



Heinz Hellerer

Soft Skills für Softwaretester und Testmanager

Kommunikation im Team, Teamführung,
Stress- und Konfliktmanagement

dpunkt.verlag



Dr. Heinz Hellerer ist freiberuflicher Softwaretester, Test- und Qualitätsmanager und Berater für Softwarequalität. Ehemals als Programmierer tätig (C und Java) gehören heute zu seinen beruflichen Schwerpunkten der Test von Anwendungssoftware und Webanwendungen sowie die Analyse von Testumgebungen und Testprozessen. Er leitet Testteams und berät seine Kunden in Fragen zur Sicherstellung und Verbesserung der Softwarequalität. Branchenschwerpunkte sind die Versicherungswirtschaft, Automotive, Telekommunikation und Logistik.

Heinz Hellerer

Soft Skills für Softwaretester und Testmanager

**Kommunikation im Team, Teamführung,
Stress- und Konfliktmanagement**



dpunkt.verlag

Dr. Heinz Hellerer
qualitaet@hellerer-it.de
www.hellerer-it.de

Lektorat: Christa Preisendanz
Copy Editing: Alexander Reischert, Redaktionsbüro ALUAN, www.aluan.de
Herstellung: Nadine Thiele
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:
Buch 978-3-89864-831-8
PDF 978-3-86491-208-5
ePub 978-3-86491-209-2

1. Auflage 2013
Copyright © 2013 [dpunkt.verlag](http://www.dpunkt-verlag.de) GmbH
Ringstraße 19B
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Vorwort

*The goal of a software engineer is to retire
without having caused any major catastrophe.*

Dilbert

Soft Skills sind der entscheidende Faktor für den Erfolg oder Misserfolg von Projekten. Umfragen in Unternehmen sowie zahlreiche Untersuchungen von Unternehmensberatungen bestätigen immer wieder, dass es gerade die unausgesprochenen und unausgelebten Konflikte, fehlende Motivation und mangelnder Teamgeist sind, die Projekte letztlich zu Fall bringen.

Im Umfeld der Software- und der Komponentenentwicklung haben es Tester und die zugehörigen Testmanager nicht leicht. Sie sollen zwar Qualität garantieren, aber möglichst nicht den Betrieb mit nervigen Fehlermeldungen, Rückfragen und Retests aufhalten. Tester sind oft die Überbringer schlechter Nachrichten: Was im Entwicklertest eben noch so gut funktioniert hat, hält plötzlich einem Performancetest nicht stand oder verträgt die Portierung auf eine andere Systemumgebung nicht wirklich. Dann ist es nicht immer einfach, einen Entwickler zur Einsicht zu bringen, dass sein Code tatsächlich fehlerhaft sein könnte.

Tester benötigen deshalb »Fingerspitzengefühl« ebenso wie Durchsetzungsvermögen, wenn sie sich in einem Entwicklungsteam behaupten und ernst genommen werden wollen. Tester müssen Entwickler und Projektleiter davon überzeugen können, dass ihre Arbeit grundlegende Bedeutung für die Qualität und den Erfolg einer Applikation hat und dass es auch für jedes Entwicklungsteam äußerst wichtig ist, über die Entwicklungsqualität jederzeit informiert zu sein. Im Spannungsfeld zwischen Projektleitung (»Wir haben keine Zeit für langwierige Tests«) und Entwicklern (»Lief doch gerade eben noch, das ist sicher ein Bedienfehler«) sind für den Tester und den Testmanager gerade die »weichen Faktoren« überlebenswichtig. Tester kommunizieren stets

nach mehreren Seiten und stehen dann manchmal plötzlich im Mittelpunkt des allgemeinen Interesses, wenn sie z.B. einen »Showstopper« entdecken. Mit solcherlei Anforderungen und »widerborstigen« Umgebungen muss jeder Tester und Testmanager im Alltag zurechtkommen.

Dieses Buch ist aus der Praxis des Softwaretestens heraus entstanden und will Testern wie Testmanagern Hintergrundwissen über Soft Skills geben, das sie bei ihrer täglichen Arbeit anwenden können und das von ihnen auch immer öfter erwartet wird. Das letzte Kapitel nimmt die agilen Projekte in den Fokus und gibt Einblick in die Besonderheiten, auf die sich Tester und Testmanager in agilen Umgebungen einrichten sollten.

Das Buch wendet sich in erster Linie an Tester und Testmanager im Bereich Softwareentwicklung. Die hier behandelten Soft Skills haben aber genauso für Qualitätsbeauftragte aus anderen technischen Bereichen ihre Berechtigung und Gültigkeit; auch für Projektleiter und Entwickler enthält es – hoffentlich – wichtige Informationen. Das Buch will Hintergrundwissen bereitstellen, das sich in der Projektpraxis bewährt hat und immer wieder bewährt.

