was tun Sie, wenn Leute nicht einverstanden sind? Einseitige Entscheidungen schließen Menschen aus. Mehrheitsentscheide resultieren in einer enttäuschten Minderheit. Ein Konsens dauert zu lange und kann in einer Sackgasse enden.

Verwenden Sie stattdessen eine *Abstimmung nach Konsent*. Bei einer Abstimmung nach Konsent macht jemand einen Vorschlag, daraufhin stimmen alle ab mit »Ich stimme zu« (Daumen hoch bei einer Abstimmung vor Ort, »+1« in einem Gruppenchat) oder »Ich trage die Entscheidung der Gruppe mit« (Daumen zur Seite oder eine »+0«) oder »Ich stimme dem Vorschlag nicht zu *und möchte erklären, warum*« (Daumen nach unten oder »–1«). Um zu vermeiden, dass jemand versehentlich unter Druck gesetzt wird, können Sie optional bis drei zählen und anschließend alle ihre Stimme gleichzeitig abgeben lassen.

Wenn niemand mit »Ich stimme zu« abstimmt, wird der Vorschlag aufgrund mangelnden Interesses abgelehnt. Wenn auf der anderen Seite niemand mit »Ich stimme dagegen« abstimmt, wird dem Vorschlag zugestimmt. Sollte jemand ablehnen, erklärt diese Person ihre Einwände und die Gruppe passt den Vorschlag an, um darauf einzugehen. Solange es Einwände gibt, wird einem Vorschlag nicht zugestimmt.

Abstimmungen nach Konsent funktionieren aus zwei Gründen. Erstens lassen sie Leuten den Raum, ihre Unterstützung zu verweigern, ohne einen Vorschlag zu verhindern. Zweitens muss jemand, der sich stark genug fühlt, ein Veto gegen einen Vorschlag einzulegen, dieses auch begründen. Das gibt der Gruppe die Möglichkeit, auf die Bedenken einzugehen.

Experimenten zustimmen

Wenn Sie bemerken, dass eine Diskussion in Spekulationen ausartet, schlagen Sie ein konkretes Experiment vor.

Für einige Entscheidungen wird es keine offensichtlichen Antworten geben oder es gibt mehrere valide Optionen. Diskussionen über die Entscheidungen können leicht in endlose Spekulationen abdriften, was schiefgehen *könnte*.

Wenn Sie bemerken, dass eine Diskussion in Spekulationen ausartet, schlagen Sie ein konkretes Experiment vor. Beispielsweise enthält Extreme Programming die »Zehn-Minuten-Regel«: Wenn beim Pair Programming über eine Entwurfsentscheidung länger als zehn Minuten diskutiert wird, teilen sich die Personen auf, jeder schreibt temporären Code, der die Entwurfsidee veranschaulicht, und anschließend vergleichen sie die Ergebnisse.

Physische Teamräume

Teamräume können physisch oder virtuell sein. Sollte es für ein Team möglich sein, vor Ort zu arbeiten, bauen Sie einen physischen Teamraum auf. Er ist teurer als ein virtueller Teamraum, doch trotz der technologischen Fortschritte bleiben Gespräche von Angesicht zu Angesicht der effektivste Weg für die Zusammenarbeit von Teams.

Bjorn Freeman-Benson, eine Führungsperson im Technologiebereich, mit vielen Jahren Erfahrung darin, verteilt arbeitende Teams zu leiten, sagt: »Wir haben deutlich weniger Kreativität in unseren [verteilten Teams]. Wir mussten das Personal aufstocken, um das gleiche Maß an Kreativität zu erreichen … Das Wichtigste in allen Unternehmen, in denen ich bisher war, ist die kreative Leistung. [In einem Team, das verteilt arbeitet,] bekommen Sie weniger davon aufgrund der Reibung. Sie bekommen vielleicht mehr erledigte Arbeitspakete, aber mit Jira-Tickets lassen sich keine Rechnungen bezahlen« [Shore 2019].

Der Cocktailparty-Effekt

Ein Grund dafür, dass physische Teamräume effektiver sind, ist der *Cocktailparty-Effekt*, den [Cockburn 2006] *osmotische Kommunikation* nennt. Haben Sie sich jemals mit einer Person in einem vollen Raum unterhalten und aus dem Nichts Ihren Namen gehört? Obwohl Sie sich auf Ihre Unterhaltung konzentriert haben, hat Ihr Gehirn all den anderen Unterhaltungen im Raum Aufmerksamkeit geschenkt. Als es Ihren Namen hörte, spielte es die Klänge in Ihr Bewusstsein zurück. Sie hören nicht nur Ihren Namen, Sie hören auch einen Teil der Unterhaltung drum herum.

Stellen Sie sich ein *Delivering*-Team in einem Raum vor. Die Teammitglieder arbeiten in Paaren und unterhalten sich leise. Dann erwähnt jemand etwas über

die Verwaltung von Datenbankverbindungen und eine andere Programmiererin schaut auf. »Oh, Kaley und ich haben den Verbindungspool der Datenbank letzte Woche überarbeitet. Du musst die Verbindungen nicht mehr manuell verwalten.« Wenn Teammitglieder sich damit wohlfühlen, sich auf diese Art zu Wort zu melden, geschieht dies häufig – mindestens einmal am Tag – und jedes Mal spart es Zeit und Geld.

Gestaltung Ihres Teamraums

Gestalten Sie Ihren Teamraum so, dass er Zusammenarbeit fördert. Stellen Sie gerade Tische bereit, die es erlauben, beim Arbeiten nebeneinanderzusitzen, anstelle der L-Form mit dem Bildschirm in der Ecke. Bieten Sie viele Whiteboards und Platz an den Wänden an, um Ideen zu skizzieren und Diagramme aufzuhängen. Sorgen Sie dafür, dass es einen Bereich für Unterhaltungen gibt mit einem großen Tisch, auf dem das Team Karteikarten verteilen und Visualisierungen erstellen kann. Stellen Sie außerdem, wenn möglich, einen Projektor oder einen großen Fernseher für Gruppendiskussionen bereit, bei denen ein Computer verwendet wird.

Gruppieren Sie die Leute anhand der Gespräche, die sie mitbekommen sollten. Typischerweise sollten Entwickler (Programmierer, Tester, Betriebsleiter usw.) zusammensitzen. Kunden, die vor Ort sind, müssen nicht so dicht dabei sitzen, aber sie sollten doch nah genug sein, sodass sie bei Bedarf Fragen beantworten können.

In ähnlicher Weise sollten Sie Ihren Arbeitsplatz so gestalten, dass störende Geräusche auf ein Minimum reduziert werden. Der Bereich für Besprechungen sollte entfernt von den Schreibtischen liegen. Überlegen Sie sich, einen abgeschlossenen Raum für Telefonate und private Gespräche zur Verfügung zu stellen. Insbesondere wenn Sie Leute im Team haben, die viel Zeit mit Telefonaten und Videokonferenzen verbringen.

Achten Sie auch auf die menschliche Seite.

Achten Sie schließlich auch auf die menschliche Seite. Menschen fühlen sich wohler, wenn ihr Arbeitsplatz natürliches Licht, Pflanzen und Farbe enthält. Lassen Sie auch Raum für Individualität. Wenn die Teammitglieder keine zugewiesenen Schreibtische haben, wie es bei Mob und Pair Programming häufig der Fall ist, sollten Sie sicherstellen, dass es einen Platz für persönliche Gegenstände gibt. Stellen Sie Bücher zur Verfügung – wie dieses hier –, in denen die Leute blättern oder nachschlagen können.

Sorgen Sie dafür, dass alle Möbel beweglich sind und nicht festgeschraubt werden, damit die Teammitglieder ihren Arbeitsbereich an ihre Bedürfnisse anpassen können.

Mehrere Teams

Agile Teams produzieren einen Schwall an Gesprächen in ihren Teamräumen und gelegentlich wird auch ausgelassen gefeiert. Das wird die Teammitglieder vermutlich nicht stören, es könnte aber die Nachbarn stören. Achten Sie darauf, dass Ihr Team getrennt von der restlichen Organisation arbeitet.

Sollten Teams jedoch häufig zusammenarbeiten müssen, setzen Sie sie nebeneinander und stellen Sie sicher, dass die Teams sich hören können. Teams, die nicht miteinander arbeiten, sollten stärker voneinander getrennt werden – entweder durch Abstand, Trennwände oder Lärmschutzwände.

Persönliche Ausstattung und Zubehör

Statten Sie Ihren physischen Raum mit folgender Ausrüstung und Material aus. Investieren Sie in physisches Material, selbst wenn einiges davon durch elektronische Werkzeuge ersetzt werden kann. Es ist nicht sehr teuer und es erlaubt Ihrem Team, die Stärken der Zusammenarbeit von Angesicht zu Angesicht voll auszuschöpfen.

- Zwei große magnetische Whiteboards für Planungen. Ich verwende gerne ein einzelnes, doppelseitiges, etwa zwei Meter breites, magnetisches Whiteboard mit Rollen. Das macht es für Teams leichter, ihre Pläne in einen Besprechungsraum zu rollen. Es muss magnetisch sein, damit man Karteikarten mit Magneten leicht daran befestigen kann.
- Viel zusätzliche Fläche auf dem Whiteboard, vorzugsweise magnetisch, für Diskussionen und Diagramme
- Ein großer fortlaufender, laminierter Kalender (drei Monate oder mehr), um wichtige Daten zu markieren
- Schalldämmende Trennwände, um den Arbeitsbereich des Teams abzustecken und Lärmbelästigung zu verhindern, oder einen geschlossenen Teamraum
- Hocker oder andere Sitzgelegenheiten, um vorübergehend mit jemandem zusammenzusitzen
- Notizblöcke oder tragbare Whiteboards, um im Sitzen Ideen zu skizzieren
- Verschiedene Materialien und Gesprächsgegenstände, um Diskussionen und Interaktionen von Teammitgliedern zu inspirieren
- Flipchart-Papier, ein oder zwei Flipchart-Aufsteller und eine Möglichkeit, Flipchart-Papier an Wänden zu befestigen (z.B. blaues Klebeband, Plakatnadeln oder Reißzwecken)

- Karteikarten in verschiedenen Farben (mindestens 2.000 Stück von jeder Farbe). Stellen Sie sicher, dass alle aus dem Team die Farben unterscheiden können.[9]
- Haftnotizen in verschiedenen Farben, Größen und Formen
- Bleistifte, Kugelschreiber und Filzstifte auf Wasserbasis zum Beschreiben der Karten und Haftnotizen. Vermeiden Sie Permanentmarker, so wie Edding; sie haben einen strengen Geruch und werden ständig auf einem Whiteboard verwendet.[10]
- Trocken abwischbare Stifte für Whiteboards, auf Wasser basierende Stifte für Flipcharts und mit Wasser abwischbare Stifte für dauerhafte Whiteboard-Notizen und den fortlaufenden Kalender. Vermeiden Sie stark riechende Stifte.
- Magnete, um Karteikarten und Dokumente auf dem Whiteboard zu befestigen
- Ein Exemplar dieses Buches und anderes nützliches Referenzmaterial

Teams, die vor Ort Pair Programming einsetzen, benötigen auch Stationen zum gemeinsamen Programmieren (siehe Abschnitt »Pair Programming« auf Seite 448 für Details):

- Breite Tische, die es erlauben, nebeneinander zu arbeiten. Einige Teams bevorzugen eine Kombination aus Steh- und Sitztischen oder höhenverstellbare Tische.
- Ein für die Entwicklung geeigneter Computer an jeder Pairing-Station
- Zwei Tastaturen und Mäuse an jeder Station. Einige Leute bevorzugen es, ihre eigene Tastatur und Maus zu verwenden; stellen Sie in diesem Fall sicher, dass die USB-Zugänge der Computer gut erreichbar sind, denn die Entwicklerinnen werden zwischen den Pairing-Stationen mehrfach am Tag hin- und herwechseln.
- Mindestens zwei Bildschirme an jeder Station

Teams, die vor Ort arbeiten und Mob Programming einsetzen, brauchen eine Mobbing-Station (siehe Abschnitt »Mob Programming« auf Seite 461 für Details):

- Tische mit genügend Stühlen für alle Teammitglieder und zusätzlich Sitzplätze für eventuelle Gäste – und ausreichend Platz, sodass die Leute leicht die Sitzplätze tauschen können.

9. Einer von acht Männern leidet an einer Form von Farbenblindheit.
10. Profi-Tipp: Sie können Permanentmarker von einem Whiteboard entfernen, indem Sie mit einem für Whiteboards geeigneten Stift darüber schreiben und das Geschriebene direkt im Anschluss entfernen.

- Einen »Fahrersitz«[11], leicht zugänglich, mit einer Maus, Tastatur und einem für die Entwicklung geeigneten Computer
- Einen Sitz für den »Navigator«[12] am selben Tisch wie der Fahrersitz oder nah genug, sodass der Fahrer und der Navigator gut miteinander sprechen können.
- Mindestens einen Bildschirm, typischerweise mit 60"-Diagonale oder mehr, also groß genug, sodass Inhalte von allen Sitzplätzen aus gut sichtbar sind. Stellen Sie sicher, dass die Sichtbedürfnisse aller Beteiligten berücksichtigt werden.

> Kaufen Sie keine Software, die den agilen Lebenszyklus verwaltet.

Kaufen Sie *keine* Software, die den agilen Lebenszyklus verwaltet, oder andere Software zur Nachverfolgung, es sei denn, das Team fragt explizit danach. Warten Sie selbst dann einige Monate ab, bis das Team Erfahrungen darin gesammelt hat, die Praktiken für das Planning aus diesem Buch zu verwenden (siehe Abschnitt »Unternehmensweite Werkzeuge zur Nachverfolgung« auf Seite 378 für Details).

Beispiele für Teamräume

Der Arbeitsbereich, der in Abbildung 7–1 zu sehen ist, basiert auf einem Teamraum, den ich in der Zentrale von Spotify im Jahr 2015 gesehen habe. Jeder Raum hatte einen Arbeitsbereich mit vielen Whiteboards, einen Bereich für Besprechungen und einen Raum für vertrauliche Gespräche, der drei bis fünf Personen Platz bot. Außerhalb des Raums befand sich ein breiter Flur mit komfortablen Sofas und Stühlen und einer Garderobe.

