

SOFTWARE BRAUCHT „SOFT SKILLS“

Was sollte ein Softwareentwickler oder -architekt eigentlich können? Aktuelle Technologien und technische Konzepte sollte er beherrschen. Aber ist das wirklich ausreichend? Nein, denn Softwareentwicklung braucht mehr als harte Technik – sie braucht zunehmend scheinbar weiche Kompetenzen, so genannte „Soft Skills“. Das liegt in ihrer Natur, wird allerdings häufig verdrängt. Dieser Artikel zeigt, warum Qualitäts- und Produktivitätseinbußen die Folge sein müssen, wenn Soft Skills zu wenig Aufmerksamkeit geschenkt wird.

Softwareentwickler sollen programmieren. Sie sollen die Technologien und Konzepte ihres Metiers verstehen – das ist das Wichtigste. So denken zumindest immer noch viele Geschäftsführer, Abteilungsleiter und Teamchefs. