Kundenkommunikation: Was ich mir hätte schriftlich geben lassen sollen

Kunde – Entwickler, Entwickler – Kunde

Wenn Kunden als „Nicht-Techies“ mit Softwareentwicklern zusammentreffen und ein Projekt bewältigen wollen, ist das zentrale und wichtigste Thema für einen erfolgreichen Abschluss die Kommunikation zwischen den Beteiligten. Dass dieses Thema allerdings auch das vielleicht Komplizierteste während eines Projekts ist, stellt sich immer wieder heraus. Welches könnten die Wege sein, um einen reibungslosen Projektablauf zu gewährleisten?

von Andreas Wenk

Es ist kein Geheimnis: Wir Softwarearchitekten, Developer, Coder, Sysadmins und Hacker sind irgendwie alle mehr oder weniger Nerds [1]. Einige von uns sind aber in Positionen, in denen wir mit dem Kunden in Kontakt treten und ein Softwareprojekt wuppen wollen. Und einige aus dieser Gruppe haben die Aufgabe, das Projekt mit unserem Kunden zu planen, durchzuführen und abzuschließen. Und – Hand aufs Herz – alle kennen wir die Probleme, die dabei entstehen. Sätze dem Kunden gegenüber wie „das habe ich Ihnen beim letzten Meeting beschrieben und dachte das sei klar“ (war aber wohl doch nicht klar) oder vom Kunden „beim letzten Meeting haben Sie aber gesagt, dass das

so und so funktionieren wird und bis morgen fertig ist“ (wurde so aber nie gesagt und über ein Timing gar nicht geredet) sind uns nicht fremd. Wo liegt das Problem? Das Problem ist mangelhafte Kommunikation. Und ich meine alles, was mit Kommunikation während einem Projekt zu tun hat: Angebote, Meetings, Telefonate mit dem Kunden, Timing – um nur einige zu nennen.

In diesem Artikel gehe ich der Problematik auf den Grund und stelle aus meiner Sicht geeignete Lösungsansätze dar, die sich allgemein bewährt haben. Dieser Artikel befasst sich dabei nicht mit Projektmanagement im klassischen Sinn – dafür gibt es im w3 viele gute Ressourcen. Nein, es geht um das wirkliche Leben. Wir haben keine Zeit, noch ein Regelwerk und Ablaufmuster zu lernen (und womöglich eh nicht zu nutzen), sondern wir müssen zuerst einmal verstehen, wie wir mit sehr einfachen Mitteln unseren Kunden zufriedenstellen, indem wir verstehen, was er von uns will, ein gutes Ergebnis – respektive eine gute Software – entwickeln und auch noch Spaß (was sich durch Erfolg auszeichnet) dabei haben. Ganz im Sinne von „Keep it simple!“.

Ein Meeting halten – aber richtig

Ein Meeting ist ein Treffen einer Gruppe von Personen, die gemeinsam ein Ziel erreichen wollen und dieses besprechen. Dabei ist es extrem wichtig, dass eine Person ein Protokoll schreibt und dieses nach dem Meeting an alle Beteiligten sendet und an einer zentralen, für alle Beteiligten zugänglichen Stelle, ablegt. Der Initiator sollte vorweg eine Agenda erstellen, die er allen Beteiligten zukommen lässt. Bei regelmäßigen Meetings kann das auch im Vorfeld von allen Beteiligten geschehen, um ihre zu besprechenden Punkte anzumelden. Das Protokoll sollte am Ende kurz vorgelesen werden, damit kontrolliert werden kann, ob alle Punkte richtig dargestellt und notiert wurden.

Merke: Ein Meeting ohne Protokoll hat nicht stattgefunden!

Es ist sehr, sehr hilfreich, wenn es einen Moderator für das Meeting gibt. Das kann der Initiator sein oder aber auch ein anderer Beteiligter. Weiterhin sollte drauf geachtet werden, dass ein Meeting keine „Überlängen“ hat – sprich: Halten Sie es so kurz wie möglich, aber so lang wie nötig.

Der Anfang: Kunde droht mit Auftrag

Szenario: Alle sind happy. Ein Kunde meldet sich bei uns und will sein Vertrauen in uns setzen. Wir sind die von ihm ausgewählte Firma, Agentur oder der Freelancer, der seine nächste Killer-App umsetzen soll. Wir treffen uns mit dem Kunden, trinken Kaffee, lachen und freuen uns. Dann gehen wir auseinander, haben festgestellt, dass die Umsetzung überhaupt kein Problem ist und haben bereits zugesagt, dass der Zeitplan kein Problem darstellt. Wir setzen uns also ein paar Tage später an unseren Rechner (wir haben nicht sofort Zeit das zu tun, weil wir noch ein anderes Projekt für einen mittlerweile ungemütlichen Kunden abschließen müssen, bei dem wir ziemlich hinten dran sind) und schreiben zwei (evtl. auch drei) Seiten Angebot, in dem wir aus dem Kopf grob das besprochene aufnehmen. Darunter setzen wir einen Preis, von dem wir denken, dass der Kunde auf jeden Fall Ja sagt – vielleicht ein bisschen knapp bemessen, aber das wird schon.

Wenn Sie sich in diesem Szenario wiederfinden, sollten Sie unbedingt weiterlesen. Ich prophezeie nämlich, dass dieses Projekt tüchtig in die Hose gehen wird. Denn schon in dieser kurz beschriebenen ersten Phase sind eklatante Fehler vorhanden.

Käffchen?

Das erste Treffen und Kaffee trinken mit dem Kunden ist eine wichtige Sache und sollte nicht unterschätzt werden. Man lernt sich kennen, man beschnuppert sich und stellt dabei fest, ob man „miteinander kann“. Klar können wir uns als Auftragnehmer den Kunden nicht aussuchen. Aber der Kunde kann sich aussuchen, ob er mit uns zusammenarbeiten will. Deshalb ist es wichtig, sich bei solch einem Treffen auf den Kunden einzulassen. Wir sollten also schon vorher Informationen über die Firma und unsere Gesprächspartner einholen. Natürlich sprechen wir auch in allgemeinem Charakter über das Projekt. Aber das war’s dann auch schon. Ganz wichtig für diesen ersten Termin sind folgende Fragestellungen:

- Wie schätze ich den Kunden ein (locker und zielstrebig, ernst und hektisch, technisch versiert oder nicht, ehrlich, hinterlistig usw.)?
- Welchen von uns geschätzten Umfang hat das Projekt?
- Wie wichtig ist das Projekt für den Kunden?
- Welches Timing strebt der Kunde an?
- Wie sitzt das Geld? Locker oder in einem Safe?

Wenn es möglich ist, sollten immer zwei Personen von beiden Parteien dabei sein. Denn (wichtig!) einer schreibt mit. Lesen Sie zum Thema „Meeting“ bitte den Kasten „Ein Meeting halten – aber richtig“. Ein weiterer wichtiger Punkt: Beim ersten Meeting werden keine Zusagen oder Schätzungen in Sachen Kosten und Timing getroffen. Diese sind in 90 % der Fälle unrealistisch, drängen uns dazu, im Nachhinein diese erfüllen zu wollen und wecken bei unserem Kunden eventuell falsche Aussichten oder Hoffnungen. Das wiederum kann schon die erste Enttäuschung für den Kunden bedeuten, wenn wir ihm ein paar Tage später (im Angebot) sagen müssen, dass es doch länger dauert und teurer wird. Misstrauen ist geboren.

Gesprächszusammenfassung des ersten Meetings

Im nächsten Schritt sind wir an der Reihe. Wir schreiben das Protokoll ins Reine, senden es unserem Kunden zu und bitten ihn, zu bestätigen, dass alles richtig ist.

Übrigens: Ist Ihnen aufgefallen, dass unser vermeintlicher Kunde noch gar kein Kunde ist? In der Tat: Unser vermeintlicher Kunde ist erst Kunde, wenn er uns den Auftrag erteilt hat, indem er uns eine schriftliche Auftragsbestätigung zugesendet hat. Solange das nicht der Fall ist, gibt es auch keinen Grund, uns ausschließlich mit diesem „potenziellen Kunden“ zu beschäftigen. Eine Trefferquote für eine Auftragszusage von 1:10 ist nicht zu pessimistisch geschätzt.

Vom Gleichen sprechen – was soll getan werden?

Sobald der Kunde unserem Protokoll zugestimmt hat, machen wir uns an eine Anforderungsanalyse. Wooo – klingt akademisch. Ist es aber gar nicht. Dieser Schritt dient dazu, zu erarbeiten, dass von uns und dem Kunden von der gleichen Sache

gesprochen wird (dass richtig kommuniziert wird). Dafür können wir jetzt eine 100-seitige Beschreibung des Projekts verfassen, laufen aber Gefahr, dass diese nicht gelesen wird. Viel einfacher (mit etwas Übung) ist es, die Anforderung an die zu erstellende Software in einer Mindmap [2], einem Flussdiagramm [3] (das mit den Kästchen, Kreisen, Pfeilen und Dreiecken) oder irgendeiner anderen grafischen Form festzuhalten (aber bitte kein UML-Diagramm – denn das versteht unser „zukünftiger" Kunde leider nicht). Das Ganze natürlich auch mit einer Anforderungsbeschreibung. Hier gilt allerdings auch wieder: Keep it simple!

Dieses Vorgehen hat den großen Vorteil, dass Sie selbst viel besser verstehen, was genau getan werden bzw. wie die Software aussehen soll. Denn beim Erstellen der grafischen Visualisierung stoßen Sie bereits auf Fallstricke und müssen diese berücksichtigen. Das wird sich zum einen im Preis und im Timing niederschlagen, weil ein Mehraufwand entsteht. Aber es trägt auch maßgeblich zum schnelleren, sichereren und exakteren Erfüllen der Anforderungen bei.

Nachdem wir die Projektbeschreibung mit dem Kunden besprochen haben und sicher sind, dass wir „vom Gleichen sprechen", können wir uns an das Angebot und das Pflichtenheft machen.

Angebot und Pflichtenheft sind zwei Paar Schuhe – denn nicht jeder muss später die Kosten kennen.

Angebot und Pflichtenheft

Ist das nicht dasselbe? Nein! Das Angebot beschreibt die Kosten des Projekts bzw. unserer Dienstleistung. Das Pflichtenheft beschreibt die Funktionalität der zu erstellenden Software. Das sind aus meiner Sicht zwei verschiedene Paar Schuhe. Außerdem soll jeder Projektbeteiligte das Pflichtenheft bekommen, aber nicht jeder muss wissen, welche Kosten entstehen.

Das Angebot: In großen Softwareunternehmen wie SinnerSchrader aus Hamburg gibt es eine Vertriebsabteilung und eine Entwicklungsabteilung. Die Aufgabenverteilung ergibt sich bereits aus der Definition der Abteilungen. Der Vertrieb ist für das Angebot zuständig und die Entwicklungsabteilung für das Pflichtenheft, wobei beide Abteilungen natürlich interagieren und eng zusammenarbeiten. In kleineren Agenturen oder Firmen ist das nicht immer getrennt. Deshalb müssen auch die Entwickler bei der Erstellung des Angebots mithelfen oder dieses sogar schreiben (wie ich das regelmäßig mache).

Beim Erstellen des Angebots bietet sich ein Tabellenkalkulationsprogramm (z. B. Calc aus der OpenOffice-Familie [4]) an. Schreiben Sie die einzelnen Punkte nach Bereichen auf. Legen Sie dann einen zentralen Stundenpreis fest und