»Man muss ins Gelingen verliebt sein, nicht ins Scheitern«, lautet ein bekanntes Statement des Philosophen Ernst Bloch. Dieses Buch will dazu beitragen, dass Softwareprojekte eher gelingen als scheitern – und vielleicht manchmal sogar auch noch Spaß machen.

Übrigens: Wenn in diesem Buch von »Tester« oder »Testmanager« die Rede ist, dann ist damit ebenso die Testerin und die Testmanagerin gemeint. Testerinnen und Testmanagerinnen mögen sich ebenso angesprochen fühlen wie Tester und Testmanager. Die männliche Form drückt keine Wertung aus, sondern wurde nur wegen der Einfachheit und besseren Verständlichkeit gewählt.

Heinz Hellerer
Herrsching, August 2012

Inhalt

1	Einleitung: Warum Soft-Skills-Projekte erfolgreich machen	1
1.1	Ziel des Buchs	1
1.2	Was dieses Buch leisten kann und was nicht	2
1.3	Was sind Soft Skills?	3
1.4	Warum Softwareprojekte scheitern	5
1.5	Was sich Tester von ihren Kunden wünschen	10
2	Die Rolle des Testers: Überleben im Spannungsfeld der Stakeholder	11
2.1	Klarstellen der Erwartungen: Die Rollen von Testern und Testmanagern	11
2.2	Rollenkonflikte	13
2.3	Umgang mit Budgets und Schätzungen	21
3	Soziale Kompetenz: Die Intelligenz der Gefühle	27
3.1	Motive und Antreiber	27
3.2	Motivation und Arbeitszufriedenheit	31
4	Die dunkle Seite: Konflikte, Mobbing, Stress	45
4.1	Wie es zu Konflikten kommt	45
4.2	Strategien der Konfliktbewältigung	56
4.3	Wenn's stressig wird: Stress und Stressbewältigung	60
5	Kommunikative Kompetenz: Reden verbindet	67
5.1	Kommunikation ist entscheidend	67
5.2	Warum Kommunikation für Testprojekte so wichtig ist	72
5.3	Kommunikation in Foren, Blogs, Mailinglisten und Wikis	78
5.4	Kommunikation mit Projektleitern und anderen Stakeholdern	84
6	Führungskompetenz: Der Testmanager als Teamchef	89
6.1	Was ist Führung?	89
6.2	Teambildung und Teamentwicklung	98

7	Mittendrin statt nur dabei: Tester und Testmanager in agilen Teams	109
7.1	Warum es agile Entwicklungsprozesse gibt	109
7.2	Kommunikation der Tester mit Product Owners und Scrum Masters	119
7.3	Wo die Reise hingeht	121
	Anhang	123
A	Theoretische Grundlagen und Ergänzungen	125
A.1	Das Konzept der Rolle	125
A.2	Erkenntnis eigener und fremder Lebensmotive nach Steven Reiss ..	127
A.3	Transaktionen und Spiele nach Eric Berne	132
A.4	Tuckmans Phasenmodell der Teambildung	140
A.5	Die neun Eskalationsstufen des Konflikts nach Friedrich Glasl	141
A.6	Techniken der Konfliktbewältigung in Unternehmen	143
A.7	Stressmodelle	153
A.8	Das Kommunikationsmodell von Schulz von Thun	156
A.9	Führungsstile	160
A.10	Das »Agile Manifest« im Originalwortlaut und in der Übersetzung	164
B	Literaturverzeichnis	167
	Index	171

1 Einleitung: Warum Soft-Skills-Projekte erfolgreich machen

Wir müssen als Regel annehmen, dass wir von zwanzig Projekten zehn verlieren, bei fünf auf unsere Kosten kommen, bei vier ordentlich und bei einem tüchtig gewinnen.

F.A. Brockhaus

1.1 Ziel des Buchs

Auch wenn es für Projektleiter, Testmanager und Tester nicht immer einleuchtend ist: Soft Skills, die soziale Seite der Entwicklungsprojekte, entscheiden über den Erfolg oder Misserfolg von Projekten. Fachwissen ist unverzichtbar und ohne Fachwissen gäbe es keine Software, aber Fachwissen im stillen Kämmerlein ist schlicht unnütz. Erst in der Interaktion, in der Zusammenarbeit und Kommunikation mit anderen wird Fachwissen fruchtbar. Doch da beginnen auch die Probleme und der Einsatz der Soft Skills.

