

„Niemand schaut mir über die Schulter, und trotzdem machen alle das Richtige“ – Führung in verteilten Teams

Steven Schwenke, msg DAVID

Nachdem in den letzten beiden Ausgaben grundsätzliche Prinzipien des Remote-Arbeitens erläutert und Erfolgskonzepte für das Moderieren verteilter Meetings gezeigt wurden, beleuchtet dieser vorerst letzte Artikel der Serie die Führung in verteilten Teams.

Die wichtige Nachricht gleich vorneweg: „Führung“ ist im Rahmen dieses Artikels als die Summe aller Maßnahmen zu verstehen, ein verteiltes Entwicklungsteam zu einem gemeinsamen Erfolg zu füh-

ren. Dabei wird Wert darauf gelegt, sämtliche Aufgaben als eine organisatorische Einheit zu lösen, anstatt Aufgabenpakete „über den Zaun zu Nearshore“ zu werfen.

Hintergrund dieses alle Maßnahmen bestimmenden Gedankens ist der Entstehungsprozess des Projekts, in dem der Autor als Tech Lead für die technische Qualität und den Entwicklungsprozess verantwortlich ist. Als das Projekt auf der grünen Wiese gestartet wurde, war der Autor einer der beiden Entwickler, die von einem erfahrenen Architekten geführt wurden. Aufgrund der Größe und insbesondere der Firmenkultur wurde sehr viel Wert auf das gemeinsame Lernen und das nachhaltige Miteinander-Arbeiten ge-

legt. Sämtliche Fachlichkeit wurde von Anfang an allen Teammitgliedern im Detail erklärt und auch wichtige Entscheidungen, wie zum Beispiel das Schneiden von Aufgaben, wurden offen im Team diskutiert und entschieden.

Nach einiger Zeit wuchs das Team auf insgesamt sechs Kollegen an und der Autor unterstützte die Entwicklung ein halbes Jahr als Scrum-Master, um dann in die Rolle des Tech Lead zu wechseln. Aufgrund des bereits langen gemeinsamen Arbeitens auf Augenhöhe wurde die Tradition der größtmöglichen Transparenz und Offenheit im Team weitergeführt. Dies führte zu einer natürlichen und von allen respektierten Rollenverteilung, die zusammen mit ständiger Transparenz in sämtlichen Entscheidungsprozessen zum Projekterfolg beitrug.

Leading by Example

Wer kennt sie nicht, die sonore und scheinbar über allen potenziellen Problemen stehende Pilotenstimme aus dem Cockpit. Wenige kennen den Ursprung dieser Eigenart vieler Berufspiloten. Chuck Yeager war der erste Pilot, der am 14. Oktober 1947 mit der „X-1“ die Schallmauer durchbrach (siehe *Abbildung 1*). Aufgrund seiner Herkunft aus einer ländlichen Gegend in West Virginia und des damals unter Testpiloten verbreiteten Überlegenheits-Denkens [1] entwickelte er eine besondere Sprechweise, die ihn gegen alle Arten von Problemen komplett immun erscheinen ließ. Spätestens mit seinem erfolgreichen Überschallflug galt er als Ikone der Fliegerei und selbst viele Berufspiloten ahmten diese distanzierte Sprechweise nach.

Auch wenn die Herkunft dieses besonderen Dialekts nichts mit Problemen moderner Software-Entwicklung zu tun hat, so hilft eine ruhige Stimmlage in kritischen Situationen aller Art sehr. Eine gute Führungspersönlichkeit nimmt sich selbst dann die Zeit für ein paar ruhige Atemzüge, wenn der Produktiv-Server das Log mit Fehlermeldungen flutet. Sie spricht ruhig, unaufgeregt und faktiziert und stellt mehr Fragen, als Anweisungen zu geben. Doch nicht nur in Ausnahmesituationen ist Führung wichtig. Stets zu Terminen pünktlich zu sein und Team-Mitglieder zu fragen, ob sie gerade Zeit haben, statt einfach Aufmerksamkeit zu nehmen, zeugt von gegenseitigem Respekt und erhöht mittelfristig die Produktivität aller Beteiligten.

All dies sind Beispiele für „Leading by Example“. Sehr anschaulich ist dies in der bekannten Fernsehserie „Star Trek: The Next Generation“ dargestellt. In kritischen Situationen versammeln sich alle Brückenoffiziere in einem Besprechungsraum, um dem Captain Handlungsoptionen aufzuzeigen. Dieser hört (meist) ruhig zu und stellt Verständnisfragen. Erst ganz zum Schluss, wenn alle Informationen allen bekannt sind, trifft er eine Entscheidung und verteilt eindeutige Aufgaben.

Trust Equation

Neben dem gegenseitigen Respekt ist Vertrauen eine der wichtigsten Komponenten eines gut funktionierenden Teams. Dieses kann im Wesentlichen durch drei Faktoren vergrößert werden:

1. Glaubwürdigkeit
2. Verlässlichkeit
3. Vertrautheit



Abbildung 1: Chuck Yeagers X-1

Glaubwürdigkeit ist in der Theorie einfach zu erreichen, nämlich indem man genau das vorlebt, was man sagt. Verlässlichkeit in Teams ist das Verhältnis von Versprochenem zu Eingehaltenem. Dieser Aspekt lässt sich insbesondere bei der Übertragung von Aufgaben und Verantwortung sicherstellen. In dieser Situation ist die explizite und unmissverständliche Kommunikation von Erwartungshaltungen und Rahmenbedingungen wichtig. Nach den Erläuterungen hilft die einfache Frage „Fühlst du dich mit dieser neuen Aufgabe wohl?“, da mit ihr nicht nur die Erfolgswahrscheinlichkeit erhöht werden kann, sondern auch die Zuversicht auf ein gutes Ergebnis.

Vertrautheit-Schaffen ist eine sehr individuelle Angelegenheit. Das Einbringen von Persönlichem zusätzlich zur Arbeitsebene ist wohl der einfachste Weg. Über den eigenen Urlaub, die eigenen Hobbies oder sogar die Familie zu reden, lockert in den richtigen Situationen nicht nur auf, sondern schafft auch Verbundenheit auf einer ganz menschlichen Ebene. Dies ist im Umfeld des verteilten Arbeitens wichtig, da der sogenannte „Watercooler-Effect“ (im Deutschen am besten „Flurfunk“) fehlt und man sich neben den geschäftlichen Meetings oft wenig Zeit für ein einfaches Gespräch nimmt.

Glaubwürdigkeit, Verlässlichkeit und Vertrautheit steigern, wie beschrieben, das Vertrauen in einem Team. Laut der Trust Equation gibt es nur einen Aspekt, der diesem positiven Effekt entgegenwirkt: Selbstorientierung oder Egoismus. Stellt man das Team voran und achtet darauf, dass sich jeder im Team abgeholt, eingebunden und nicht hintergangen fühlt, stellt sich eine vertraute Atmosphäre wie von selbst ein. Diese Betrachtung ist bis hierhin natürlich reichlich theoretisch und trocken, deshalb folgen ein paar Beispiele.

Der Urlaubsantrag eines rumänischen Teammitglieds für nur einen Vormittag hat uns in Deutschland verwundert; normalerweise nimmt man sich ja immer einen ganzen Tag frei. Auf Nachfrage stellte sich heraus, dass eine Motorrad-Führerscheinprüfung abgenommen wurde. Aus diesem kleinen Detail ergaben sich mehrere Gespräche über Motorräder, schöne Strecken in Rumänien und Unterschiede im allgemeinen Straßenverkehr. Außerdem wissen nun



Delete!
Delegate.
Do :(

Abbildung 2: „3Ds“ aus „Getting Things done“

alle im Team, dass die rumänische Entwicklerin und der Projektleiter ein gemeinsames Hobby haben.

Wie in vergangenen Artikeln erwähnt, arbeitet der Autor viel von Zuhause aus. Dadurch ergeben sich fast schon zwangsmäßig Einblicke in das Privatleben. Angefangen beim Bücherregal, das bei jeder Skye-Session im Hintergrund zu sehen ist und von einigen Kollegen bei Gelegenheit auf Veränderungen geprüft wird, bis hin zur über den Schreibtisch laufenden Katze. Letztere ist bestens geeignet, um angestrengte Fachgespräche aufzulockern.