Die Teamräume von Spotify waren die besten, die ich gesehen habe, doch sie hatten auch ein paar Schwächen, wie ich von Leuten, mit denen ich gesprochen habe, erfahren habe. Die Trennwand zwischen dem Teamraum und dem Flur sollte ursprünglich aus Glas sein, stattdessen war es eine Art Gitter (möglicherweise aufgrund von Brandschutzbestimmungen), dadurch konnten laute Unterhaltungen im Flur das Team stören. Außerdem war der Teamraum nicht flexibel: Auch wenn Spotify verschieden große Räume für Teams unterschiedlicher Größe hatte, mochten es die Teams nicht, den Raum zu wechseln, wenn sie größer oder kleiner wurden.

---

11. Anm. d. Übers.: Der *Fahrer* oder die *Fahrerin* (engl. Driver) ist beim Mob Programming die an der Tastatur sitzende Person, die die Anweisungen der restlichen Teilnehmenden umsetzt. Als *Fahrersitz* wird der Platz dieser Person mit Tastatur und Maus, um durch den Quellcode zu »fahren«, bezeichnet.
12. Anm. d. Übers.: Der *Navigator* leitet die Fahrerin verbal durch den Quellcode. Dies tut er in derart abstrakter Form, wie es für die Fahrerin und die restlichen Teilnehmenden möglich ist.

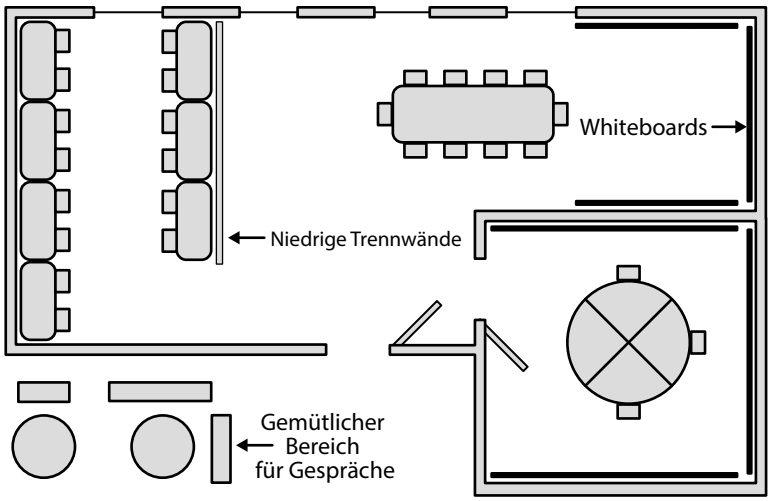

Abb. 7–1 *Von Spotify inspirierter Teamraum*

Der Arbeitsbereich, der in Abbildung 7–2 zu sehen ist, basiert auf einem Teamraum, der von einem aufstrebenden Start-up kreiert wurde, als dieses in ein neues Büro umzog. Sie verfügten nicht über das Budget für einen schicken Arbeitsbereich und begnügten sich mit dem, was sie hatten.

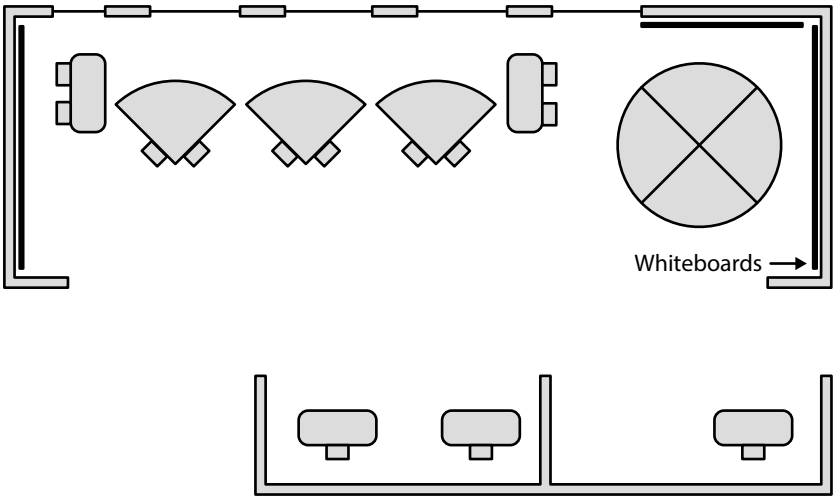

Abb. 7–2 *Ein kostengünstiger Teamraum*

Sie platzierten fünf Pairing-Stationen entlang einer langen Wand mit Fenstern nach draußen. Zwei der Tische waren Stehpulte und die anderen drei wurden aus Teilen eines alten runden Konferenztisches zusammengebaut. Ein zweiter runter Konferenztisch wurde für Gruppenbesprechungen genutzt. Der Raum wurde

durch Trennwände mit Whiteboards abgegrenzt. Für die Teammitglieder gab es ein paar Arbeitsnischen an der Seite und in der Nähe befanden sich Konferenzräume, die für vertrauliche Gespräche genutzt werden konnten.

Es war ein großartiger Arbeitsbereich mit einem schwerwiegenden Problem: Es gab eine Abtrennung zwischen den Programmierern und der Produktmanagerin. (Ich habe sie aus der Abbildung entfernt.) Die Folge war, dass die Produktmanagerin fast außerhalb des Teamraums saß – gerade noch! –, und das reichte aus, dass sie die Diskussionen des Teams weder mitbekam noch daran teilnehmen konnte. Die Teammitglieder bekamen keine Antworten auf ihre Fragen und hatten häufig Probleme mit Anforderungen.