»Soft Skills für Tester und Testmanager« – dieses Buch will Sie in Ihren Projekten begleiten und Ihren Arbeitsalltag angenehmer sowie effizienter machen. Es will Sie dabei unterstützen, Ihre Potenziale möglichst optimal zu entfalten und Ihr Fachwissen möglichst effektiv anzuwenden bzw. zu verwerten. Der Vergleich mit einem Rennwagen mag das anschaulich werden lassen: Was bei einem Rennwagen der Motor, sind Ihre Fachkenntnisse sowie Ihr Management- und Projektleitungswissen. Jetzt kommt es darauf an, die Leistung des Motors auch auf die Straße zu bringen. Die Reifen, das sind Ihre Soft Skills. Je besser sich die Reifen für die jeweilige Situation eignen, sei es Regen, Schnee, Hitze oder einfach Schönwetter, desto besser werden die Ergebnisse sein. So ähnlich sieht es mit den Soft Skills aus: Je besser Sie die richtige Vorgehensweise und den richtigen Umgang mit Ihren Kollegen, Mitarbeitern und Kunden wählen, je begabter und geübter Sie sich in der Kommunikation zeigen, desto besser wird das Ergebnis des Gesamtprojekts und speziell Ihres Testteams werden. Aus einer Gruppe von Individuen wird

nämlich dann ein Testteam, das an einem Strang zieht und sich für die gemeinsame Sache einsetzt. An dieser Stelle sollte auch einmal betont werden, dass es sich bei Soft Skills und Sozialkompetenzen nicht um ein Geheimwissen handelt oder um schwer definierbare Eigenschaften wie »Charme«. Soft Skills sind bis zu einem gewissen Grad erlernbar, aber wer sich, seine Persönlichkeit und seine Verhaltensweisen ändern will, benötigt viel Ausdauer. Dieser Einblick in die Welt der Soft Skills zeigt Ihnen auf, auf welchen Gebieten Sie sich weiterentwickeln und welche Skills Sie verbessern können. Seien Sie sich aber auch bewusst, dass mit dem Lesen allein noch keine Verhaltensänderung einhergeht. Entsprechende Soft-Skills-Seminare oder Workshops können hier unterstützen.

Gerade diejenigen Leser, die eine technische Ausbildung hinter sich haben, wurden bisher vermutlich wenig mit Themen aus der Psychologie und Organisationstheorie konfrontiert. Dieser Zielgruppe will das Buch einen ersten Eindruck vermitteln und sie zur Weiterbeschäftigung mit dieser im Projektalltag immer wichtiger werdenden Thematik animieren.

1.2 Was dieses Buch leisten kann und was nicht

Dieses Buch liefert Ihnen einen kompakten Überblick über das momentane Wissen bezüglich Soft Skills, wobei es ausschließlich um solche Soft Skills geht, die für die Bereiche Test und Qualitätssicherung in Softwareprojekten relevant sind.

Ein Buch kann Informationen vermitteln, aber es wird ad hoc keine Verhaltensänderungen herbeiführen, weder beim Leser noch seinen Geschäftspartnern. Wer – vielleicht aufgrund dieses Buchs – bei sich oder anderen Soft Skills entdeckt, die er gerne entwickeln und verbessern möchte, versucht dies am besten mithilfe eines Coachs oder in dieser Beziehung erfahrenen Psychologen. Alle Soft Skills haben mit der Persönlichkeit eines Menschen zu tun. Wer seine eigene Persönlichkeit besser kennenlernen, Teilbereiche ändern oder innere Barrieren abbauen will, der lässt sich auf einen längeren Prozess ein, bei dem es mit dem Lesen eines Buchs allein nicht getan ist. Persönlichkeit ist nichts, in dem man mal eben so nebenher mit einem Buch als »Gebrauchsanweisung« herumstochern könnte. Wer sein Verhalten wirklich ändern und sich entwickeln will, der hat sich einer langfristigen Aufgabe verschrieben und wird eventuell professionelle Hilfe in Anspruch nehmen müssen, um ans Ziel zu gelangen.

Erste Schritte bei der Verhaltensänderung anderer unternimmt man am besten erst einmal vorsichtig in seinem Projekt im kleinen Kreis. Auch hier macht die Übung den Meister.

1.3 Was sind Soft Skills?

Die wörtliche Übersetzung mit »weiche Fähigkeiten« ist irreführend und ein Terminus wie »soziale Kompetenz« trifft es deutlich besser. Inzwischen hat sich der Begriff »Soft Skills« in der Projektwelt und auch bei Stellenausschreibungen so sehr eingebürgert, dass man nicht mehr auf ihn verzichten kann.