Als letztes Beispiel soll die offene und direkte Kommunikation im Team des Autors dienen. Es hat sich als extrem vertrauensbildend herausgestellt, möglichst viel Drumherum in alle Richtungen des Teams zu kommunizieren. Organisatorische Details, Berichte darüber, was man so für andere Baustellen außerhalb des Teams getan hat, das Aufklären von Gerüchten oder einfach nur interessante Begebenheiten aus anderen Bereichen tragen nicht direkt zum Projekt-Fortschritt bei, schaffen aber Nähe. Mit einer solchen stabilen und über lange Zeit erarbeiteten Vertrauensgrundlage werden potenziell negative Ereignisse viel entspannter aufgenommen: „Wenn sich wirklich etwas Schlechtes für uns ergeben sollte, werden wir wie immer zeitnah und direkt informiert. Wir brauchen uns jetzt noch keine Gedanken zu machen.“

Delegation

Gekonntes Delegieren erleichtert das (Projekt-)Leben ungemein. Schon im Selbstmanagement-Klassiker „Getting Things done“ (GTD) [2] wird empfohlen, eintreffende Aufgaben am besten gleich wieder zu löschen, wenn es denn geht. Falls das nicht der Fall ist, sollte man sie (richtig) delegieren (siehe Abbildung 2). Nur im Notfall ist eigenes Handeln angesagt. An dieser Stelle ist es wichtig zu unterstreichen, dass dieses Prinzip nicht zur Handlungsunfähigkeit und zum „Nur Weiterleiten“ aufruft. Es geht hier um den bestmöglichen Einsatz der einem zur Verfügung stehenden Zeit. Ziel ist es, dass jeder das tut, was er am besten kann.

Wenn man dieses Prinzip verfolgt, stellt sich natürlich die Frage: Wie delegiert man nun am besten? In verteilten Teams ist dieses Thema noch viel wichtiger, als bei an einem Ort arbeitenden Kollegen. Bei

falscher Delegation besteht nicht nur die Gefahr eines falschen oder ungewünschten Ergebnisses. Schlimmer: Dieses wenig brauchbare Ergebnis wird erst nach einigen Tagen bekannt, wenn man sich nicht regelmäßig über den Fortschritt austauscht. So kann sehr effektiv und massiv Zeit verbraucht werden. Einem verteilten Team das richtige Delegieren beizubringen, ist also eine wichtige Investition in die Arbeitsfähigkeit des Teams. Außerdem wird so die in diesem Setting oft drohende „Micro-Management-Trap“ vermieden, in der jeder kleinste Schritt erklärt und überprüft wird, statt Verantwortlichkeiten zu übertragen und nur das Endergebnis zu prüfen.

Ein wichtiger Aspekt beim Delegieren ist die Deadline. Nach dem Parkinson'schen Gesetz [3] nimmt jede Aufgabe so viel Zeit in Anspruch, wie man ihr einräumt. Wird also keine Deadline gesetzt, werden Aufgaben potenziell nie fertig. Deshalb ist bei jeder delegierten Aufgabe immer eine Deadline zu setzen. Diese hat nicht die Aufgabe, unnötig sportlich anzutreiben. Wenn man als Delegierender weiß, dass die Aufgabe vermutlich fünf Tage dauert, hat es keinen Sinn, nur drei Tage Zeit zu gewähren.

Das Ziel einer Deadline ist nicht die Optimierung, sondern die Ermöglichung der Selbstorganisation bei dem ausführenden Teammitglied. Die Angabe des Zeitpunkts der Deadline muss so exakt und genau wie möglich erfolgen. Die Auskünfte „in den nächsten Tagen“ oder „im Winter 2018“ sind nicht nur ungenau, sondern auch fehleranfällig: Ist der Winter der Zeitpunkt, an dem es das erste Mal schneit oder ganz exakt der 21. Dezember? Besser ist folgender Satz: „Ich benötige den Text am Freitag, den 9. Juni um 15 Uhr deutscher Zeit. Dann kann ich ihn über das Wochenende lesen, um ihn am Montag bei meinem Termin mit dem Kunden in den Vertrag aufzunehmen.“ Schon an der Länge erkennt man, dass hier viel mehr Informationen übermittelt werden.