Einführung eines physischen Teamraums

Einige Leute sträuben sich dagegen, in einen physischen Teamraum umziehen. Häufige Bedenken sind der Verlust der Individualität und Privatsphäre, eine vermeintliche Statusverschlechterung durch Aufgeben des eigenen privaten Büros und die Nichtanerkennung individueller Beiträge durch das Management.

Meiner Erfahrung nach genießen die meisten Menschen die Vorteile, die das Zusammensitzen mit sich bringt. Aber der Übergang dahin kann ein paar Monate dauern. Teams, mit denen ich gearbeitet habe und die in ihren Teamräumen viel Platz für Einzelpersonen vorgesehen haben, nutzten diesen letztendlich nicht. Die Fallstudie von Teasley kam zu ähnlichen Ergebnissen: Die Teammitglieder zogen anfangs Einzelbüros oder Arbeitsnischen dem offenen Arbeitsbereich vor, doch am Ende der Studie bevorzugten sie den offenen Arbeitsbereich.

Trotzdem ist es keine gute Idee, Menschen zum Zusammensitzen zu zwingen, in der Hoffnung, dass sie sich daran gewöhnen werden. Sprechen Sie stattdessen im Team über die Bedenken und die Kompromisse, in einen Teamraum zu wechseln. Diskutieren Sie, wie die Teammitglieder in einer agilen Umgebung bewertet werden – der Schwerpunkt sollte dabei auf den Beiträgen des Teams liegen, wie im »Collective Ownership« auf Seite 268 beschrieben – und welche Vorkehrungen Sie für die Privatsphäre treffen können.

Eine Managerin, mit der ich gesprochen habe, erstellte für ihr Team einen Raum mit Pairing-Stationen, aber verlangte von den Teammitgliedern nicht, diese zu benutzen. Einige Leute aus dem Team wollten Pair Programming ausprobieren und zogen dafür in den Teamraum. Mit der Zeit wechselten immer mehr Leute aus dem Team in den Teamraum, damit sie mit den Personen dort arbeiten konnten. Schließlich ist das gesamte Team zu Pair Programming und zum gemeinsamen Teamraum übergegangen, ohne dass es ihnen aufgezwungen wurde.

Virtuelle Teamräume

Wenn es sich bei Ihnen um ein Team handelt, das verteilt arbeitet, dann können Sie mit Online-Tools einen virtuellen Teamraum einrichten. Das funktioniert für

Teams, die hybrid arbeiten, und auch für die, die zum Teil verteilt arbeiten.[13] Doch seien Sie vorsichtig: Gespräche, die vor Ort stattfinden, schließen Personen aus, die aus der Ferne arbeiten.

Wenn einige Leute aus der Ferne arbeiten, müssen diejenigen, die sich vor Ort befinden, für die Zusammenarbeit auch den virtuellen Teamraum nutzen. Die Entscheidung für einen virtuellen Teamraum ist eine Entscheidung dafür, sich so zu verhalten, als würden alle Personen remote arbeiten.

Ausrüstung und Tools für verteiltes Arbeiten

Teams, die verteilt arbeiten, benötigen eine elektronische Version eines Arbeitsbereichs für das Team:

- Eine Software für Videokonferenzen in Echtzeit, zum Beispiel Zoom
- Eine Software, um kurze Nachrichten für die asynchrone Kommunikation zu verschicken, zum Beispiel Slack
- Eine Software für ein virtuelles Whiteboard, zum Beispiel Miro oder Mural, für die gleichzeitige, frei gestaltete Zusammenarbeit
- Kollaborative Versionen von aufgabenspezifischen Tools, wie Figma für UX und UI Design
- Eine Dateiablage, zum Beispiel Dropbox, Google Drive oder ein Wiki
- Günstige Tablets für gemeinsames Skizzieren auf dem Whiteboard
- Einen zusätzlichen Bildschirm oder ein Tablet für Videokonferenzen, damit sich die Leute beim gemeinsamen Arbeiten sehen können
- *Delivering*-Teams benötigen Tools für das gemeinsame Programmieren, zum Beispiel Tuple oder Visual Studio Live Share, die Pair oder Mob Programming unterstützen (siehe die Abschnitte »Pair Programming« auf Seite 448 und »Mob Programming« auf Seite 461 für Details).

Wie bei einem Arbeitsbereich vor Ort sollten Sie *keine* Software kaufen, die den agilen Lebenszyklus verwaltet, oder andere Software zur Nachverfolgung.