Trotz aller Gebräuchlichkeit macht der Ausdruck in der Projektwelt Probleme und wird gerne falsch verstanden. Die Konnotationen des Wortes »soft« mit »sanft«, »nachgiebig«, »schwächlich« schwingen bei dem Begriff »Soft Skill« im Subtext immer mit. Er lässt viele, gerade wenn sie aus dem technischen Bereich kommen und mit der »Psychodenke« nichts anfangen können, an Schmusekurs und »Sozialklimbim« denken, an eine Veranstaltung für Weicheier und Kamillenteetrinker, mit der gestandene Tester und Testmanager nichts zu tun haben (wollen). Darum geht es aber nicht.

Es geht bei Soft Skills unter anderem um die Fähigkeit, sich selbst zu motivieren und im Team ein bestimmtes Ziel – in diesem Falle ein Projektziel – zu erreichen. Leider gibt es keine eindeutige und anerkannte Definition dessen, was Soft Skills im Kern ausmacht und wo sie sich genau abgrenzen lassen. Einigkeit herrscht jedoch insoweit, als es sich bei den Soft Skills um eine Gruppe von Eigenschaften handelt, die im Zusammenhang mit dem Umgang mit sich selbst und mit anderen stehen. Zum Umgang mit sich selbst gehören Selbstvertrauen, Selbstwert und Eigenverantwortung, was u.a. durch Selbstbeobachtung und Disziplin erreicht und gestärkt werden kann. Im Umgang mit anderen gehören zu den Soft Skills Kritikfähigkeit, Menschenkenntnis, Zivilcourage, Konfliktfähigkeit, Motivation und Teamfähigkeit. Beim Testmanager in der Rolle einer Führungspersönlichkeit darf man entsprechend die Vorbildfunktion zu den Soft Skills zählen, die mit Entscheidungsfähigkeit, konsequentem Handeln und Durchsetzungsvermögen zusammenhängt.

Definition Soft Skills

Es handelt sich bei den Soft Skills also um ein nur schwer abgrenzbares Bündel an individuellen Fähigkeiten und Eigenschaften, die die Zusammenarbeit in einem Team oder die Führung eines solchen ermöglichen. Eine eindeutige und allgemein anerkannte Definition ist in der entsprechenden Literatur nicht vorhanden. Soft Skills haben jedenfalls damit zu tun, wie ein Mensch auf die Herausforderung einer

bestimmten Situation reagiert, ob der Situation angemessen und ob er sie zum eigenen Vorteil und zum Vorteil seiner Gruppe nutzen kann. »Nutzen« meint dabei nicht »ausnutzen«, sondern etwas, von dem alle Gruppenmitglieder profitieren und das soziale Beziehungen nicht gefährdet. Soft Skills sind eine intelligente Antwort auf persönliche Herausforderungen und Herausforderungen im Umgang mit anderen. Soft Skills bilden also keine Reihe von Eigenschaften, die sich einfach aufzählen und kategorisieren lassen – es kommt auf die jeweilige Situation an. Beispielsweise kann »Durchsetzungsfähigkeit« in manchen Fällen eine der Situation angemessene Verhaltensweise sein. Wenn ein Testmanager gegenüber der Projektleitung Durchsetzungsfähigkeit zeigt, kann das ein echter Soft Skill sein. Wenn die Durchsetzungsfähigkeit aber zur kompromisslosen Rücksichtslosigkeit ausartet, ist sie sicher kein Soft Skill mehr, da das Verhalten der Situation nicht mehr angemessen ist.

*Sind Soft Skills ein
Geschenk der Evolution?*

Offensichtlich sind Soft Skills – so verstanden – kein reines Zivilisationsprodukt, sondern haben ihre Basis in der Biologie des Menschen. Für den Göttinger Neurobiologen Gerald Hüther – er spricht von »emotionaler Kompetenz« – bilden Soft Skills ein wesentliches Ergebnis der menschlichen Evolution und eine Voraussetzung für das Überleben der Spezies Mensch: »... das war möglicherweise das wirklich entscheidende und während der gesamten Menschheitsgeschichte in allen Kulturen für den Fortpflanzungserfolg bedeutsame Kriterium der Partnerwahl sowohl von Männern als auch von Frauen – musste der jeweilige Fortpflanzungspartner die für eine gelingende Aufzucht der gemeinsamen Kinder erforderlichen psychoemotionalen Eigenschaften besitzen: Einfühlungsvermögen, Umsicht und Verlässlichkeit, also das, was wir heute noch als emotionale Kompetenz bezeichnen. Diese hochkomplexen, während der frühen Kindheit durch Erziehung und Sozialisation gewonnenen Fähigkeiten und die ihnen zugrunde liegenden Anlagen sind durch den Prozess der sexuellen Selektion während der gesamten Phase der Menschheitsentwicklung bevorzugt ausgelesen worden« [Hüther 2010, S. 96 f.].