Sicher ist es nicht unbedingt notwendig zu wissen, was mit dem textuellen Ergebnis der übertragenen Aufgabe passiert. Ein vertragsrelevanter Text, der eins zu eins so dem Kunden übergeben wird, erzeugt jedoch sicherlich ein anderes Gefühl beim Schreiber als eine unwichtigere Aufgabe. Die Information, dass der Delegierende sich wirklich auf das Eintreffen des Textes bis zur angegebenen Zeit verlassen können muss, da ansonsten der ganze Zeitplan durcheinanderkommt, bietet Einblick in den viel größeren Kontext und ermöglicht damit mehr Verständnis. Der Zeitpunkt der Deadline ist jedenfalls absolut unmissverständlich – selbst über Zeit-zonen hinweg.

Das hervorragende Buch „Management 3.0“ von Jurgen Appelo [4] beinhaltet eine ganze Checkliste von Aspekten guter Delegation. Angefangen beim Risiko-Faktor, nämlich der Problematik, ob das Team-Mitglied eine Aufgabe aufgrund seiner Fähigkeiten und Kenntnisse überhaupt sinnvoll bearbeiten kann, bis hin zur Frage „Weiß der Kollege, wie Fortschritt an der Aufgabe aussieht?“ werden viele unterschiedliche Aspekte abgedeckt.

Als letzter, aber absolut nicht unwichtiger Punkt zum Thema „Delegation“ sei das Verhalten im Fehlerfall genannt. Wenn Delegation schief läuft, und das wird früher oder später garantiert der Fall sein, stellt sich manchmal eine ernüchterte Stimmung ein. „Warum haben die Kollegen die Aufgabe nicht verstanden?“ „Was war daran so schwer?“ – diese Fragen kommen schnell, helfen jedoch nicht. Die

einzigste Frage, die wirklich hilft, ist: „Was kann ich als Delegierender beim nächsten Mal besser machen, damit es funktioniert?“

Der Zuschauer-Effekt

Einer der ungünstigeren Zustände, in die ein verteiltes Team fallen kann, hat mit dem Zuschauer-Effekt [5] zu tun. Dieser beschreibt das Verhalten größerer Menschenansammlungen bei Notfällen oder in besonderen Situationen. Je größer die Menge, desto geringer die Chance, dass helfend eingegriffen wird. „Es sind so viele Menschen hier, irgendeiner wird schon helfen“, wird oft gedacht. Blöd nur, wenn das alle denken.

Insbesondere bei Retrospektiven, wenn über Probleme und Hindernisse geredet wird, kann dieser Effekt auch ein verteiltes Team treffen. Das beispielhaft genannte Problem schlechter User-Stories ist von den Entwicklern in der Retro genannt und pflichtbewusst vom Protokollanten niedergeschrieben wurden. Dann schweigen alle, auch die Entwickler und sogar der im Team vorhandene Anforderungsmanager, der die Stories geschrieben hat. Warum sagt keiner etwas?

Erst nach einiger Moderationsarbeit kommt heraus, dass sich tatsächlich jeder zwar als Zuschauer und irgendwie Leidtragender, aber eben nicht als Verantwortlicher sieht: Die Entwickler haben den Punkt ja aufgebracht – der Anforderungsmanager muss besser schreiben, klarer Fall. Dieser ist mit seinen Stories allerdings zufried-

den und muss sein Verhalten nicht ändern. Jetzt, wo die Entwickler grundsätzlich willkommene Vorschläge für die Stories haben, werden sie sich ja schon mit Fragen melden.

Der Tech-Lead denkt sich, dass er damit ja sowieso nicht viel zu tun hat. So blicken alle Anwesenden auf das eben notierte Problem und jeder denkt vom anderen, er würde handeln. Doch es passiert nichts. Deshalb muss der Moderator (oder Scrum-Master, je nach Rollenverteilung) dafür sorgen, dass konkrete und korrekt delegierte Aufgaben mit dem Team entwickelt und dann ebenfalls ins Protokoll aufgenommen werden. Das Niederschreiben schafft Verbindlichkeit. Das Nennen genau eines Verantwortlichen schafft Ansprechbarkeit. Das Verwenden direkter Sprache, wie „Steven, wir wollen, dass du das eben beschriebene Problem bis Mittwoch analysierst und uns bis spätestens 15 Uhr eine Mail mit den Ergebnissen schickst“, schafft Klarheit über die nächsten Schritte.