Gestaltung der Zusammenarbeit aus der Ferne

Zusammenarbeit fällt leicht, wenn wir gemeinsam vor Ort sind. Denselben Grad an Zusammenarbeit bei verteiltem Arbeiten zu erreichen, braucht eine sorgfältige Planung mit Finger-

Verwandtes Thema

Ausrichtung (S. 165)

13. Ein hybrides Remote-Team trifft sich an einigen Tagen vor Ort und arbeitet an den anderen Tagen verteilt. Bei einem Team, das zum Teil verteilt arbeitet, befinden sich einige Mitglieder vor Ort und andere entfernt.

spitzengefühl. Wenn Ihr Team seine Arbeitsvereinbarungen während der Session zur Charta der gemeinsamen Ausrichtung (engl. *Alignment Chartering*) festlegt, sollten Sie unbedingt besprechen, wie Sie zusammenarbeiten werden. Vergessen Sie nicht, dass es das Ziel ist, die Leistung des Teams zu maximieren, nicht die der einzelnen Person. Während die Arbeit vorangeht, sollten Sie Ihre Kommunikationstechniken regelmäßig überprüfen und verbessern.

Ich habe Leute, die Erfahrung in guter Zusammenarbeit haben, sowohl vor Ort als auch verteilt, gefragt, wie ihre Tricks für verteilte Zusammenarbeit sind.[14] Es gab mehrere ausgezeichnete Vorschläge:

- *Schaffen Sie Zeit für persönliche Kontakte.* Teams, die vor Ort arbeiten, bauen Freundschaft und gegenseitigen Respekt auf, das erlaubt ihnen, Entscheidungen schnell und effektiv zu treffen. Nehmen Sie sich in einem verteilten Team Zeit, um Kontakte zu knüpfen und über das Leben der anderen informiert zu bleiben. Möglichkeiten dafür sind virtuelle Kaffeepausen, um Spannungen abzubauen, ein spezieller Chatkanal für Begrüßungen und persönliche Mitteilungen beim Betreten und Verlassen des Büros sowie ein täglicher 30-minütiger Anruf zum Austausch oder Spielen. Ein Team hatte die Angewohnheit, die ersten fünf bis zehn Minuten jedes Meetings für Geselligkeit zu reservieren; die Leute konnten entweder früh erscheinen, um zu plaudern, oder nur für den inhaltlichen Teil kommen, je nachdem, wie ihre Stimmung war. Ein anderes Team nahm sich speziell Zeit, um Erfolge zu feiern.
- *Gewährleisten Sie Sicherheit.* Vor Ort können Leute Späße machen und sie selbst sein, ohne sich darüber zu sorgen, dass es sie eines Tages einholen wird. In einer virtuellen Umgebung kann alles aufgezeichnet werden. Etablieren Sie klare Richtlinien, wann es in Ordnung ist, eine Unterhaltung aufzunehmen oder zu teilen. Eine weitere Möglichkeit, um Sicherheit zu gewährleisten, ist, einen privaten Kanal einzurichten, zu dem nur das Team Zugang hat.
- *Machen Sie das Implizite explizit.* In einem persönlichen Gespräch gibt es viele menschliche Signale durch subtile Veränderungen in der Mimik und der Körpersprache. In einer verteilten Umgebung sind die Hinweise häufig unsichtbar. Machen Sie sie daher explizit. Zum Beispiel erstellte eine Person Karteikarten, die sie während Videokonferenzen hochhalten konnte, mit Sprüchen wie »+1«, »Bedenken«, »+1.000.000« und »Jaaa, los geht's«. (Haben Sie Spaß damit!) Geben Sie auch deutlich Ihre Verfügbarkeit an. Stellen Sie Nachrichten in den Gruppenchat, wo Sie sind, was Sie tun und ob Sie für Unterhaltungen verfügbar sind.

14. Vielen Dank an Dave Pool, Gabriel Haukness, Alexander Bird, Chris Fenton, Brian Shef und Dave Rooney für das Teilen ihrer Techniken auf *Twitter*. Ebenso großen Dank an Brent Miller, Dennis McMillan, Seth McCarthy, Jeff Olfert und Matt Plavcan für ihre Vorschläge.

- *Rüsten Sie Ihr Kommunikationsmedium auf.* Kommunikation mit geringer Bandbreite, wie z.B. für Textnachrichten, eignet sich gut für kurze Updates. Aber wenn es um wichtige Themen geht, kann das zu einem langen und langsamen Hin und Her führen. Wenn Sie bemerken, dass so etwas passiert, wechseln Sie zu einem Kommunikationsmedium mit mehr Bandbreite. Zum Beispiel wechselte ein Team ab dem Moment zu einer Besprechung per Video, wenn mehr als zwei Leute eine Diskussion im Chatkanal des Teams hatten.
- *Ermöglichen Sie gleichzeitige Gespräche.* Wenn mehrere Leute zur gleichen Zeit an einer Visualisierung arbeiten, teilen sie sich häufig in kleine Diskussionsgruppen von zwei bis drei Personen auf. Finden Sie Wege, um den gleichen Nutzen mit Ihren Onlinewerkzeugen zu erreichen. Videokonferenzwerkzeuge verfügen über Gruppenräume, und Anwendungen für Gruppenbesprechungen unterstützen das Erstellen separater Kanäle oder Themenbereiche.
- *Erstellen Sie eine »Wand«.* In einem Teamraum vor Ort werden Informationen, die jeder benötigt oder an die sich alle erinnern sollen, an eine Wand gehängt. Ziehen Sie für Ihr verteiltes Team auch in Erwägung, ein paar geteilte Dokumente mit solchen Informationen zu speichern.
- *Besorgen Sie Tablets.* Es ist viel bequemer, Diagramme mit einem Tablet als mit einer Maus oder einem Trackpad zu skizzieren. Und sie sind nicht besonders teuer. Besorgen Sie für jedes Teammitglied ein Tablet und melden Sie es bei Ihrem virtuellen Whiteboard an. Wenn Sie in einer Besprechung etwas skizzieren müssen, sind die Tablets sofort einsatzbereit.
- *Gehen Sie auf unterschiedliche Bandbreiten beim Zugang ein.* Die Internetverbindungen können sehr stark variieren. Nur 10 Kilometer können den Unterschied zwischen einer städtischen Hochgeschwindigkeitsbandbreite und einer schwankenden ländlichen Bandbreite ausmachen. Ganz zu schweigen von den Unterschieden zwischen verschiedenen Ländern. Sprechen Sie über die Bedürfnisse der Personen und finden Sie eine Kommunikationsstrategie, die für alle passt.

Unerfahrene Teammitglieder

Bjorn Freeman-Benson beschreibt »Das Problem mit unerfahrenen Leuten« als eine der drei Herausforderungen verteilter Teams:

> Wenn Entwickler gerade starten, haben sie viele Fragen. Sie wissen nicht, wie sie ihre Arbeit tun sollen. Gleichzeitig stehen sie am unteren Ende der Hierarchie. Sie zögern, sich einzumischen, und wollen nicht als Last empfunden werden.

> … Mit InVisions 100 % verteilten Teams [bei denen Bjorn CTO war] konnten Junior-Entwickler nicht sehen, was andere Entwickler tun. Sie hatten Angst zu stören. »Sie erstarrten und steckten fest«, sagte Bjorn, »bis sie viel später explizit danach gefragt wurden, was sie taten« [Shore 2019].
>
> – Bjorn Freeman-Benson: Drei Herausforderungen verteilter Teams

Verwandte Themen

Pair Programming (S. 448),
Mob Programming (S. 461)

Sorgen Sie dafür, dass unerfahrene Teammitglieder schnell eingearbeitet werden, ohne sich verloren oder im Weg stehend zu fühlen. Pair Programming oder Mob Programming sind beides hierfür hervorragende Techniken. Wenn die beiden Techniken nicht gut zu Ihrem Team passen, überlegen Sie tägliche Check-ins, 1:1-Mentoring oder andere Wege, um sicherzugehen, dass die unerfahrenen Teammitglieder nicht zurückgelassen werden.

Fragen

Mein physischer Teamraum ist zu laut für mich zum Konzentrieren. Was kann ich tun?

Verwandte Themen

Ausrichtung (S. 165),
Retrospektiven (S. 398)

Manchmal wird es im Team ein bisschen laut und unruhig. Es ist in Ordnung, um Ruhe zu bitten. Wenn Sie Ihre Arbeitsvereinbarungen in Ihrem ersten Treffen zur Ausrichtung diskutieren, sprechen Sie darüber, wie Sie sicherstellen, dass die Bedürfnisse aller erfüllt werden. Sollte das nicht ausreichen, adressieren Sie es auch in den Retrospektiven des Teams.

Erinnern Sie sich daran, dass Pair Programming hervorragend dafür geeignet ist, Ihre Aufmerksamkeit von Hintergrundgeräuschen abzulenken. Die Geräusche werden Sie nicht so sehr ablenken, wenn Sie sich mit Ihrem Programmierpartner austauschen. Mob Programming behebt das Problem völlig, indem es den Fokus aller auf dieselbe Sache richtet.

Unabhängig davon, wann eine Unterhaltung beginnt, das komplette Team stoppt das, was es gerade tut, um zuzuhören. Was können wir dafür tun, dass Leute nicht so leicht abgelenkt werden?

Insbesondere am Anfang ist es möglich, dass das komplette Team wirklich diese Unterhaltungen hören muss. Es hilft dabei, Kontext herzustellen, und schafft eine gemeinsame Grundlage. Mit der Zeit werden die Teammitglieder lernen, welche Unterhaltungen sie getrost ignorieren können.

Sollte es zu einem fortlaufenden Problem werden, versuchen Sie, insbesondere in einem Teamraum, sich ein wenig weiter vom restlichen Team zu entfernen, wenn eine Unterhaltung beginnt. Interessierte Teammitglieder können an der Besprechung teilnehmen, während der Rest des Teams seine Arbeit fortführt.

