
Technische Projektleitung

Ich wurde vor vielen Jahren technische Projektleiterin. Damals war ich zum Senior Engineer ernannt worden und arbeitete in einem kleinen Team mit einer Reihe weiterer Senior Engineers zusammen. Die Ernennung zur Leiterin kam recht überraschend, eigentlich war ich nicht die erfahrenste Person im Team – weder nach Titel noch nach Betriebszugehörigkeit. Rückblickend hatte ich ein paar Vorteile. Ich war mehr als eine gute Entwicklerin. Ich war eine gute Kommunikatorin. Ich konnte klare Dokumente schreiben, Präsentationen halten, ohne einen Zusammenbruch zu erleiden, und Kollegen in anderen Teams und anderen Rollen erklären, was gerade geschah. Zudem konnte ich gut priorisieren. Ich wollte die Arbeit erledigt bekommen und entscheiden, was dann zu tun war. Und schließlich war ich dazu bereit, alles zu tun, was notwendig war, um voranzukommen. Vermutlich war diese pragmatische Dringlichkeit der entscheidende Faktor. Die Rolle der technischen Projektleiterin ist schließlich eine Führungsrolle, auch wenn es keine Managementposition ist.

Ich habe auch schon technische Projektleiter gesehen, die ins Straucheln geraten sind. Besonders im Gedächtnis geblieben ist mir eine Person, die ein toller Entwickler war und großartigen Code schrieb, es aber gehasst hat, mit Leuten zu reden, und sich leicht durch technische Details ablenken ließ. Ich beobachtete, wie sich dieser Projektleiter immer mehr in unwichtigen Dingen verhedderte, während der Produktmanager seine Chance sah und den Rest des Teams zu Features überredete, die schlecht entworfen waren und zu voreilig freigegeben wurden. Das Projekt war ein einziges Chaos, und was tat der technische Projektleiter? Er kümmerte sich um die nächste Refaktorisierung, weil er sicher war, dass die Probleme lediglich in der Strukturierung des Codes lagen. Sie kennen die Geschichte bestimmt – sie passiert überall. Die Idee ist, die Rolle des technischen Projektleiters automatisch dem erfahrensten Entwickler zu geben – demjenigen, der die komplexesten Features beherrscht oder den besten Code schreibt. Aber diese Idee ist ein Irrglaube, dem selbst erfahrene Manager verfallen. Der technische Projektleiter ist kein Job für eine Person, die sich voll und ganz auf die Details ihres eigenen Codes konzentrieren möchte. Aber wer ist dann der Richtige für diesen Job? Was erwarten wir von dieser Person?

Wie bei vielen Titeln in der Softwareentwicklung fehlt auch beim »technischen Projektleiter« eine allgemeine Definition. Ich kann hier nur aus meiner eigenen Erfahrung und der von anderen berichten. Meine Aufgabe als technische Projektleiterin bestand darin, zwar weiterhin Code zu schreiben, aber zusätzlich verantwortlich die Gruppe gegenüber dem Management zu repräsentieren, unsere Pläne zum Feature Delivery zu überprüfen und mich um viele Details des Projektmanagementprozesses zu kümmern. Obwohl ich nicht die erfahrenste Person im Team war, konnte ich technische Projektleiterin sein, weil ich dazu bereit und in der Lage war, die Verantwortlichkeiten der Rolle zu übernehmen, während der Rest des Teams mehr daran interessiert war, sich ausschließlich auf die Software zu konzentrieren, die es schrieb. Als mein Team bei Rent the Runway unsere Jobhierarchie für die Entwickler erstellte, entschieden wir uns bewusst dafür, die Rolle des technischen Projektleiters in Form einer Reihe von Eigenschaften zu definieren, die ein Entwickler an vielen Positionen auf der Karriereleiter annehmen konnte und nicht nur auf einer einzigen Ebene. Wir wählten diesen Ansatz, weil wir anerkennen wollten, dass die Rolle des technischen Projektleiters bei sich ändernden und sich weiterentwickelnden Teams von ganz unterschiedlich erfahrenen Entwicklern besetzt sein und vom einen zum anderen Programmierer weitergegeben werden könnte, ohne dass dabei unbedingt dessen funktionale Jobebene angepasst werden müsste. Der technische Projektleiter hat nicht in jeder Firma zwingend die gleiche Rolle – selbst Teams innerhalb eines Unternehmens können sie verschieden interpretieren. Wir wissen aber aufgrund des Titels, dass es sich zugleich um eine technische wie auch um eine Führungsrolle handelt, die häufig nicht einen permanenten Titel, sondern eher einen zeitlich begrenzten Satz an Verantwortlichkeiten umfasst. Nach all diesem Vorgeplänkel: Was *ist* nun ein technischer Projektleiter? Dies ist die Beschreibung, die wir bei Rent the Runway erstellt haben:

Die Rolle des technischen Projektleiters ist kein Schritt auf der Karriereleiter, sondern umfasst eine Reihe von Verantwortlichkeiten, die ein Entwickler übernehmen kann, sobald er die Senior-Stufe erreicht hat. Zu dieser Rolle kann Personalverantwortung gehören, muss es aber nicht. Wenn sie dazugehört, wird erwartet, dass der technische Projektleiter die Teammitglieder nach den hohen Managementstandards von RTR Tech betreut. Dazu gehören:

- *Regelmäßige (wöchentliche) 1-1-Meetings.*
- *Regelmäßiges Feedback zu Karriere, Zielerreichung, verbesserungswürdigen Bereichen und Anerkennung (wie zugesagt).*
- *Zusammenarbeit mit Mitarbeitern, um Bereiche mit Weiterbildungsbedarf zu ermitteln, und Unterstützung beim Lernen durch Projektarbeit, externes Lernen oder zusätzliches Mentoring.*

Hat ein technischer Projektleiter keine Personalverantwortung, wird trotzdem erwartet, dass er den anderen Mitgliedern des Teams durch Mentorship und Unterstützung beiseitesteht.

Der technische Projektleiter lernt, wie er ein guter technischer Projektmanager sein kann, dabei erhöht er seine Produktivität, indem er Aufgaben effektiv delegiert, ohne in Mikromanagement zu verfallen. Er konzentriert sich auf die Produktivität des

gesamten Teams und arbeitet daran, den Einfluss der durch das Team geschaffenen Produkte zu steigern. Er kann für das Team unabhängige Entscheidungen treffen und lernt, mit schwierigen Management- und Führungssituationen umzugehen. Zudem lernt er, effektiv mit Produkt-, Analytik- und anderen Geschäftsbereichen zusammenzuarbeiten.

Es ist nicht erforderlich, dass ein Entwickler als technischer Projektleiter tätig wird, um mit seiner Karriere voranzukommen, aber es ist der normale Weg, um als Entwickler von Senior Engineer 1 zu Senior Engineer 2 zu gelangen. Erforderlich ist dieser Schritt auf jeden Fall, um von Senior Engineer 2 nach Engineering Lead wechseln zu können. Realistisch gesehen ist es aufgrund der Notwendigkeit der Erfahrung mit Führung und Verantwortung sehr schwer, über den Senior Engineer 2 hinauszuwachsen, ohne je als technischer Projektleiter tätig gewesen zu sein.

Eine vielleicht bessere und kürzere Form dieser Beschreibung stammt von Patrick Kua aus seinem Buch »Talking with Tech Leads«:¹

Ein Leiter, der für ein (Software-)Entwicklungsteam verantwortlich ist und mindestens 30% seiner Zeit damit verbringt, zusammen mit dem Team Code zu schreiben.

Technische Projektleiter sind in der Position, ihre Erfahrung in einem größeren Rahmen einbringen zu können, sodass ihr gesamtes Team besser wird. Sie können unabhängige Entscheidungen treffen und eine wichtige Rolle bei der Koordination mit anderen nicht technischen Partnern spielen, die das Team eventuell hat. Vielleicht ist Ihnen aufgefallen, dass es hier nicht speziell um technische Arbeit geht. Es handelt sich um eine Senior-Developer-Rolle, aber man sollte nicht davon ausgehen, dass der erfahrenste Entwickler im Team automatisch diese Rolle bekommt. Sie können keine Führungsrolle einnehmen, wenn Sie sich nicht mit anderen Menschen beschäftigen wollen, und soziale Kompetenz ist gerade das, was sich der neue technische Projektleiter aneignen soll – viel mehr als weitere technische Expertise. Aber technische Projektleiter werden auch an einer wichtigen neuen technischen Fähigkeit arbeiten: dem Projektmanagement. Das Unterteilen eines Projekts hat große Ähnlichkeit mit dem Designen von Systemen, und das Erlernen dieser Fähigkeit ist auch für Entwickler nützlich, die kein Team führen wollen.

Finden Sie sich in der Rolle des technischen Projektleiters wieder, gratuliere ich Ihnen! Jemand ist der Meinung, dass Sie alles Notwendige mitbringen, um Ansprechpartner eines Teams zu sein. Jetzt ist es an der Zeit, ein paar weitere Fähigkeiten zu erlernen.

Ein technischer Projektleiter sein

Die Tätigkeit eines technischen Projektleiters ist eine Übung, Einfluss zu nehmen, ohne die entsprechende Autorität zu besitzen. Als technischer Projektleiter leite ich ein Team, aber wir berichten alle an den gleichen Entwicklungsmanager. Ich muss also nicht nur meine

¹ Patrick Kua: *Talking with Tech Leads*. Create Space Independent Publishing Platform 2015 (<https://leanpub.com/talking-with-tech-leads>)

Kollegen beeinflussen, sondern auch meinen Manager, um sicherzustellen, dass wir die richtigen Aufgaben priorisieren. Es gab bei mir eine Situation, in der das besonders schwierig war: Eines der ersten Projekte, das ich angehen wollte, nachdem ich technische Projektleiterin geworden bin, war das Stoppen der gesamten Feature-Entwicklung, damit wir uns auf die technischen Schulden konzentrieren konnten. Mir war klar, dass wir diese technischen Schulden schon viel zu lange vor uns herschoben – das Deployen neuen Codes war schwierig, das Betreuen der bestehenden Services aufwendig – und die Bereitschaftsdienste waren die Hölle. Ich war davon überzeugt, dass wir ein wenig Fahrt herausnehmen mussten, um später umso schneller sein zu können. Aber das ließ sich den anderen Entwicklern nicht gut verkaufen, denn sie wollten neue Features schreiben. Und auch mein Manager war darüber nicht glücklich, denn er bekam fortlaufend neue Feature-Wünsche von den Kunden. Ich konnte meine Ideen aber umsetzen, indem ich mich auf die jeweiligen Auswirkungen auf die einzelnen Teammitglieder konzentrierte. Bei manchen ging es darum, einen zuverlässigeren Service zu erhalten, andere wollten eine schnellere Iteration haben, und dritte wünschten sich, die Bereitschaftsdienste ruhiger zu gestalten, sodass sie nachts besser schlafen könnten. Beim Gespräch mit meinem Manager hob ich die reduzierten operativen Aufwände hervor, sodass wir in Zukunft mit dem Team mehr Features erstellen könnten.

Als ich technische Projektleiterin wurde, musste ich meinen Fokus verschieben. Bei der Arbeit geht es nun weniger um mich und meine Arbeit an technisch herausfordernden Ideen oder um das interessanteste Projekt, sondern um mein Team. Wie bringe ich es voran? Wie räume ich Hindernisse aus dem Weg, die es aufhalten? Das Überarbeiten von Code oder das Schaffen eines tollen neuen Features, das mir dabei geholfen hat, all mein technisches Können unter Beweis zu stellen, hat vielleicht mehr Spaß gemacht, aber damals musste das Team die technischen Schulden angehen und sich auf Operations konzentrieren. Das Team hat dann die Anzahl der kritischen Paging-Warnungen um 50% reduziert und im folgenden Quartal die Menge an Deploys nahezu verdoppelt.

– Caitie McCaffrey

Alle guten technischen Projektleiter kennen diesen einen verrückten Trick

Sie sind technischer Projektleiter – heißt, Sie wissen etwas über Software und Ihr Manager denkt, Sie seien reif für mehr Projektverantwortung. Aber die technischen Kniffe und die Erfahrung helfen nicht, wenn Sie nicht den wichtigsten Trick kennen, um ein guter technischer Projektleiter zu sein: den Willen, vom Code wegzutreten und herauszufinden, wie Sie Ihre technischen Zusagen mit der Arbeit zur Deckung bringen können, die das gesamte Team dafür erledigen muss. Sie dürfen sich nicht mehr nur auf Ihre *alten* Fähigkeiten verlassen, sondern müssen damit beginnen, *neue* zu erlernen – in erster Linie die Kunst des Abwägens.

Wohin auch immer Ihre Karriere Sie in Zukunft bringen wird – das Abwägen wird sehr wahrscheinlich eine Ihrer größten Herausforderungen sein. Wollen Sie über Ihre Arbeit selbst bestimmen und entscheiden können, wann Sie woran arbeiten, müssen Sie die Herrschaft über Ihre Zeit und deren Einsatz gewinnen. Schlimmer noch ist es, dass Sie häufig Dinge, die Sie kennen und mögen – wie zum Beispiel das Schreiben von Code –, gegen andere Dinge abwägen müssen, von denen Sie nicht wissen, wie sie funktionieren. Es ist ganz natürlich, Bekanntes zu bevorzugen. Haben Sie weniger Zeit für das, was Sie beherrschen, und müssen stattdessen Neues lernen, ist das unbequem.

Möglicherweise wird es schwierig, das Projektmanagement und den Überblick über die Arbeit im Team mit der direkten technischen Arbeit in Einklang zu bringen. Manchmal sind Sie mehr der Programmierer, manchmal eher der Manager. Durch Versuch und Irrtum müssen Sie lernen, wie Sie Ihre Zeit so einteilen, dass Sie passende Zeitabschnitte zum Arbeiten finden. Das Schlimmste, was Ihrem Zeitplan passieren kann, ist, zufällig in irgendeine Meetings gezerzt zu werden. Es ist schwierig, sich auf das Schreiben von Code zu konzentrieren, wenn man permanent durch Meetings unterbrochen wird.

Selbst bei sorgfältiger Planung werden Sie häufig keine Gelegenheit haben, sich über mehrere Tage auf Programmierprobleme zu fokussieren. Hoffentlich haben Sie bereits ein paar Tricks gelernt, um Ihre eigene Arbeit so aufzuteilen, dass Sie nicht so viele Tage am Stück für die Erledigung technischer Aufgaben benötigen. Sie wissen auch, dass es wichtig ist, Ihrem Team eine geeignete Arbeitszeitplanung zu ermöglichen, sodass es sich längere Zeit auf die Entwicklung konzentrieren kann, denn Ihre Kollegen *werden* sich mehrere Tage auf Codeprobleme fokussieren müssen. Ein Teil Ihrer Führungsrolle besteht darin, den anderen Stakeholdern, wie zum Beispiel Ihrem Boss und dem Produktmanager, dabei zu helfen, den Teamfokus zu respektieren und Meeting-Zeitpläne aufzusetzen, die die einzelnen Beteiligten nicht überlasten.

Grundlagen der technischen Projektleitung

Nehmen wir an, Sie tun sich für mehrere Wochen mit einem Produktmanager und einem Team mit vier weiteren Entwicklern zusammen, um eine neue Initiative zu starten. Der technische Projektleiter ist in diesem Szenario für eine Reihe von Dingen verantwortlich, die jeweils davon abhängen, wo Sie sich gerade im Lebenszyklus des Produkts befinden. Sicher, Sie werden Code schreiben und ein paar technische Entscheidungen treffen müssen. Aber das ist nur eine der Rollen, die Sie hier spielen, und vermutlich nicht einmal die wichtigste.

Die wichtigsten Rollen eines technischen Projektleiters

Ihre oberste Priorität als technischer Projektleiter ist, den Überblick über die Arbeit zu behalten, sodass Sie das Projekt voranbringen können. Wie schaffen Sie

es also, vom Organisieren und Planen des Codes, den Sie selbst schreiben müssen, zum Organisieren und Leiten des gesamten Projekts zu kommen?

Systemarchitekt und Businessanalyst

In den Rollen des Systemarchitekten und des Businessanalysten ermitteln Sie, welche Systeme am dringendsten verändert und welche Features unbedingt gebaut werden müssen, um die anstehenden Projekte auch liefern zu können. Ziel ist hier, eine Struktur bereitzustellen, auf der Schätzungen aufsetzen können und mit der sich die Arbeit organisieren lässt. Sie müssen nicht jedes einzelne Element eines Projekts perfekt ausarbeiten, aber es lohnt sich, Zeit damit zu verbringen, die externen Effekte und Probleme rund um ein Projekt zu durchdenken. *Für diese Rolle müssen Sie ein gutes Gespür für die Gesamtarchitektur Ihrer Systeme und ein solides Verständnis für das Designen komplexer Software haben.* Zudem müssen Sie vermutlich dazu in der Lage sein, *die Geschäftsanforderungen zu verstehen und diese in Software umwandeln zu können.*

Projektplaner

Projektplaner unterteilen die Arbeit grob in Deliverables. Haben Sie diesen Hut auf, müssen Sie lernen, effiziente Wege zu finden, um die Arbeit so aufzuteilen, dass das Team schnell vorankommt. Ein Teil der Herausforderung ist, so viel produktive Arbeit wie möglich parallel erledigt zu bekommen. Das kann knifflig werden, denn Sie sind es vermutlich gewohnt, nur über Ihre eigene Arbeit nachzudenken, statt sich um die Arbeit einer Gruppe von Leuten zu kümmern. Entscheidend ist, Stellen zu finden, an denen zuvor besprochene Abstraktionsebenen dafür sorgen können, dass parallel gearbeitet werden kann. Haben Sie zum Beispiel ein Frontend, das JSON-Objekte aus einer API konsumiert, sollte es nicht notwendig sein, dass diese API vollständig abgeschlossen ist, damit die Frontend-Entwicklung beginnen kann. Stattdessen einigen Sie sich auf das JSON-Format und beginnen die Programmierung und Benutzung dieses Formats mithilfe von Dummy-Objekten. Wenn Sie Glück haben, ist Ihnen dieses Vorgehen schon zuvor begegnet, und Sie können es einfach übernehmen. In diesem Stadium werden Sie *Informationen von den Experten* in Ihrem Team einholen wollen und mit denjenigen reden, die die betroffenen Teile der Software sehr gut kennen, sodass Sie bei den Details helfen können. Auch werden Sie damit beginnen, die *Prioritäten* als Teil dieses Prozesses zu ermitteln. Welche Elemente sind *entscheidend*, welche *optional*? Wie können Sie möglichst früh an den entscheidenden Komponenten arbeiten?

Softwareentwickler und Teamleiter

Softwareentwickler und Teamleiter schreiben Code, kommunizieren Probleme und delegieren. Im Laufe des Projekts werden sich unerwartete Probleme zeigen. Manchmal fühlen sich technische Projektleiter dazu verleitet, den Helden zu spie-

len, indem sie diese Hindernisse selbst angehen. Um alles in Ordnung zu bringen, verbringen sie dann viel zu viel Zeit im Büro. In Ihrer Position als technischer Projektleiter sollten Sie weiterhin Code schreiben, aber nicht zu viel. Auch wenn Sie gern selbst ein Kaninchen aus dem Hut zaubern wollen, müssen Sie ein entsprechendes Hindernis zuerst kommunizieren. Ihr Produktmanager sollte so früh wie möglich alle denkbaren Herausforderungen kennen. Holen Sie sich bei Bedarf Ihren IT-Entwicklungsmanager dazu. In einer guten Firma muss man sich nicht dafür schämen, frühzeitig auf Probleme aufmerksam zu machen. Teams scheitern schnell mal, weil sie sich selbst in ein Feature verbeißen, auf das der Produktmanager durchaus verzichten könnte. Nähert sich ein großes Projekt seinem Lieferzeitpunkt, wird es immer Kompromisse bezüglich der Funktionalität geben müssen. Suchen Sie nach Gelegenheiten, *Arbeit zu delegieren*, insbesondere wenn es Teile des Systems gibt, die Sie eigentlich selbst bauen wollten, zu denen Sie aber bisher noch nicht gekommen sind.

Wie Sie aus dieser Beschreibung ersehen, müssen Sie als technischer Projektleiter Softwareentwickler, Systemarchitekt, Businessanalyst und Teamleiter sein, der weiß, wann er etwas allein erledigt und wann er die Arbeit an andere delegiert. Glücklicherweise müssen Sie all diese Rollen nicht gleichzeitig erfüllen. Es mag zunächst ungewohnt sein, aber Zeit und Erfahrung werden helfen, eine gute Balance zu finden.

Frag den CTO: Ich hasse es, technischer Projektleiter zu sein!

Ich dachte, es wäre toll, technischer Projektleiter zu werden, aber jetzt erwartet mein Manager von mir, alle Details für den Projektstatus zusammenzusuchen und ihm Bescheid zu sagen, wann Dinge erledigt sein werden – und ich hasse das. Warum hat mir keiner gesagt, dass es als technischer Projektleiter so furchtbar ist?

Ich weiß, die neuen Verantwortlichkeiten sind hart. Dieses spezielle Problem bezeichne ich gerne als den »Stein des Triumphs« (Simpsons-Fans werden das verstehen). Dabei handelt es sich um eine Metapher für eine Anerkennung, bei der man später feststellt, dass sie einen hohen Preis hat. Das gilt zwar in vielen Stadien einer Entwickler-Führungskarriere, aber der technische Projektleiter muss ohne Zweifel den größten Stein mit sich herumschleppen. Nur sehr selten erhält er eine Gehaltserhöhung oder einen neuen Titel, und Leute, die noch nie zuvor technischer Projektleiter waren, haben meist keine Ahnung davon, wie schwer die neue Verantwortung wiegen kann. Schon bei der Definition der Position habe ich erwähnt, dass sie in vielen Firmen eher als temporärer Titel angesehen wird, als Satz von Verantwortlichkeiten, den Sie in Ihrer Karriere immer wieder einmal übernehmen und auch wieder abgeben. Die Position kann als Sprungbrett in höhere Senior-Positionen dienen, aber selbst bringt sie meist keine direkte Belohnung mit sich.

Warum trägt man in der Rolle des technischen Projektleiters so eine schwere Last? Der technische Projektleiter hat eine viel umfangreichere Verantwortung als ein Senior-Developer. Er wird gerufen, um Architekten bei einem Projekt zu helfen und dann die

eigentliche Planung der Arbeit Schritt für Schritt durchzuführen. Der technische Projektleiter soll sicherstellen, dass das Team die Anforderungen des Projekts vollständig versteht, dass die Aufgaben geplant werden und dass das Team effektiv ist und gut vorankommt – und all das, ohne zwingend Managementverantwortung und meist auch kein spezifisches Training erhalten zu haben. Realistisch gesehen, erwarten die meisten Manager von ihren technischen Projektleitern, dass sie weiterhin fast genauso viel Code schreiben wie zuvor. Man hat also im Allgemeinen einfach mehr Verantwortung und mehr Aufgaben. Sind Sie das erste Mal technischer Projektleiter, werden Sie eher in Arbeit ertrinken.

Also: Herzlichen Glückwunsch, Sie haben jetzt den Stein des Triumphs erhalten! Glücklicherweise wird Sie das Herumschleppen dieser Last schließlich stärker machen und Ihnen Fähigkeiten verleihen, die Sie brauchen, um in Ihrer Karriere voranzukommen. Es wird nicht immer so schwer sein, wie es jetzt aussieht.

Projekte managen

Ich kann mich noch sehr lebhaft an meine ersten Erfahrungen mit dem Management eines komplexen Projekts erinnern. Damals war ich erstmals technische Projektleiterin, und mein Team hatte eine sehr vielschichtige Aufgabe vor sich. Wir hatten ein bestehendes System, das an die Grenzen seiner Belastbarkeit gelangt war. Nachdem wir wirklich alles andere ausprobiert hatten, entschieden wir uns, herauszufinden, wie wir es auf mehreren Rechnern parallel laufen lassen konnten. Das war in den sehr frühen Tagen verteilter Systeme, als die meisten Softwareentwickler wirklich nicht allzu viel darüber wussten, wie dort die Best Practices aussehen. Aber wir hatten ein tolles Team mit vielen klugen Leuten und waren zuversichtlich, es zu schaffen.

Tatsächlich fanden wir es – langsam, aber sicher – heraus. Wir dachten lange über das Design und das sinnvolle Aufteilen der Berechnungen auf verschiedene Rechner nach. Und dann holte mich eines Tages mein Chef Mike in sein Büro und erzählte mir, ich müsse einen Projektplan machen.

Das war eine der schlimmsten Erfahrungen meines Lebens.

Ich musste mich um diese unglaublich komplizierten Aufgaben kümmern und versuchen, herauszufinden, welche Abhängigkeiten es zwischen ihnen gab und welcher Art sie waren. Wie würden wir das Ganze in das komplexe Test-Framework einbinden können, das wir verwendeten? Wie würden wir es deployen? Wann mussten wir die Hardware bestellen, um die Tests ausführen zu können? Wie lange würden die Integrationstests benötigen? Und immer mehr solcher Fragen. Ich ging in Mikes Büro. Wir saßen uns am Schreibtisch gegenüber und besprachen alle Aufgaben, Termine und Aufteilungen. Er half mir bei manchen und schickte mich dann mit den Teilen wieder weg, die noch überarbeitet werden mussten.

Spaß machte das nicht. In Erinnerung geblieben ist mir eine Folge frustrierender und nerviger Schritte, bei denen ich meine Unsicherheit und meine Angst vor Fehlern und übersehenen Aufgaben überwinden musste, um einen Plan aufzustellen, der von Mike als gut beurteilt werden würde. Dann drehten wir eine weitere mühsame Runde und brachten alles in eine Form, die wir dem Führungsteam vorstellen konnten, sodass sie den Plan akzeptieren würden. Es brachte mich fast um. Und es war eine der wichtigsten Lektionen meiner Karriere.

Erspart man sich mit agiler Softwareentwicklung nicht gerade die Notwendigkeit des Projektmanagements? Nein. Agile Softwareentwicklung ist eine tolle Möglichkeit, die Arbeit neu zu durchdenken, weil es Sie zwingt, sie in kleinere Häppchen zu unterteilen, diese durchzuplanen und neue Werte inkrementell bereitzustellen, statt alles auf einmal auszuliefern. Nichts davon sorgt dafür, dass Sie Projektmanagement außen vor lassen können. Sie werden Projekte haben, die – aus welchen Gründen auch immer – nicht in einem einzelnen oder in zwei kleinen Sprints abgeschlossen werden können. Sie werden die Projektdauer für Ihr Managementteam abschätzen und Details dazu liefern müssen, warum das Ganze so lange dauert. Es gibt Projekte – meist enthalten sie Begriffe wie *Infrastruktur*, *Plattform* oder *System* –, für die Architekturüberlegungen oder ausgesprochen komplexe Planungen notwendig sind. Bei solchen Projekten, die auch noch viele unbekannte Parameter und wenig flexible Deadlines enthalten, werden Sie merken, dass diese nur schlecht in Standard-Agile-Prozesse passen.

Im Laufe Ihrer Karriere werden Sie verstehen müssen, wie Sie Arbeit unterteilen, deren Komplexität weit über das hinausgeht, was Sie als Einzelperson tun können. Das Projektmanagement für ein lang laufendes, teambasiertes Projekt ist nicht für alle spaßig. Ich finde es mühsam und manchmal irgendwie beängstigend. Ich möchte Dinge bauen und Werte schaffen, statt darüber nachzudenken, wie ich ein Projekt aufteile, das im Detail noch erstaunlich unspezifisch ist. Ich habe Angst, dass ich verantwortlich gemacht werde und etwas wichtiges vergessen könnte, was dazu führt, dass das Projekt scheitert. Aber die Alternative ist, dass das Projekt dann zwar auch fehlschlägt, aber nur langsamer.

Projektmanagement ist nicht im Detail für jede einzelne Aufgabe notwendig, und es wird in manchen Unternehmen auch überstrapaziert. Eigentlich mag ich das Hinzuziehen von Projektmanagern nicht einmal, weil sie Entwicklern häufig als Krücke dienen, um nicht lernen zu müssen, wie sie selbst ihre zukünftigen Aufgaben durchdenken und die richtigen Fragen nach dem Was und Warum stellen. Die Anwesenheit von Projektmanagern bedeutet zudem, mehr Wasserfallprojekte zu bekommen, statt einen agilen Prozess verfolgen zu können. Aber Projektmanagement muss trotzdem stattfinden. Als technischer Projektleiter sollten Sie es erledigen, wenn es notwendig ist – insbesondere für technisch besonders herausfordernde Projekte.

Der Wert einer Planung liegt schließlich nicht darin, den Plan perfekt umzusetzen, jedes Detail im Voraus zu bedenken oder die Zukunft vorherzusagen – der Plan

zwingt Sie einfach dazu, über das Projekt ausreichend nachzudenken, bevor Sie sich hineinstürzen und sehen, was geschieht. Das Ziel ist eine gewisse Voraussicht in Bereichen, in denen Sie vernünftig planen können. Der Plan selbst – so genau er auch sein mag – ist weniger wichtig als die Zeit, die Sie für das Planen investiert haben.

Zurück zu meiner ersten Projektmanagementenerfahrung. Lief das Projekt genau nach Plan? Natürlich nicht. Es gab Hindernisse, Fehler, unerwartete Verzögerungen und Dinge, die wir vergessen hatten. Erstaunlicherweise schafften wir es trotzdem, das Projekt nahezu pünktlich fertigzustellen, und es bedurfte keiner durchgearbeiteten Nächte vor dem Ende. Wir konnten die Änderungen umsetzen, die notwendig waren, um dieses komplexe System in ein verteiltes, deploybares Artefakt umzuwandeln, während wir gegen den Master-Branch arbeiteten, an dem auch noch 40 andere Entwickler eigene Änderungen vornahmen. Das gelang nur, weil wir ein tolles Team waren und einen Plan besaßen. Wir hatten uns überlegt, wie der Erfolg aussehen müsste und welche Risiken es gäbe, die den Plan scheitern lassen könnten.

Parallel zu dieser frustrierenden Reihe von Meetings mit Mike hatte ich eine eigene Serie von Projektplanungsmeetings, bei denen ich an Mikes Platz saß und mir gegenüber Carlo, Alicia oder Tim. Sie waren selbst genauso frustriert, weil dem Plan die Details fehlten. Nach diesen Meetings kümmerten sie sich um die unangenehme Aufgabe, über Dinge nachzudenken, die kein Code waren und die sich nicht perfekt vorhersagen ließen. Dank dieser Arbeit führten sie selbst komplexe Projekte zum Erfolg und sind nun besser dazu in der Lage, umfangreichere Systeme zu bauen und größere Teams zu führen, weil sie verstanden haben, was das Unterteilen eines Projekts wirklich bedeutet.

Nehmen Sie sich Zeit für Erklärungen

Einer der letzten Schritte bei einer Promotion ist deren Verteidigung. Hier präsentieren Sie als Doktorand nach Jahren der Forschung einer Expertenrunde aus Ihrem Fachgebiet Ihre Ergebnisse, und diese bewertet dann, ob Ihre Erkenntnisse einen Dokortitel wert sind. Vor Jahren durfte ich selbst einen PhD in Mathematik von einer der renommiertesten Fakultäten für angewandte Mathematik in den USA erhalten. Einer der Prüfer war ein angesehener Mathematiker aus dem Bereich der numerischen Analyse. Er sagte nach meiner (erfolgreichen) Verteidigung etwas zu mir, dass mir während meiner beruflichen (nicht mathematischen!) Karriere im Gedächtnis geblieben ist: »Ihre Dissertation war eine der übersichtlichsten und klarsten, die ich in den letzten Jahren gelesen habe. Danke sehr!« Ich hatte angenommen, dass er als Weltklasse-Mathematiker alles wissen würde und nur schauen wollte, wie meine Dissertation »ausgehen würde«. Aber er erklärte mir, dass er das nur konnte, weil ich mir die Arbeit gemacht hatte, die grundlegenden Ideen des Problemraums und die Motivation hinter meinen Ideen zu erläutern. Diese Lektion habe ich nie vergessen. Ich arbeite nun seit vielen Jahren im Softwareumfeld und in großen Firmen und kann diesen Hinweis immer besser würdigen.

Wir glauben, dass das Management »versteht«, was wir als Entwickler machen. »Die müssen doch nur den Code lesen!« Die Software, mit der wir jeden Tag verbringen, sollte jedem, der im Technologieumfeld unterwegs ist, offensichtlich verständlich sein, oder? Nein. Technologiemanager stellen (hoffentlich) die besten Leute ein, damit diese die schwierigen Probleme lösen. Aber sie »verstehen« nicht alles. Ich bin immer wieder davon überrascht, wie dankbar Entwicklungsmanager im oberen Management sein können, wenn ich ihnen ein paar grundlegende moderne Ideen erkläre (zum Beispiel, worum es bei diesem NoSQL-Kram eigentlich geht und warum ich mir darüber Gedanken machen sollte), ohne sie dabei herablassend zu behandeln.

Gerade neulich fragte mich ein Senior-Businessmanager privat, warum es für uns so wichtig sei, unsere klassisch deployte Fat-Client-Architektur auf eine gehostete Plattform zu migrieren. Er stand intern unter starkem Druck, diese Aufwände mit Budget zu versehen, hatte aber keine Ahnung, warum das überhaupt notwendig ist. Zudem war es ihm vermutlich zu peinlich, öffentlich zu fragen. Ich verbrachte zwei sehr produktive Stunden damit, es ihm zu erklären (ohne PowerPoint). Mittlerweile lasse ich keine Gelegenheit aus, Grundlagen von Technologien und die Motivation für bestimmte Aktivitäten erfahrenen und unerfahrenen Kollegen zu erklären. Sie lernen etwas dabei, ohne sich unwürdig vorkommen zu müssen, sie lernen, meiner Einschätzung und meinen Ratschlägen zu vertrauen, und wir bringen Änderungen besser voran. Es ist daher sehr wichtig, sich die Zeit zu nehmen, Dinge zu erläutern.

– *Michael Marçal*

Ein Projekt managen

Beim Projektmanagement geht es darum, ein komplexes Ziel in kleinere Einheiten herunterzubrechen, diese grob in eine möglichst effektive Reihenfolge zu bringen, in der sie erledigt werden sollten, herauszufinden, welche Einheiten parallel und welche sequenziell abzuarbeiten sind, und die Unwägbarkeiten des Projekts möglichst klein zu halten, die den Erfolg verzögern oder ganz verhindern könnten. Sie spüren Unsicherheiten auf, versuchen, Unbekanntes aufzuklären, und erkennen, dass Sie in diesem Prozess trotz größter Mühe Fehler machen und Dinge vergessen werden. Dies sind ein paar Hinweise dazu:

- 1. Teilen Sie die Arbeit auf.** Nehmen Sie sich eine Tabellenkalkulation, einen Gantt-Chart oder was immer für Sie sinnvoll ist, und beginnen Sie damit, Ihre großen Ziele (zum Beispiel das Neuschreiben Ihres Rechnungssystems) in Aufgaben zu unterteilen. Starten Sie bei den größten Aufgaben, teilen Sie diese in kleinere ein und diese wiederum in noch kleinere. Sie müssen dabei nicht alles selbst machen. Gibt es Teile des Systems, die Sie nicht so gut verstehen, bitten Sie jemanden, der hier kompetenter ist, um Hilfe. Brechen Sie die größten Teile herunter und schauen Sie dann erst einmal, in welcher Reihenfolge diese entstehen müssen. Was kann sofort begonnen werden. Übergeben Sie

solche Aufgaben an die Mitarbeiter, die in der Lage sind, sie in Ticket-taugliche Einheiten herunterbrechen.

2. **Wühlen Sie sich durch Details und Unbekanntes.** Der Trick beim Projektmanagement ist, nicht dann aufzuhören, wenn Sie das Gefühl haben, festzustecken, oder genervt sind. Wie ich schon sagte: Es ist ermüdend und nervig – und vielleicht etwas, das Sie nicht so gut können. Also kämpfen Sie sich durch solche unsicheren Abschnitte, durch Langeweile und Schmerz. Ein guter Manager wird sich mit Ihnen zusammensetzen und Ihnen Schwächen in der Planung zeigen, Fragen stellen, um auf Wichtiges hinzuweisen, oder vielleicht sogar Teile mit Ihnen zusammen angehen. Uns macht das auch keinen Spaß, aber es gehört zu den Lektionen, die man lernen muss. Gehen Sie das Unbekannte an, bis Sie wirklich das Gefühl haben, dass Sie mit noch mehr Zeit keinen weiteren Mehrwert hinzufügen.
3. **Führen Sie das Projekt durch und passen Sie den Plan dabei an.** Der Wert eines guten Planungsprozesses liegt darin, dass er Ihnen dabei hilft, ungefähr zu erkennen, wie weit das Projekt schon ist und was noch zur Fertigstellung fehlt. Wenn es Verzögerungen gibt (und die wird es geben), informieren Sie jeden über den Status. Aber jetzt können Sie nicht nur schätzen, wie viel noch zu tun ist, sondern klar auf die Meilensteine hinweisen, die erreicht wurden, und die zu erwartende verbleibende Arbeit umreißen.
4. **Nutzen Sie die Erkenntnisse aus dem Planungsprozess, um Änderungen an Anforderungen richtig einzuplanen.** Sie haben durch das Unterteilen des Projekts auf Grundlage der ursprünglichen Anforderungen viel gelernt. Wenn sich die Anforderungen zwischendurch ändern, nutzen Sie diese Einblicke und wenden Sie sie auf die Änderungen an. Bedeuten die Änderungen ein deutlich erhöhtes Risiko und führen sie zu umfangreichen Neuplanungen oder schlicht viel mehr Arbeit, dann weisen Sie deutlich auf diesen Mehraufwand hin. Arbeiten Sie auf eine feste Deadline hin, ist ein grober Überblick über die Zusatzarbeit beim Priorisieren, Zurechtstutzen und Vereinfachen der Arbeit hilfreich, um den besten Kompromiss in Bezug auf Features, Qualität und Lieferzeitpunkt zu erreichen.
5. **Überarbeiten Sie die Details, wenn Sie sich dem Abschluss nähern.** Zum Ende des Projekts kommt der Überdruß zurück. Es ist an der Zeit, sich wirklich den letzten Details zu widmen. Was fehlt? Was wird getestet? Was wird überprüft? Führen Sie einen Risikocheck bzw. ein *Premortem* durch – eine Übung, bei der Sie all die Dinge durchgehen, die beim Bereitstellen dieses großen Projekts schiefgehen könnten. Entscheiden Sie, wo die Grenze für »gut genug« ist, machen Sie das bekannt und stehen Sie dazu. Stoppen Sie Arbeiten, die über dieses »gut genug« hinausgehen, und fokussieren Sie das Team auf die wichtigsten letzten Details. Erarbeiten Sie einen Launch-Plan und dazu einen Rollback-Plan. Und vergessen Sie nicht, am Ende zu feiern!

Frag den CTO: Ich bin nicht sicher, ob ich ein technischer Projektleiter sein möchte

Meine Managerin drängt mich dazu, eine technische Projektleitung zu übernehmen. Sie möchte, dass ich ein großes Projekt leite. Würde ich diese Rolle übernehmen, weiß ich, dass ich viel weniger Zeit hätte, Code zu schreiben, weil ich in vielen Meetings anwesend sein und mich mit einem Haufen Koordinationskram herumschlagen müsste. Ich glaube, ich will das nicht, aber wie kann ich das entscheiden?

Ich habe eine ziemlich eindeutige Meinung dazu, wenn es darum geht, jemanden in Managementpositionen zu drängen: Man sollte es nicht tun. Wenn Sie nicht dazu bereit sind, Managementverantwortung zu übernehmen, bringt es nichts. Es ist nichts Verkehrtes daran, in der technologischen Schiene zu bleiben, insbesondere wenn Sie das Gefühl haben, noch viel lernen zu müssen, um ein Experte zu sein.

Gute Manager sind immer auf der Suche nach talentierten Leuten, die eine (eventuell weitreichendere) Führungsrolle einnehmen könnten, aber manchmal führt das dazu, dass Mitarbeiter vom Coden weggeholt werden, bevor sie dazu bereit sind. Diese Praxis kann sehr negative Auswirkungen auf die Karriere haben. Auf höheren Karriereebenen kann es für die, die »nicht technisch genug sind«, ausgesprochen schwierig werden, eine Beförderung mit noch mehr Verantwortung zu erhalten. Es ist viel einfacher, sich auf seine technische Rolle zu konzentrieren und all das zu lernen, was es dort zu lernen gibt, statt all diese Fähigkeiten erwerben zu müssen, während man gleichzeitig auch noch Management-Dinge lernen muss.

In Ihrer Karriere wird es eventuell irgendwann einen Punkt geben, an dem Sie technischer Projektleiter werden müssen, um weiterzukommen, auch wenn Sie eigentlich gern reiner Entwickler bleiben würden und lieber eine Fachkarriere verfolgen möchten. Das heißt aber nicht, dass Sie das jetzt tun müssen. Wenn Sie das Gefühl haben, dass es für Sie im Team noch viel Technisches zu lernen gibt und Sie lieber an diesem Projekt arbeiten, während es jemand anderes leitet, dann übernehmen Sie eben nicht die Rolle des technischen Projektleiters. Wenn Sie andererseits nicht glauben, dass Ihre Entwickler-Rolle Sie weiter herausfordern wird, ist es vielleicht an der Zeit, andere Fähigkeiten zu erwerben – und die des technischen Projektleiters sind solche, die sich gut ausprobieren lassen.

Der Zeitpunkt der Entscheidung: Fachkarriere oder Managementkarriere?

Die Entscheidung, ob Sie ein Manager werden oder die Fachkarriere weiterverfolgen wollen, ist nicht leicht. Sie hängt sehr von der Gesamtsituation ab, und ich kann Ihnen wirklich nicht sagen, was Sie machen sollen. Aber als jemand, der sich beide Wege erwünscht und sie dann auch beschritten hat, weiß ich, wie ich mir diese Rollen vorgestellt habe und wie sie dann in Wirklichkeit aussahen. Mit der Warnung, dass es sich im Folgenden nur um eine skizzenhafte Darstellung handelt, die nicht in Stein gemeißelt ist, möchte ich Ihnen erzählen, wie sich Wunsch und Wirklichkeit bei mir unterschieden.

Das gewünschte Leben eines erfahrenen Entwicklers

Sie verbringen Ihre Tage mit einem Mix aus scharfsinnigem Nachdenken, dem Lösen schwieriger Probleme, die Ihren Intellekt fordern, dabei aber trotzdem Spaß machen und neu sind, und der Zusammenarbeit mit anderen scharfsinnigen Denkern. Es ist Software, also wissen Sie, dass es vielleicht auch Unsinniges zu erledigen gibt, aber auf der anderen Seite erhalten Sie einige der interessantesten Aufgaben und haben die Macht, selbst zu entscheiden, woran Sie arbeiten. Sie lieben es, Code zu schreiben, zu reparieren, schneller zu machen und Computer neue Dinge tun zu lassen – und die meiste Zeit sind Sie damit auch beschäftigt.

Aufgrund Ihrer Erfahrung fragt der Manager Sie um Rat, wie die Entwicklung angegangen werden sollte, bevor ein Projekt beginnt, daher wissen Sie über alles, was geschieht, Bescheid, müssen sich aber nicht mit den ganzen Details der anderen herumschlagen. Sie werden zu genau den richtigen Meetings eingeladen, bei denen die wichtigen Entscheidungen fallen, aber es sind auch nicht so viele, dass Ihr Flow unterbrochen wird. Die unerfahreneren Entwickler schauen zu Ihnen auf und hängen an Ihren Lippen. Sie bitten Sie um Rat, belästigen Sie aber nicht zu sehr, sodass immer noch genug Zeit zum Nachdenken bleibt.

Ihr Aufwärtstrend reißt nicht ab, und es gibt immer neue, größere Probleme, die Sie lösen und damit zeigen, wie wertvoll Sie für die Firma sind. Sie arbeiten hart, werden aber nur selten gebeten, länger zu bleiben oder am Wochenende zu kommen, denn wir wissen ja alle, dass es unmöglich ist, so viele Stunden die Woche qualitativ hochwertige Kopfarbeit zu leisten. Wenn Sie noch spät arbeiten, liegt das daran, dass Sie gerade im Flow stecken und das Feature gern abschließen oder den Fehler beheben wollen.

Sie schreiben Bücher, halten Vorträge und arbeiten an Open Source mit – und mit ein bisschen Glück und Ausdauer werden Sie in der ganzen Branche bekannt. Keinen stört es, dass Sie ein bisschen tollpatschig oder scheu sind, und niemand erwartet, dass Sie groß an Ihren Kommunikationsfähigkeiten arbeiten, denn was Sie sagen, ist wichtig. Jeder in Ihrer Firma weiß, wer Sie sind, versteht, wie wertvoll Ihre Arbeit ist, und nimmt Ihre Meinung ernst.

Kurz gesagt: Sie haben die perfekte Balance aus guter Arbeit, Ruhm und gesammelter Erfahrung, die Sie so wertvoll und respektiert, hoch bezahlt und einflussreich macht.

Das echte Leben eines erfahrenen Entwicklers

Wenn Sie das richtige Projekt und den richtigen Zeitpunkt im richtigen Projekt finden, ist Ihr Leben toll. Sie werden herausgefordert und lernen neue Dinge. Sie haben recht viel Kontrolle über Ihre tägliche Arbeit und mit Sicherheit weniger Meetings als Ihr Management-Gegenüber. Aber Ihre Tage verlaufen nicht immer in einem glückseligen Flow. In jedem Projekt gibt es einen Zeitpunkt, an dem Sie eine Idee haben und Ihre Kollegen unbedingt von ihrem Ansatz überzeugen wol-

len. Oder Sie haben das System implementiert, müssen aber jetzt die anderen Teams dazu bringen, es auch zu verwenden. Also sitzen Sie tagelang mit ihnen zusammen, zeigen Vor- und Nachteile auf, erklären, warum es nützlich ist, und versuchen, sie davon zu überzeugen, sich bei ihren Managern dafür einzusetzen, dass es übernommen wird.

Ihr Aufwärtstrend ist nicht ganz so steil und einfach, wie Sie es sich erhofft hatten. Tatsächlich ist er eher flach. Diese großen Projekte, die zeigen, was für ein unschätzbar wertvoller Architekt Sie sind, gibt es nicht sehr häufig. Das Team braucht keine neue Programmiersprache, keine neue Datenbank und auch kein neues Web-Framework. Ihr Manager hat keine besonderen Aufgaben für Sie, die der ganzen Firma Ihr Talent demonstrieren – er geht davon aus, dass *Sie* für *ihn* herausfinden, wo diese Gelegenheiten sind. Das Finden guter Projekte scheint reines Glück zu sein. Wählen Sie ein falsches aus, verbringen Sie Monate oder Jahre mit etwas, das trotz aller Bemühungen irgendwann abgebrochen wird. Auf Ihre Freunde im Management sind Sie ein wenig eifersüchtig, denn sie scheinen schneller befördert zu werden und ihre Teams wachsen.

Die anderen Entwickler sind ein gemischter Haufen. Sie selbst sind sehr nett, daher sehen einige zu Ihnen auf und hören auf Ihre Meinung, aber andere scheinen Ihnen Ihren Einfluss zu neiden. Neue Entwickler wollen entweder zu viel von Ihrer wertvollen Zeit in Anspruch nehmen, oder sie haben – warum auch immer – vor Ihnen Angst. Bezüglich der Leitung großer, interessanter Projekte gibt es zwischen Ihnen und Ihren Kollegen auf jeden Fall ein gewisses Konkurrenzdenken.

Ihre Managerin ist auch keine große Hilfe. Sie findet Ihre Idee, ein System als Open Source zu bringen, das nach Ihrer Meinung eine Verbesserung für das Logging in Ihrer Branche ermöglicht, nicht so gut. Sie schlägt vor, dass Sie doch in Ihrer Freizeit Bücher schreiben oder Vorträge halten könnten. In technischen Dingen werden Sie von ihr um Rat gefragt, aber manchmal vergisst sie, Ihnen von neuen Projekten zu erzählen, und dann ist es zu spät, sich noch einbringen zu können. Sie haben den Verdacht, dass wichtige Informationen nicht bei Ihnen ankommen, weil Sie nicht in den richtigen Meetings sitzen, aber jedes Mal, wenn Sie von ihr zu Meetings eingeladen werden, fällt Ihnen wieder ein, wie langweilig und ineffizient diese sind und wie viel wertvolle Zeit Sie verlieren. Und sie hat nicht viel Geduld mit Ihnen, wenn es um Ihren Wunsch geht, langweilige Arbeiten zu ignorieren – wie zum Beispiel das Beantworten von E-Mails, das Unterhalten mit Bewerbern oder das zügige Reagieren auf Code-Reviews.

Allerdings sind Sie den größten Teil der Zeit tatsächlich damit beschäftigt, Dinge zu bauen. Sie können sich auf technische Probleme, das Design von Systemen und Entwicklungsherausforderungen konzentrieren und müssen sich nicht allzu viel mit anderen Leuten herumschlagen oder in langweiligen Meetings sitzen. Häufig können Sie sich Ihre Projekte aussuchen und auch das Team wechseln, wenn Sie etwas Neues machen wollen. Und Sie haben gerade herausgefunden, dass Sie mehr verdienen als Ihre Managerin. So schlecht ist das Leben also gar nicht.

Das gewünschte Leben eines Managers

Sie haben ein Team, Sie haben die Kontrolle, Sie können Entscheidungen treffen und andere dazu bringen, ihre Aufgaben so zu erledigen, wie Sie es sich vorstellen. Ihr Team respektiert Sie und achtet Ihre Autorität in allen Situationen. Sie glauben, das Team sollte mehr Tests schreiben? Sie sagen: »Schreibt mehr Tests«, und sie tun es! Sie wollen sicherstellen, dass jeder unabhängig von Geschlecht, Rasse und so weiter gleichbehandelt wird? Sie achten darauf, dass das geschieht, und feuern jeden, der die Grenzen überschreitet und für eine Umgebung sorgt, in der sich der Rest des Teams nicht wohlfühlt.

Weil Sie sich um Ihre Leute kümmern, wissen diese, dass Sie immer alles versuchen, um für sie Ihr Bestes zu geben, auch wenn Sie unterschiedlicher Meinung sind. Sie gewähren Ihnen einen Vertrauensvorschuss und geben bei 1-1s offenes Feedback, wenn Sie selbst Mist gebaut haben. Gleichzeitig freuen sie sich über Feedback von Ihnen. Sicher, die Arbeit mit Menschen ist anstrengend, aber Ihre Mitarbeiter wissen, dass Sie sich um sie kümmern, sodass die Kommunikation sehr bereichernd ist. Und seit Sie die entsprechende Autorität besitzen, sehen Sie auch, wie schnell sich Ihr Coaching auswirkt.

Bemerken Sie bei einem anderen Manager, dass bei ihm etwas falsch läuft, können Sie ihm genau so Ratschläge erteilen, wie Sie anderen Entwicklern Hilfe beim Systemdesign geben. Andere Manager haben immer ein offenes Ohr für Ihre Gedanken und können sehen, wie effektiv Ihr Team arbeitet, wie gut Sie sich um den Zustand der Firma kümmern und wie ehrlich Sie daran interessiert sind, dass alles besser wird.

Ihre Managerin unterstützt Sie durch umfangreiches Coaching, schreitet aber nur selten ein, um zu sagen, was Sie zu tun haben. Sobald Sie das Gefühl haben, jetzt für ein größeres Team bereit zu sein, kümmert sie sich darum, dass Sie mehr Mitarbeiter bekommen. Ihre Ziele sind klar und ändern sich nicht ständig. Auch wenn Sie viel Verantwortung tragen, gibt es immer etwas Zeit, Blogposts zu schreiben oder Vorträge zu halten. Das wird zudem aktiv unterstützt, weil Ihr Team dadurch in der Technologiebranche einen besseren Stand bekommt und Sie leichter gute Mitarbeiter finden.

Kurz gesagt: Sie können Entscheidungen treffen, die Kultur vorgeben, und Ihre Effektivität ist für alle in Ihrer Umgebung deutlich sichtbar, sodass Sie bald aufsteigen und Ihre Karriere weiterverfolgen können.

Das echte Leben eines Managers

Sie haben ein Team. Sie haben ein gewisses Maß an Kontrolle, aber Ihnen ist auch schnell klar geworden, dass es leichter gesagt als getan ist, Mitarbeiter dazu zu bringen, etwas Bestimmtes zu tun. Über Ihre eigene tägliche Arbeit haben Sie die Kontrolle längst verloren. Die meiste Zeit verbringen Sie in Meetings. Sie wussten, dass das kommen würde, aber erst jetzt verstehen Sie, was es bedeutet. Als Sie nur

ein kleines Team hatten, konnten Sie immer noch Zeit finden, selbst ein bisschen Code zu schreiben, aber mit mehr Mitarbeitern ist Ihnen der Kontakt zum Code verloren gegangen. Sie wissen, dass es besser wäre, wenigstens noch eine kleine Verbindung dazu zu halten, aber es fehlt schlicht die Zeit. Immer wenn Sie ein paar Stunden freimachen können, um Code zu schreiben, wird Ihnen klar, dass es gegenüber dem Team unverantwortlich wäre, den jetzt einzuchecken und von Ihren Mitarbeitern warten zu lassen. Also kommt nicht mehr als ein Skript oder das Debuggen eines kleinen Problems heraus. Sich auf das Bauen eines größeren Features konzentrieren zu können, ist nur eine vage Erinnerung.

Sie treffen Entscheidungen – nun, manche Entscheidungen. Realistisch gesehen, können Sie vielleicht die Dinge eingrenzen, die entschieden werden. Sie können Ihr Team auf Themen fokussieren, wie zum Beispiel auf das Schreiben besserer Tests, aber es gibt immer noch eine Produkt-Roadmap, die umgesetzt werden muss, und Ihre Mitarbeiter haben ihre eigenen Ideen, welche technischen Aufgaben mit Priorität anzugehen sind. Statt also selbst Entscheidungen zu treffen, helfen Sie Ihrem Team dabei, Entscheidungen zu fällen. Ihr Manager gibt Ihnen Ziele vor, die sich manchmal komplett ändern – und es liegt bei Ihnen, Ihrem Team das zu erklären.

Sie geben die kulturellen Standards für das Team vor, was sowohl gut als auch schlecht ist. Gut ist, wenn es sich an Ihren guten Eigenschaften orientiert. Schlecht ist, wenn es leider auch Ihre negativen Seiten als Vorbild nutzt.

Ihr Team stimmt Ihnen nicht automatisch zu, es respektiert Sie nicht von allein – ja, es mag Sie nicht einmal unbedingt. Sie erkennen, dass für Autorität mehr als nur ein Titel notwendig ist. Sie selbst hadern damit, Ihr Team in kritischen Situationen motivieren zu müssen, wenn die Projekte ins Stolpern geraten sind oder Sie jemandem sagen müssen, dass es mit der Beförderung dieses Jahr nichts wird, eine Gehaltserhöhung leider ausfällt und auch kein Bonus in Sicht ist. Manche machen sich dann nicht die Mühe, Ihnen zu sagen, dass sie unglücklich sind – wenn sie keine Lust mehr haben, kündigen sie, bevor Ihnen klar wird, dass etwas schiefgegangen ist. Läuft es für die Firma gut, ist ausreichend Geld vorhanden, um Gehaltserhöhungen zu verteilen, und gibt es viele spannende Projekte, macht das Leben Spaß – aber wenn es stressig ist, sehen Sie, wie wenig Möglichkeiten Sie haben, um die Leute glücklich zu machen. Schlimmer ist noch, dass Sie niemanden entlassen können, ohne vorher einen aufwendigen HR-Prozess durchlaufen zu müssen! Sie sehen aber, dass Ihre Arbeit von einigen durchaus wertgeschätzt wird und sie durch Ihr Coaching glücklicher und erfolgreicher werden. Diese kleinen Glücksmomente bringen Sie dann auch durch harte Zeiten.

Andere Manager sind an Ihrem Feedback nicht interessiert. Tatsächlich empfinden sie es sogar als aufdringlich und werden unfreundlich, wenn sie das Gefühl haben, Sie mischen sich in ihre Angelegenheiten ein. Ihr eigener Manager ist nicht der Meinung, dass Sie schon für ein größeres Team bereit sind, kann aber auch nicht genau erklären, warum er das denkt – seine Coaching-Fähigkeiten sind

durchaus noch ausbaufähig. Hat er vielleicht Angst, dass Sie ihn in den Schatten stellen wollen? Er will auf jeden Fall nicht, dass Sie Ihre ganze Zeit mit Gesprächen verbringen – wenn Sie zu wenig an Ihrem Schreibtisch sitzen, nervt ihn das. Es ist schwieriger als gedacht, eine Führungsfigur zu sein und gute Politik zu machen, ohne Ihre Managementkollegen oder Ihren Chef gegen sich aufzubringen. Aber wenn Sie dieses größere Team bekommen können, wissen Sie, dass Sie auch die Beförderung dazu erhalten – zumindest ist also Ihr Weg klar. Als Sie erkannt hatten, dass der Entwickler, der für Sie arbeitet, mehr verdient als Sie, wollten Sie fast schon aufgeben. Überlegen Sie sich also, wie Sie dieses größere Team schnell bekommen können. Denn warum sollte man sich sonst diesen ganzen Stress und Ärger antun?

Mein abschließender Rat ist, daran zu denken, dass Sie die Karriererichtung wechseln können, wenn Sie das möchten. Es ist nicht ungewöhnlich, dass Leute die Managementschiene ausprobieren, feststellen, dass es nichts für sie ist, und dann zurück in die Fachkarriere wechseln. Die Wahl gilt nicht lebenslang, aber bringen Sie auf jeden Fall ausreichend Neugier mit. Jede Rolle hat ihre Vor- und Nachteile, und es liegt bei Ihnen, herauszufinden, was Ihnen am meisten Spaß macht.

Guter Manager – böser Manager: der Prozess-Zar

Der Prozess-Zar glaubt, dass es den einen wahren Prozess gibt, der, korrekt implementiert und umgesetzt, alle großen Probleme des Teams lösen wird. Prozess-Zaren können von Agile besessen sein, von Kanban, Scrum, Lean oder sogar von Wasserfall-Methoden. Es ist möglich, dass sie eine ziemlich genaue Vorstellung davon haben, wie der Bereitschaftsdienst zu organisieren ist, wie Code-Reviews durchzuführen sind oder wie der Release-Prozess ablaufen soll. Sie tendieren dazu, sehr organisiert und detailfixiert zu sein, sie kennen die Regeln genau und folgen ihnen aufs Wort.

Prozess-Zaren finden sich häufig in QA, im Helpdesk oder in Produktmanagementgruppen. Auch in Beratungsfirmen und an anderen Orten, an denen das Messen spezifischer Arbeitsfortschritte belohnt wird, gibt es sie. Sie können sich auf das operative Geschäft fokussieren, aber es gibt meiner Erfahrung nach nur sehr wenige dieser Menschen in klassischen System-Operations-Teams. In einem Projektmanagementteam können sie sich als unverzichtbar erweisen, weil sie sicherstellen, dass keine Aufgabe vergessen wird und alles so verpackt ist, wie es sein soll.

Prozess-Zaren kommen in Schwierigkeiten, wenn ihnen nicht klar wird, dass die meisten Menschen Prozessen nicht so gut folgen können wie sie selbst. Dann tendieren sie dazu, alle Probleme darauf zurückzuführen, dass dem besten Prozess nicht gefolgt wurde, statt sich einzugestehen, dass manchmal auch Flexibilität notwendig ist und sich unerwartete Änderungen ergeben können. Häufig konzentrieren sie sich auf leicht zu messende Werte, wie zum Beispiel die Anwesenheitszeit im Büro, und verpassen dabei die wichtigen Details.

Entwickler, die an das »richtige Werkzeug für den Job« glauben, verwandeln sich manchmal in Prozess-Zaren, wenn sie zu technischen Projektleitern werden und in dieser Position das eine Tool suchen, das alle Probleme rund um Planung, Fokus, Zeitmanagement und Priorisierung abdeckt. Sie lassen alle Arbeiten stoppen, während sie den perfekten Prozess finden wollen, oder beglücken ihr Team fortlaufend mit neuen Werkzeugen und Prozessen, um das vermeintliche Chaos zu beseitigen, das zwischenmenschliche Interaktion erzeugt.