Genau diese nächsten Schritte werden oft vergessen: Eine sehr einfache Technik zur Vermeidung des Zuschauereffekts ist die Frage „Was ist der nächste Schritt?“, die man immer dann stellen sollte, wenn eine Aufgabe scheinbar in der Luft hängt und droht, vom nächsten Tagesordnungspunkt überholt zu werden.

Regelmäßige One-on-Ones

Als „One-on-Ones“ werden regelmäßig stattfindende Treffen zwischen zwei Team-Mitgliedern bezeichnet. Der Autor führt solche

- Anzeige -

Software Engineer (m/w) Java/Scala

für den Standort Stuttgart

Fashion und Lifestyle, 5.500 Mitarbeiter, 11 Department Stores, 1 E-Shop, 1000 Marken, 15 Restaurants & Confiseries, 15 erstklassige Services und stets ein besonderes Einkaufserlebnis – das ist Breuninger.

Fashion and IT? OF COURSE - Let's talk innovation!

Uns hast du hier nicht erwartet oder? Aber mal ehrlich: Mode und IT sind Welten die man selten miteinander verbindet, aber hier bei Breuninger tun wir genau das! Während wir unsere neue wegweisende Multichannel Plattform von Grund auf neu entwickeln, arbeiten wir an unserer IOS und Android App, Beacon Technologien, Data Science Projekten und einer Menge mehr! Nach dem Ansatz der Self Contained Systems arbeiten sechs Teams befreit von Fesseln, innovativ und ergebnisorientiert am Shop der Zukunft.

Wir sind immer auf der Suche nach DIR, denn wir haben in unserer vertikalen Organisation viele interessante Aufgaben – und das nicht nur für DevOps, App- und Software-Engineers, Product Manager und -Owner! Du willst etwas bewegen? Schau doch einfach auf Team.Breuninger.de bzw. in unseren Stellenanzeigen vorbei oder ruf direkt Sebastian unter der 0172/3070386 an und lass uns über DICH sprechen – **Wir freuen uns auf dich!**



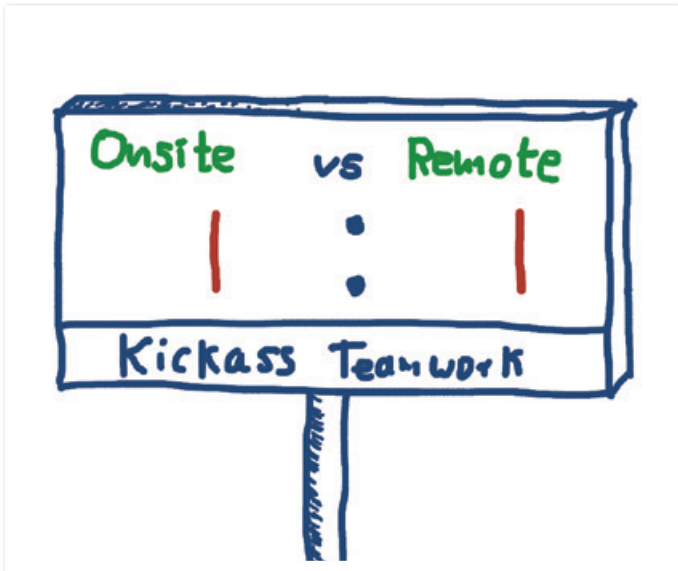


Abbildung 3: Regelmäßige One-on-Ones sind nicht nur ein Gewinn für das Team-Mitglied, sondern für das ganze Team.

Treffen mit jedem Mitglied seines Teams mindestens alle sechs Monate durch. Ziel ist, in einer entspannten und vertrauten Atmosphäre darüber zu reden, was dem Teammitglied für seine tägliche Arbeit wichtig ist, und daraus konkrete Maßnahmen abzuleiten (siehe Abbildung 3).

Einführen kann man solche Gespräche immer sehr gut mit einer Wiederholung, wie die Welt beim letzten Gespräch aussah. Was gab es für Herausforderungen, was lief damals gut und was nicht? Beim ersten One-on-One kann man auf dem technischen Background und der bisherigen Werdegeschichte aufbauen.

Anschließend sollten Parallelen und Unterschiede zur aktuellen Situation identifiziert werden. Welche Probleme gibt es noch immer? Was hat sich wesentlich verbessert? Was wurde seit dem letzten Treffen gelernt und wie kann das neue Wissen nun angewendet werden?

Der dritte Teil betrifft die Zukunft bis zum nächsten Treffen und sollte zur Formulierung klar zugeordneter Aufgaben führen. Dazu ein Beispiel aus dem Team des Autors: Das zweite One-on-One mit einem im Team relativ neuen Entwickler wurde in der ersten Phase genutzt, um die Einarbeitung ins Projekt zu besprechen. Der Entwickler hat sich im Team gut integriert und eine Nische gefunden, in der er sich gut einbringen konnte. In der zweiten Phase wurde festgestellt, dass mit den ersten Erfolgen und einem sicheren Umgang mit dem Projekt eine Weiterentwicklung beginnen sollte.

Im dritten Teil wurde die Möglichkeit besprochen, in eine Architekten-nahe Richtung zu gehen und das Projekt zum Beispiel durch die Untersuchung geeigneter neuer Bibliotheken oder Tools zu ergänzen. Als ganz konkrete Aufgabe hat der Autor sich verpflichtet, dem Kollegen Einblick in seine Architektur-Aufgaben-Liste zu geben und erste, kleinere Aufgaben daraus zu übertragen. Diese sollten als Sprungbrett in kompliziertere Aufgabenbereiche dienen. Zum Thema One-on-Ones ist der Artikel „Stop wasting my Time with your shitty One-on-One Meetings“ von Mike Acton empfohlen [6].

Fazit

Wie ganz am Anfang erwähnt, ist das Wort „Führung“ nicht unbedingt passend für die in diesem Artikel beschriebenen Maßnahmen. Ein erfolgreiches Team ist selbstorganisierend und findet eigene Wege, Tools und Prozesse, um die bei verteilten Teams vorhandene geographische Entfernung abzubauen und so erfolgreich zu sein. Einer der wichtigsten Aspekte dabei ist, wie so oft, die gelungene Kommunikation. Angefangen bei einer auch in Stresssituationen ruhigen Stimme, die faktiziert spricht und nach einem gemeinsamen Findungsprozess Aufgaben korrekt delegiert, über langfristig angelegten Vertrauensaufbau bis hin zu regelmäßigen One-on-One-Meetings ist der Weg zur erfolgreichen Führung lang und teilweise mühselig. Doch es lohnt sich. Ein derart empowertes Team wird Schwierigkeiten wie von selbst souverän meistern und braucht langfristig wesentlich weniger Management-Aufmerksamkeit.

Der Autor jedenfalls ist immens stolz auf die Kollegen, mit denen er die in dieser Artikelserie beschriebenen Erkenntnisse gemeinsam erarbeitet hat und mit denen er sowohl durch gute als auch durch etwas mühseligere Projekt-Situationen ging und dabei stets etwas gelernt hat. Vielleicht tragen die hier beschriebenen Konzepte zum Erfolg anderer Teams bei. Damit diese auch über Aktualisierungen des Themas „Remote Working“ informiert bleiben, sei noch einmal auf das Repository der msg DAVID verwiesen, in dem all das hier Geschriebene und einiges mehr frei erhältlich ist [7].

- [1] Siehe „The right Stuff“, Tom Wolfe
- [2] <https://gettingthingsdone.com>
- [3] https://de.wikipedia.org/wiki/Parkinsonsche_Gesetze
- [4] <https://management30.com>
- [5] <https://de.wikipedia.org/wiki/Zuschauereffekt>
- [6] https://medium.com/@mike_acton/stop-wasting-my-time-with-your-shitty-one-on-one-meetings-258b5c5c10cc
- [7] <https://github.com/msg-DAVID-GmbH/RemoteWorking>



Steven Schwenke

steven.schwenke@msg-david.de

Steven Schwenke ist Software Craftsman und liebt, was er tut. Als Technical Teamlead leitet er ein verteiltes Team und führt regelmäßig Inhouse-Workshops durch. Zusätzlich organisiert er Veranstaltungen wie den HackTalk und hält Vorträge bei Konferenzen. Er teilt seine Erfahrung gern mit anderen und freut sich auf spannende Gespräche.