Aus dem Blickwinkel der Neurobiologie sind also grundlegende Sozialkompetenzen in jedem Menschen bereits genetisch angelegt. Jetzt kommt es »nur« noch darauf an, wie wir mit dieser Veranlagung entsprechend intelligent umgehen.

Was haben das Sozialverhalten und die emotionale und soziale Kompetenz nun mit der rauen Projektwirklichkeit zu tun? Das wird deutlich, wenn man das Pferd von hinten aufzäumt und die Frage stellt, warum Softwareprojekte eigentlich so oft scheitern und offensichtlich so schwer in den Griff zu bekommen sind.

1.4 Warum Softwareprojekte scheitern

Nach dem Goldwyn-Report von 2008 [Goldwyn] scheitert ein Drittel aller Softwareprojekte, bei einem weiteren Drittel kommt es zu massiven Überziehungen des Zeitrahmens oder des Budgets. Nach dem Chaos-Report der Marktforschungsfirma Standish Group aus Massachusetts wurden im Jahr 2009 32% aller Softwareprojekte in den USA erfolgreich abgeschlossen, 44% waren problembehaftet und 24% wurden eingestellt. Auch wenn die Ergebnisse diskutierbar und nur begrenzt auf den deutschsprachigen Markt anwendbar sind, lässt das Ergebnis doch aufhorchen.

IT-Projekte (damit ist hier sowohl die Entwicklung als auch die zugehörige Qualitätssicherung gemeint) haben die unangenehme Eigenschaft, dass sie in sehr kurzer Zeit völlig aus dem Ruder laufen können. Da die IT noch eine relativ junge Ingenieurwissenschaft ist, fehlen in den Unternehmen die jahrzehntelangen Erfahrungen, die man z.B. mit Bauvorhaben oder in der Verfahrenstechnik hat. IT-Projekte können aber ganz schnell so komplex werden, dass dem Management, der Projektleitung und allen weiteren Beteiligten jeder Überblick verloren geht – was im Ernstfall dann aber keiner zugeben will.

Ein wunderbares Beispiel für einen derartigen Komplexitäts-GAU bietet z.B. die Firma Levis (die mit den Hosen). Levis wollte seine zersplitterte IT-Landschaft durch ein umfassendes SAP-System ablösen und verlor dabei Unsummen Geld, wie das Magazin »Harvard Business Manager« zu berichten wusste: »Das Risiko schien überschaubar. Das Budget lag unter fünf Millionen Dollar. Aber sehr bald brach bei dem Projekt die Hölle los: Ein großer Kunde, Wal-Mart, verlangte, dass das IT-System von Levi Strauss mit seinem eigenen Supply-Chain-Managementsystem verbunden werden müsse, was eine zusätzliche Schwierigkeit bedeutete. Unzulängliche Verfahren bei der Bilanzierung und betrieblichen Steuerung führten dazu, dass Levi Strauss fast jedes Mal seine Quartals- und Jahresergebnisse berichtigen musste. Während des Wechsels auf das neue System konnte das Unternehmen keine Aufträge ausführen und musste seine drei Logistikzentren in den USA für eine Woche schließen. Im zweiten Quartal 2008 musste es wegen des verpfuschten Projekts eine Ergebnisbelastung von 192,5 Millionen Dollar hinnehmen. Chief Information Officer David Bergen kostete es den Job« [Harvard].

In diesem Fall führten hohe Komplexitäten des Projekts, schlechtes Projektmanagement, gepaart mit Selbstüberschätzung und mangelnden Projektmanagementkenntnissen ins Desaster. Eine realistische Einschätzung der eigenen Möglichkeiten und Grenzen sowie der Möglich-

Wie ein Softwareprojekt aus dem Ruder laufen kann

keiten und Grenzen der Projektmitarbeiter ist wiederum ein typischer Soft Skill.

Neben all den Verlusten an Zeit und Geld führen derartige Chaosprojekte zur Demotivation aller Beteiligten einschließlich derjenigen Kollegen im Unternehmen, die das Projekt aus der Ferne beobachtet haben und anschließend für ähnlich gelagerte Vorhaben aus gutem Grund nicht mehr zu motivieren sind.

Nur wenige Projekte scheitern an technischen Problemen.

Nur bei einer Minderheit der gescheiterten Projekte sind technische Probleme ausschlaggebend für den ausbleibenden Erfolg: Hier sind der Einsatz neuer, unbekannter Tools (»First Movers«), neuer Computersprachen oder schlicht die Unterschätzung der Komplexität und des technischen Aufwandes die schlimmsten »Problembären«.

Die meisten Projekte aber scheitern an Problemen auf der menschlichen, der emotionalen und kommunikativen Ebene: an mangelnder Methodenkompetenz, an der Arroganz der Projektleitung oder der Entwickler, an mangelnder Menschenkenntnis, fehlerhafter Selbsteinschätzung und fehlender Führungskompetenz sowie an ungelösten Konflikten im Projektteam, an der unzureichenden Information an die Stakeholder, an ausbleibenden Zielvorgaben und latenter Bunkermertalität.

Bei der Auswertung von diesbezüglichen Studien, Umfragen und Untersuchungen fällt auf, dass es eigentlich immer dieselben Fehlleistungen sind, die IT-Projekte scheitern lassen. Die meisten dieser Fehlleistungen haben ihre Ursache in unzureichenden Soft Skills. Hier eine Liste der Top-Projektkiller sowie der Hinweis auf die hierbei fehlenden Soft Skills:

Mangelnde Kommunikation ist der Projektkiller Nr. 1.

■ Mangelhafte Projektkommunikation

Immer noch ist die Kommunikation in vielen Unternehmen und Projekten ein Stiefkind, dem viel zu wenig Beachtung geschenkt wird. Zur Kommunikationskultur in Projekten zählen die Informationspolitik wie auch das Management von Meetings und Besprechungen.

Gefragt ist hier der Soft Skill **Kommunikationskompetenz** und damit die Fähigkeit zu moderieren, zu überzeugen und zu verhandeln.

Die Klärung der Ziele ist kein Luxus.

■ Unklare Ziele

Projekte werden häufig vom Vertrieb oder von der Geschäftsleitung initiiert, ohne die Vorgaben und Projektziele genau zu klären. Unklare oder widersprüchliche Anforderungen führen aber zu endlosen Diskussionen, zu Konflikten und Machtspielen. Wenn eine

Qualitätsvorgabe z. B. lautet: »Das System sollte robust auf Fehlbedienung reagieren«, so sagt dies einem Manager oder Vertriebsmitarbeiter vielleicht einiges, einem Entwickler, Tester oder Qualitätsbeauftragten aber eher gar nichts. Denn was genau ist eine Fehlbedienung? Was heißt »robust«? Was darf auf keinen Fall passieren? Wer bedient die Anwendung? Welche Kenntnisse hat diese Person?

Gefragt ist hier neben dem Soft Skill **Kommunikationskompetenz** auch **Führungskompetenz**, wozu die Vorgabe klarer Haupt- und Teilziele gehört.

■ Schlechte Projektvorbereitung und -planung

Nach wie vor werden Projekte oft »aus dem Boden gestampft«, ohne Rahmenbedingungen wie die Verfügbarkeit von Ressourcen genügend zu bedenken. Gerade dann, wenn Projekte den Mitarbeitern zusätzlich zu ihrer täglichen Arbeit »aufgedrückt« werden, führt dies zu einer schwer abschätzbaren Zusatzbelastung. Mit schlecht oder gar nicht motivierten Mitarbeitern, die das Projekt vielleicht als Zumutung und reine Erschwernis ihres Tagesgeschäfts sehen, kommt es in jedem Fall zu erheblichen Zeitverzögerungen, die Kreativität bleibt auf der Strecke und der Projekterfolg ist von Anfang an gefährdet.

Gefragt sind hier **Projektmanagement-Know-how**, **Arbeitstechnik**, **Führungskompetenz** und **Konfliktlösungskompetenz**.

■ Arroganz, Selbstüberschätzung und Unerfahrenheit

Ein Hauptproblem unerfahrener Projektleiter sind viel zu optimistische Annahmen. Beispielsweise ist die Zeitschätzung von Softwareentwicklungsprojekten durchaus nicht trivial; sie erfordert viel Erfahrung und Menschenkenntnis (was kann man den Projektbeteiligten zumuten, was besser nicht?). Gerade jung-dynamische Manager und Projektleiter haben oftmals das Gefühl, sie könnten sich profilieren, indem sie Projekte nach dem Motto »Neue Besen kehren gut« besonders schnell und kostenbewusst durchdrücken. Jeder Versuch, über Druck, Wochenendarbeit u.Ä. falsche Zeitschätzungen bzw. ein schlingerndes Projekt wieder auf Kurs zu bringen, funktioniert allerdings bestenfalls kurzzeitig, vergrault aber gerade die Top-Leister, macht Engagement und Kreativität im Projekt zunichte und führt auf Dauer ins Chaos.