Das Gegenstück zum Prozess-Zar ist nicht der Manager, der Prozesse völlig in den Wind schlägt, sondern jemand, der versteht, dass Prozesse die Anforderungen des Teams und der Arbeit erfüllen müssen. Während »Agil« häufig sehr rigide implementiert wird, sind die Prinzipien des Agilen Manifests² ironischerweise eine wunderbare Zusammenfassung sinnvoller Prozessführung:

- Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge.
- Funktionierende Software ist wichtiger als umfassende Dokumentation.
- Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlung.
- Reagieren auf Veränderung ist wichtiger als das Befolgen eines Plans.

Als neuer technischer Projektleiter sollten Sie sich nicht darauf verlassen, dass Prozesse Probleme lösen, die eigentlich ein Resultat von Kommunikations- oder Führungsproblemen in Ihrem Team sind. Manchmal ist eine Prozessänderung hilfreich, aber sie ist kein Allheilmittel, und es gibt keine zwei guten Teams, die in Bezug auf Prozesse, Werkzeuge und Arbeitsweisen gleich sind. Mein zweiter Rat ist, nach selbstregulierenden Prozessen Ausschau zu halten. Finden Sie sich in der Rolle des Task-Masters wieder – und kritisieren dabei Personen, die die Regeln brechen oder nicht dem Prozess folgen –, prüfen Sie, ob der Prozess selbst geändert werden kann, damit man ihm leichter folgen kann. Es ist Zeitverschwendung, den Regelpolizisten zu spielen; durch Automation können die Regeln häufig deutlicher hervortreten.

Als Manager eines Prozess-Zaren helfen Sie dieser Person am besten dabei, besser mit Unklarheiten umzugehen. Wie bei vielen anderen Manager-Fallstricken kann eine Prozess-Besessenheit auch mit Versagensangst oder dem Wunsch zusammenhängen, alles zu kontrollieren, um Unerwartetes zu vermeiden. Wenn Sie ehrlich sind und deutlich machen, dass es in Ordnung ist, Fehler zu machen und nicht perfekt zu sein, reicht das häufig schon aus, Ihren Prozess-Zaren ein wenig entspannen zu lassen und dann und wann Unklarheiten zuzulassen. Es ist sehr wichtig, Prozess-Zaren davon abzuhalten, ihre gesamte Zeit mit der Suche nach dem perfekten Werkzeug oder dem perfekten Prozess zu verbringen, und ebenso entscheidend, sicherzustellen, dass sie ihre Teams nicht dafür bestrafen, dass diese nicht dem Prozess folgen.

2 <http://agilemanifesto.org/iso/de/manifesto.html>

Was einen guten technischen Projektleiter ausmacht

Gute technische Projektleiter besitzen eine Reihe von Eigenschaften, aber die folgenden sind die wichtigsten.

Verstehen Sie die Architektur

Wechseln Sie in die Rolle eines technischen Projektleiters und haben nicht das Gefühl, die Architektur Ihres Produkts vollständig zu verstehen, dann nehmen Sie sich die Zeit dafür. Lernen Sie sie kennen. Eignen Sie sich ein Gespür dafür an. Visualisieren Sie sie. Verstehen Sie ihre Verbindungen, die Orte, an denen die Daten abgelegt sind, und wie diese zwischen den Systemen fließen. Verstehen Sie, wie sich die unterstützten Produkte darin widerspiegeln und wo sich die zentrale Logik dafür findet. Es ist nahezu unmöglich, Projekte gut zu leiten, wenn Sie die Architektur, mit der Sie arbeiten, nicht verstehen.

Seien Sie ein Teamplayer

Sofern Sie alle interessanten Aufgaben selbst erledigen, hören Sie damit auf. Schauen Sie sich die kniffligen, langweiligen oder nervigen technischen Bereiche an und prüfen Sie, ob Sie dort etwas erreichen. Die Arbeit an den weniger spannenden Bereichen der Codebasis kann Ihnen besser Aufschluss darüber geben, wo der Prozess defekt ist. Bei langweiligen oder frustrierenden Projekten gibt es häufig ein offensichtliches Problem, das sich erkennen und beheben lässt, wenn sich jemand mit Erfahrung die Zeit nimmt und sich das Ganze anschaut. Erledigen Sie nur die langweilige Arbeit, hören Sie damit aber auch auf. Sie sind ein erfahrener Entwickler, der in diesem Bereich viele Talente besitzt, und es ist durchaus verständlich, dass Sie sich einigen kniffligeren Aufgaben widmen. Sie sollten auch andere in Ihrem Team darin unterstützen, das gesamte System kennenzulernen, und ihnen die Möglichkeit geben, besser zu werden, aber Sie müssen sich dafür in Bezug auf eigene Aufgaben nicht vollständig aufopfern. Schnappen Sie sich dann und wann eine interessante Aufgabe, solange Sie ausreichend Zeit dafür aufbringen können.

Treiben Sie technische Entscheidungen voran

Sie werden an den meisten großen technischen Entscheidungen für Ihr Team beteiligt sein. »Beteiligt« heißt aber nicht, dass Sie sie allein treffen. Wenn nur Sie entscheiden, ohne sich die Meinung Ihres Teams anzuhören, wird es Ihnen die Schuld geben, wenn etwas schiefgeht. Treffen Sie andererseits gar keine technischen Entscheidungen und überlassen alles dem Team, kann es sein, dass sich Dinge hinschleppen, die schnell hätten entschieden werden können.

Finden Sie heraus, welche Entscheidungen von Ihnen getroffen werden müssen, welche von anderen mit mehr Wissen zu fällen sind und bei welchen das ganze Team beteiligt sein muss. In allen Fällen müssen Sie deutlich machen, worüber zu diskutieren ist, und das Ergebnis bekannt machen.

Kommunizieren Sie

Ihre Produktivität ist jetzt weniger wichtig als die Produktivität des gesamten Teams. Häufig bedeutet dies, dass Sie den Preis für den Kommunikations-Overhead zahlen müssen. Statt jedes Teammitglied in Meetings zu stecken, repräsentieren Sie das Team, kommunizieren seine Bedürfnisse und übermitteln Informationen aus diesem Meeting zurück ins Team. Wenn es ein universelles Talent gibt, das erfolgreiche Führungspersönlichkeiten vom Rest der Herde unterscheidet, sind es die Kommunikationsfähigkeiten. Erfolgreiche Führungskräfte schreiben gut, lesen sorgfältig und können vor einer Gruppe sprechen. Sie sind in Meetings aufmerksam und erweitern kontinuierlich ihre eigenen Wissensgrenzen sowie die des Teams. Jetzt ist der richtige Zeitpunkt, an Ihren Schreib- und Sprechfähigkeiten zu arbeiten. Schreiben Sie Designdokumente und holen Sie sich Feedback dazu von besseren Schreibern. Sprechen Sie in Teammeetings, bei Meet-ups und sammeln Sie ganz allgemein Erfahrung mit dem Sprechen vor einer größeren Gruppe.

Vergessen Sie nicht, während Sie kommunizieren, auch zuzuhören. Geben Sie anderen die Chance, zu sprechen, und achten Sie darauf, was sie sagen. Trainieren Sie, Aussagen der anderen zu wiederholen, um sicherzustellen, dass Sie sie richtig verstanden haben. Lernen Sie, zu hören, was jemand sagt, und es mit Ihren eigenen Worten erneut wiederzugeben. Wenn Sie nicht gut darin sind, Notizen zu machen, dann üben Sie das. Es ist egal, ob Sie sich dazu entscheiden, ganz tief in die Technologie einzutauchen oder ein Manager zu werden – wenn Sie nicht kommunizieren und anderen zuhören können, werden Sie in Ihrer Karriere nicht sehr weit kommen.

Bewerten Sie Ihre eigene Situation

- Gibt es in Ihrer Firma technische Projektleiter? Findet sich eine schriftliche Jobdokumentation für diese Rolle? Wenn ja: Was steht drin? Wenn nein: Wie würden Sie die Rolle in Ihrer Firma definieren? Wie würde ein technischer Projektleiter die Rolle definieren?
- Wenn Sie sich entscheiden, ein technischer Projektleiter zu werden – sind Sie dazu bereit, Ihren eigenen Horizont zu erweitern? Ist es in Ordnung, dass Sie Aufgaben außerhalb vom Coden haben? Fühlen Sie sich in der Codebasis so weit zu Hause, dass Sie andere anleiten können, die darin tätig sind?
- Haben Sie Ihren Manager gefragt, was er vom technischen Projektleiter erwartet?
- Wer ist der beste technische Projektleiter, mit dem Sie je zusammengearbeitet haben? Was war es, dass diese Person so gut sein ließ?
- Haben Sie schon mit einem frustrierenden technischen Projektleiter gearbeitet? Was hat er getan, dass es so frustrierend war?