Voraussetzungen

Teammitglieder müssen eine gemeinsame Kernarbeitszeit haben, um effektiv zusammenzuarbeiten. Das gilt auch, wenn sie verteilt arbeiten. Wenn die Teammitglieder so weit verteilt sind, dass eine gemeinsame Arbeitszeit nicht möglich ist, haben Sie eigentlich mehrere Teams und Sie sollten Ihre Arbeit entsprechend gestalten (siehe Kapitel 6 für weitere Informationen zur Skalierung mehrerer agiler Teams).

Bei physischen Teamräumen ist der schwierigste Teil, den Raum zu schaffen. Typischerweise benötigen Sie dafür die Erlaubnis des Gebäudemanagements und des Managements. Das kann Wochen oder sogar Monate dauern, Sie sollten daher frühzeitig mit der Einrichtung des gemeinsamen Arbeitsbereichs beginnen.

Stellen Sie zusätzlich zur Erlaubnis des Gebäudemanagements und des Managements sicher, dass die Teammitglieder zustimmen, einen physischen Teamraum zu teilen. Veränderungen des Arbeitsplatzes können für einige Leute schwierig sein. Wenn sie gegen ihren Willen in eine neue Umgebung gezwungen werden, werden sie Wege finden, das Team zu verlassen. Selbst wenn das bedeutet, das Unternehmen zu verlassen (siehe Kapitel 5 für mehr Informationen dazu, wie Sie die Zustimmung der Teammitglieder einholen können).

Wenn Ihr Team kein Pair oder Mob Programming einsetzt, achten Sie darauf, Ihren Arbeitsbereich und Ihre Arbeitsvereinbarungen so zu gestalten, dass Lärm und Ablenkung minimiert werden.

Verwandte Themen

Pair Programming (S. 448),
Mob Programming (S. 461),
Ausrichtung (S. 165)

Indikatoren

Wenn Sie einen gelungenen Teamraum gestaltet haben:

- ist die Kommunikation zwischen Teammitgliedern schnell und effektiv;
- bekommen Sie mit, wenn Leute an Problemen arbeiten, bei denen Sie helfen können, und sie freuen sich über Ihre Teilnahme an der Diskussion;
- müssen Sie die Antworten nicht erraten. Wenn jemand im Team die Antwort kennt, können Sie fragen und bekommen eine schnelle Rückmeldung;
- bilden Teammitglieder spontan cross-funktionale Gruppen, um Probleme zu lösen;

- überwiegt im Team das Gefühl von Kameradschaft und des gegenseitigen Respekts.

Alternativen und Experimente

Bei dieser Praktik geht es tatsächlich um reibungslose Kommunikation und es macht keinen Unterschied, ob Sie einen realen Teamraum haben oder nicht. Aber wenn Sie noch nie die Erfahrung müheloser Zusammenarbeit in einem Teamraum – insbesondere in einem physischen Teamraum – erleben durften, probieren Sie es einige Monate aus, bevor Sie mit Alternativen experimentieren. Es fällt schwer, den Effekt wertzuschätzen, bevor man ihn selbst gesehen hat.

Verwandtes Thema

Mob Programming (S. 461)

Mob Programming erhöht das Maß an Zusammenarbeit noch einmal, indem alle im Team gemeinsam an einem Computer arbeiten. Es mag lächerlich klingen, aber in gewisser Weise ist es der »Einfache Modus« für Zusammenarbeit. Es ist besonders für verteilte Teams effektiv, die ansonsten viel Arbeit haben, bevor sie die Effektivität eines physischen Teamraums reproduzieren können.

Abseits des Arbeitens in Gruppen ist die Grundidee des gemeinsamen Arbeitsbereichs kaum zu übertreffen. Ihre besten Experimente werden sich mit Details befassen. Wie können Sie die Kommunikation verbessern? Wie können Sie regelmäßig angesetzte Treffen in kontinuierliche oder spontane Zusammenarbeit umwandeln? Wie können Sie Ihre Werkzeuge anpassen, sowohl physisch als auch virtuell, um neue Wege der Zusammenarbeit zu entdecken? Wie sieht es mit dem Arbeitsbereich aus? Gibt es Möglichkeiten, Möbel umzustellen oder Ihre Arbeitsvereinbarungen zu ändern, um die Kommunikation effektiver zu gestalten? Falls Sie bei der Arbeit feststellen, wo es in der Kommunikation Reibungsverluste gibt, dann experimentieren Sie, um diese zu beseitigen.

Weiterführende Literatur

Agile Software Development [Cockburn 2006] enthält ein hervorragendes Kapitel zu Kommunikation. Kapitel 3 »Kommunizierende, kooperative Teams« beschreibt die Ausstrahlung von Kommunikation, Kommunikationsqualität und viele andere Konzepte, die damit in Verbindung stehen, einen Teamraum zu teilen.

The Remote Facilitator's Pocket Guide [Clacey 2020] liest sich schnell und ist ein hilfreiches Buch zu Moderation. Es ist insbesondere auf das verteilte Arbeiten ausgerichtet, aber die Ratschläge darin sind auch für das persönliche Arbeiten wertvoll.