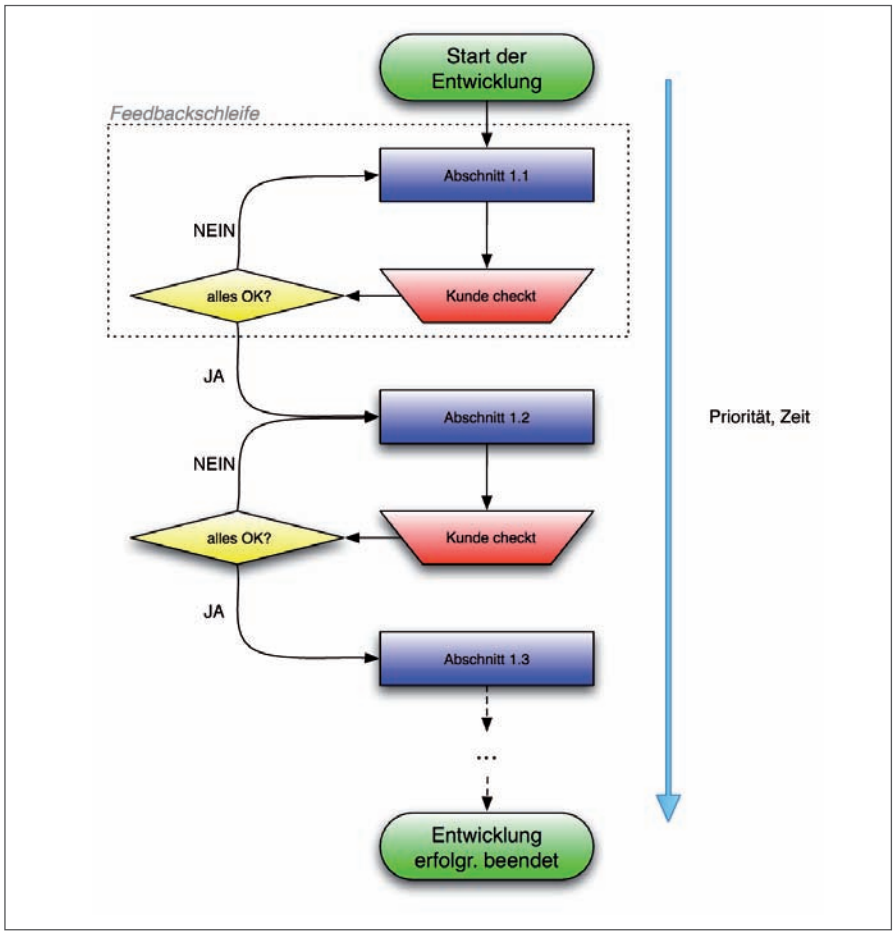

Abb. 1: Feedbackschleife zwischen Kunde und Auftragnehmer

schreiben Sie die Stunden für die einzelnen Positionen auf. Je detaillierter Sie das tun, desto genauer wird der Preis am Ende sein. Die einzelnen Bereiche werden dann einfach kumuliert. So bekommen Sie ein Gefühl, ob zum einen die Stunden und auch der Preis für den Bereich realistisch sind. Und wenn das dann alles fertig ist, schlagen sie einen „kalkulatorischen Aufschlag" von mindestens 20 % des errechneten Preises oben drauf. Besser 30 %. Das ist der Headroom oder Puffer, der Sie davor bewahrt, bei unerwarteten Komplikationen ein finanzielles Desaster vorzufinden.

Letztlich geht es dann nur noch um Fakten. Der Kunde sagt zu oder nicht. Oder er will handeln – die Entscheidung, ob Sie darauf eingehen wollen, überlasse ich Ihnen. Meine Empfehlung – bleiben Sie bei Ihrem Preis, denn wenn Sie einmal (und das gleich am Anfang) mit dem Preis runter gegangen sind, wird der Kunde immer versuchen, Sie „zu drücken". Wenn Sie richtig kalkuliert haben, ist der Preis, den Sie errechnet haben, der Preis, von dem Sie glauben, dass er Ihrer Dienstleistung gerecht wird. Warum also runtergehen?

Das Pflichtenheft: Im klassischen Projektmanagement erfolgt irgendwo an dieser Stelle (bzw. kurz davor) die Projektinitialisierungsphase (Projektleiter benennen, Projektteam zusammenstellen, Anforderungen genau definieren, Projekt starten). Dazu gehört natürlich auch das Erstellen eines Pflichtenhefts. Es ist auch denkbar, dass das Pflichtenheft Angebots- bzw. Vertragsbestandteil ist. Allerdings bedeutet das für Sie als kleineres Unternehmen oder als Freelancer, dass Sie erheblich in Vorleistung treten müssen (Zeit). Wenn Sie das tun möchten, verabreden Sie mit dem potenziellen Kunden einen Betrag für die Erstellung des Pflichtenhefts, der bei Auftragserteilung verrechnet wird, sonst aber zu zahlen ist. Weiter zum Pflichtenheft. Der Aufbau des Pflichtenhefts sollte folgende Merkmale aufweisen:

- Datum des Stands
- Kurze Projekt- bzw. Softwarebeschreibung
- Namen, Telefonnummern und E-Mail-Adressen der Ansprechpartner auf beiden Seiten
- Verfasser
- Inhaltsangabe
- Alle zu erstellenden Bereiche als Hauptpunkte mit angestrebtem Fertigstellungsdatum
- Alle Unterpunkte zum jeweiligen Hauptpunkt