Nicht zu unterschätzen:
Projektkiller Arroganz

Interessant sind in diesem Zusammenhang die Beobachtungen der Managementberatung Kienbaum aus Gummersbach zum möglichen Scheitern von High Potentials: »In Zeiten des Fach- und Führungskräfte-Mangels haben überdurchschnittlich qualifizierte Absolventen und Berufseinsteiger ausgezeichnete Karriereaussichten. Trotzdem scheitern einige der sogenannten High Potentials im Berufsleben, so die Erfahrung vieler Personalchefs in Deutschland, Österreich und der Schweiz. Gründe hierfür sind aus Sicht der HR-Leiter vor allem mangelnde Soft Skills: Scheitert ein deutscher High Potential, liegt dies in 94 Prozent der Fälle an seiner Selbstüberschätzung und zu 89 Prozent an der mangelnden Fähigkeit zur Selbstkritik, gaben die für eine aktuelle Kienbaum-Studie befragten Personaler an. In der Schweiz sind die Selbstüberschätzung (95 Prozent) und in Österreich die mangelnde Fähigkeit zur Selbstkritik (93 Prozent) ebenfalls Hauptgründe für das Scheitern von High Potentials« [Kienbaum].

Kienbaum hatte für die Studie »High Potentials 2011/2012« insgesamt 469 Unternehmen aller Größen in Deutschland, Österreich und der Schweiz befragt. Man darf also davon ausgehen, dass es sich hierbei nicht um »bedauernswerte Einzelfälle« handelt, sondern dass die unzureichende Einschätzung der eigenen Person sowie ihrer Möglichkeiten und Grenzen ein flächendeckendes Problem in der deutschsprachigen Industrielandschaft darstellt.

Das Problem der
»Arroganz der
Spezialisten«

Die »Arroganz der Spezialisten« ist in jeder Organisation ein Problem und immer ein möglicher Stolperstein für ein Projekt. Dazu der Managementexperte und Professor für Unternehmensführung Fredmund Malik: »Arroganz und Indifferenz sind die typischen Untugenden der Spezialisten, und die sind gravierende Probleme für jede Organisation. Sie gehören auf die Liste der Todsünden wider den Geist einer guten Organisation. In diesem Sinne falsch verstandenes Spezialistentum ist eine der, wenn nicht überhaupt die wesentliche Ursache für die so oft beklagten Kommunikationsprobleme und für die weniger häufig zitierten, aber mindestens so wichtigen Probleme des Realitätsverlustes in so vielen Organisationen. Spezialisten kennen ihre Realität, aber die Realität der Organisation interessiert sie nicht. Deshalb können sie völlig ungeniert mit der Selbstsicherheit des Ahnungslosen an eben dieser Realität vorbeioperieren« [Malik 2006, S. 102].

Gefragt sind hier **Menschenkenntnis** und **realistische Selbsteinschätzung**, **Führungskompetenz** sowie **Methodenkenntnisse**.

■ Fehlendes Leitungs- und Führungs-Know-how

Viele Projektleiter (wobei ich hier ausdrücklich auch den Test als ein eigenes Projekt verstanden wissen möchte – auch ein Testmanager ist also ein Projektleiter) verfügen über ein profundes Fachwissen, aber über wenig Kenntnis im Umgang mit Teams, Gruppen, schwierigen Zeitgenossen und mit Konflikten. Wenn die Projektleiter aus dem technischen Bereich kommen, haben sie in ihrer Fachausbildung meist wenig bis gar nichts über soziale Kompetenzen erfahren. Jeder Handwerksmeister mit einem Ausbilderschein weiß im Schnitt mehr über Menschenführung und Pädagogik als ein Informatiker oder Ingenieur, der frisch von der Universität kommt.

Starrheit und Dogmatismus sind dann oft die Folge einer gewissen Unsicherheit im zwischenmenschlichen Umgang. Der dogmatische Umgang mit Projektstrategien, einer einmal ausgesuchten Programmiersprache oder die frühe und diskussionslose Vorgabe von Tools sind Folgen dieser Unsicherheit. Dieses Vorgehen aber führt bei den Mitarbeitern nach einiger Zeit mit Sicherheit zu Widerstand und Renitenz: »Was die sich da wieder ausgedacht haben ...«, lautet die ausgesprochene oder unausgesprochene Frage. »Die« sind aber in jedem Fall »die anderen«, das sind nicht wir. Schon löst sich der Teamgeist auf und das Image des Managements bzw. der Projektleitung hat die ersten Kratzer.

Zu allem Überflus kämpfen Projektleiter aber nicht nur an einer Front. Neben dem »eigenen« Team haben sie noch ein ganze Reihe von Stakeholdern über und neben sich: das eigene Management, das Management des Kunden, das Controlling, das Marketing und so weiter. Alle Beteiligten erwarten Informationen über das Projekt, transparente Zeitpläne und Auskünfte über Ressourcen und Budgets. Die Kommunikationsfähigkeit eines Projektleiters wird also schon im normalen Projektalltag erheblich strapaziert, auch wenn sich noch gar nichts Besonderes ereignet hat.

*Dogmatismus und
fehlende
Diskussionskultur*

Gefragt sind hier **Führungskompetenz** und **Kommunikationskompetenz**.

Über Projektfehler und das Scheitern von Projekten finden sich inzwischen größere Mengen an Untersuchungen und Veröffentlichungen. Die Gründe, warum Projekte scheitern, sind – wenn auch mit anderen Nuancen – immer ähnlich gelagert: Das Management und die Projektleitung haben es in der Hand, ein Projekt zum Erfolg zu führen bzw. es zum Absturz zu bringen.

1.5 Was sich Tester von ihren Kunden wünschen

Allerdings – man muss es einfach mal sagen, selbst wenn es wie Publikumsbeschimpfung klingt: Auch Kunden sind keine reinen Engel.

*Wünschenswert:
Unterstützung durch den
Auftraggeber*

Sie könnten einem Testteam das Leben so leicht machen, wenn sie nur einmal klar sagen würden, was sie eigentlich wollen. Die Requirements, die verbindlichen und belastbaren Kundenanforderungen herauszubekommen, erweist sich häufig als schwieriger und langwieriger Prozess. Ganze Scharen von Business-Analysten sind damit beschäftigt, den Kunden zu löchern, UML-Charts zu zeichnen, Workshops durchzuführen, um dann zuletzt aus wenigen kryptischen Bemerkungen so etwas wie eine stabile Kundenanforderung zu destillieren. Die Tester sollten ihnen dankbar sein für diese Vorarbeit, denn für den Test gibt es nichts Schlimmeres als undurchsichtige und unklare Kundenanforderungen.

*Die Anforderungen des
Kunden sind das A und O.*

Genau an dieser Kante stürzen leider viele Projekte ab. Wer als Projektverantwortlicher nicht herausbekommt, was der Kunde wirklich will, oder kein Gefühl dafür entwickelt, was der Kunde sicher nicht will, der erleidet früher oder später Schiffbruch, allerspätestens bei der Produktpräsentation. Darum, lieber Kunde, versuche uns zu sagen, was du wirklich willst, was deine Traumsoftware am Ende leisten soll. Wenn du es selbst noch so nicht genau wissen solltest, dann richte deine Prozesse entsprechend ein. Für häufig wechselnde Kundenanforderungen eignen sich agile Vorgehensmodelle wie Scrum.

*Auch Kunden benötigen
Soft Skills.*

Kunden und Auftraggeber machen mit Soft Skills wie Kommunikationskompetenz, Entscheidungsfreude und Delegationsfähigkeit den Projekten und sich selbst das Leben leichter.

2 Die Rolle des Testers: Überleben im Spannungsfeld der Stakeholder

Das hier ist eine verdammt harte Galaxis.

Douglas Adams

2.1 Klarstellen der Erwartungen: Die Rollen von Testern und Testmanagern

Jeder Mitarbeiter in einem Unternehmen oder Projekt erfüllt eine Rolle, die durch seine Tätigkeit vorgegeben ist. In technischen Projekten wird niemand zufällig beschäftigt, sondern allein wegen der Rolle, die er in dem Projekt einnimmt. Über seine Rolle ist jeder Tester und Testmanager in das Projekt und auch in die Organisation eines Unternehmens eingebunden – in das eigene Unternehmen und/oder in das Unternehmen des Kunden.

Eine ausführliche Behandlung des Rollenkonzepts aus Sicht der Organisationspsychologie findet sich in Abschnitt A.1.

Aus der Praxis:

Man sollte meinen, die Rolle eines Testers oder des Testmanagers in einem Projekt sei eindeutig definiert: Der Tester ist eben dazu da, Tests durchzuführen und Fehler in einer Software zu finden. Der Job des Testmanagers besteht offensichtlich darin, die Tester zu koordinieren, Testziele festzulegen und die Testausführung zu überwachen. So einfach ist das im praktischen Projektleben aber nicht.

Wiederholt ist mir begegnet, dass der Kunde automatisierte Tests erwartete, bevor überhaupt die Anforderungen vollständig geklärt, geschweige denn manuelle Tests geschrieben waren (»Wir automatisieren gleich, das spart Kosten.«).