Für die Form bietet sich wieder eine tabellarische Darstellung an. Ich bin mittlerweile dazu übergegangen, die Reihenfolge der einzelnen Punkte so zu wählen, dass diese gleichzeitig auch der gewählten Priorität (also der Reihenfolge) entspricht. Das ist gerade bei größeren Projekten sehr wichtig, bei denen schon Teile an den Kunden ausgeliefert wurden. Das Pflichtenheft sollte so geschrieben sein, dass zum einen Ihr Kunde versteht, was Sie tun (programmieren) werden und sollte gleichzeitig auch für andere Projektbeteiligte (Entwickler) so aussagekräftig sein, dass diese danach arbeiten (programmieren) können. Eine gute Unterstützung für diese beiden Projektgruppen sind Flow Charts. Bilder sagen oft mehr als viele Worte.

Nun geht wieder das bekannte Ping-Pong-Spiel los. Der Kunde muss das Pflichtenheft bekommen und diesem zustimmen. Lassen Sie sich das schriftlich geben (für die ersten vermeintlichen Diskussionen später). Und dann kann die Arbeit losgehen. Eine Anmerkung noch: Wenn Teile des Projekts (der Applikation) geändert oder wenn Erweiterungen aufgenommen werden, gehören diese natürlich auch in das Pflichtenheft.

Ready to go!

Es kann losgehen. Die Umsetzung des Projekts bzw. die Programmierung startet. Jetzt zeigt sich, ob die Kommunikation mit Ihrem Kunden bis hierhin richtig gelaufen ist, ob Sie die gleiche Sprache sprechen und bis ins Detail von beiden Seiten alles verstanden wurde. Ganz ehrlich – das ist in 90 % der Fälle eher ein Wunsch als eine Tatsache. Aber das verhält sich wie beim Entwickeln von Software: keine Software ist fehlerfrei. Es geht darum, durch geziel-

te Kommunikation mit dem Kunden die Fehler so gering wie möglich zu halten. Wenn Sie den vorgeschlagenen Schritten bis hier gefolgt sind, stehen die Chancen ganz gut, dieses Ziel erreichen zu können.

Ich möchte an dieser Stelle gerne ein kleines Buch von Pascal Mangold ins Rennen werfen: „IT-Projektmanagement kompakt“ [5]. Ich habe das Buch vor ca. 4 Jahren das erste Mal gelesen und war ziemlich begeistert. Ein kurzes Buch mit extrem guten Ratschlägen, wie man ein IT-Projekt gestaltet, was es zu beachten gibt und wie man dieses durchführt. Was ich dabei am meisten mag: Es ist knapp und dabei sehr präzise.

Ein großartiger Punkt im Buch ist die so genannte „Feedbackschleife“ zwischen Kunde und Auftragnehmer während der Entwicklung von Software. Dabei wird ein Teilbereich so weit fertiggestellt, dass der Kunde sich den Bereich ansehen und Feedback dazu geben kann. Man holt den Kunden also während der Entwicklung mit ins Boot. Gibt es aus Sicht des Kunden Probleme oder wurden Funktionalitäten nicht richtig implementiert, können diese schon früh im Entwicklungsstadium entdeckt und gefixt werden (Abb. 1).

Wie Sie an der Grafik sehen, baut somit jeder Abschnitt auf einem anderen auf. Natürlich können viele derer Abschnitte parallel laufen. Wichtig ist nur die ständige Absicherung mit dem Kunden, dass alles O.K. ist. Sie werden damit effizienter vorankommen, und der Kunde wird es später schwer haben, zu behaupten, dass irgendein Bereich nicht richtig implementiert ist. Sie haben es schriftlich, und es gibt keine Diskussionen (der Untertitel dieses Artikels). Übrigens bildet das eine gute Grundlage, um Verzögerungen und kostenpflichtige Erweiterungen zu rechtfertigen. Schließlich werden Sie so Step by Step zum Ziel, dem erfolgreichen Abschluss des Projekts gelangen.

Wichtige Tools

In diesem Abschnitt möchte ich kurz auf die Tools eingehen, die Sie im Projektteam nutzen sollten, um gut und sicher arbeiten zu können. Zuallererst sollten Sie sich mit dem Kunden auf ein Dokumentenformat einigen. Als PDF werden Sie sicher das Angebot und das Pflichtenheft schicken. Allerdings gibt es bestimmt Listen, die Sie und der Kunde bearbeiten wollen. Kann der Kunde OpenOffice Spreadsheets öffnen? Egal, welches Format, es muss eben nur eins sein, das alle Beteiligten lesen können. In diesem Fall rate ich Ihnen übrigens dringend von E-Mails im HTML-Format ab.

Wo werden Featurewünsche und Bugs verfolgt? Wählen Sie eine Bugtracker-Software, die zum einen einfach für den Kunden ist, aber auch genügend Funktionalität für Sie als Entwickler hat. Ich habe im Team sehr gute Erfahrung mit Mantis [6] gemacht. Vor allem weil der Kunde es versteht. Bugzilla [7] finde ich persönlich scheußlich, FogBugz [8] ist sehr gut, aber kostenpflichtig (was natürlich keine Hürde ist). Dann gibt es noch Track [9], Jira [10], arbit project tracking [11] und mingle Agile Project Management [12], die allerdings umfangreiche Projektmanagementtools sind. Welches Tool Sie auch immer nutzen, ist egal. Aber nutzen Sie eins.

Wie wird die Codebasis verwaltet? Ganz einfach: per SVN [13] oder GIT [14]. Darüber brauchen wir nicht länger diskutieren. Wer sein Projekt nicht in einem Versionskontrollsystem verwaltet, hat kein Projekt. Ich nutze SVN und GIT.

Das sind die allerwichtigsten Tools, die Sie benötigen. Auf Coding-Styles, Dokumentation usw. gehe ich aufgrund der Kürze des Artikels nicht weiter ein.

Das Ziel erreicht

Es kann ganz einfach sein. Ich bin der Meinung, die hier im Artikel beschriebenen Punkte und die Vorgehensweise, um ein Projekt in Sachen Kommunikation mit dem Kunden für alle Beteiligten als Erfolg verbuchen zu können, sind einfach, nachvollziehbar und intuitiv. Noch einmal: Es geht hier nicht um klassisches Projektmanagement. Es geht darum, sich klar zu machen, was benötigt wird, um mit dem Kunden eine erfolgreiche Beziehung aufzubauen und Probleme, die durch schlechte Kommunikation entstanden sind, zu vermeiden. Natürlich kann dieser Artikel nur an der Oberfläche kratzen und erhebt bei Weitem nicht den Anspruch auf Vollständigkeit. Er soll einen Anreiz zum Nachdenken geben: Läuft es bei mir (alleine oder in der Firma) nicht auch oft so, wie im beschriebenen Szenario am Anfang? Kann ich nicht doch einzelne Bereiche verbessern? Schreiben Sie mir Ihre Meinung und Erfahrungen. Denn durch Erfahrung lernen wir.

Ich wünsche Ihnen viel Erfolg bei Ihren Projekten!

Featurewünsche und Bugs: Welches Tool auch immer Sie nutzen, ist egal – aber nutzen Sie eins!

Links & Literatur

[1] Nerd: http://de.wikipedia.org/wiki/Nerd

[2] Mindmaps: http://de.wikipedia.org/wiki/Mindmap, http://www.xmind.net

[3] Programmablaufplan: http://de.wikipedia.org/wiki/Programmablaufplan, http://www.omnigroup.com/applications/OmniGraffle/

[4] OpenOffice: http://www.openoffice.org

[5] IT-Projektmanagement kompakt: http://tr.im/N7cv

[6] Mantis Bug Tracker: http://www.mantisbt.org/

[7] Bugzilla: http://www.bugzilla.org/

[8] FogBugz: http://www.fogcreek.com/FogBUGZ/

[9] Trac: http://trac.edgewall.org/

[10] Jira: http://www.atlassian.com/software/jira/

[11] Arbit Project Tracking: http://www.arbitracker.org

[12] Mingle Agile Project Management Tool: http://www.thoughtworks-studios.com/mingle-agile-project-management

[13] SVN: http://subversion.tigris.org/

[14] GitHub: http://github.com

Andreas Wenk

Andreas Wenk ist Softwarearchitekt bei der NMMN in Hamburg. Er betreut unterschiedliche Projekte und spricht täglich mit den Kunden. Organisation und Kommunikation sind zwei seiner Steckenpferde. Sie erreichen ihn unter andy@nms.de und können ihm unter @awenkhh folgen.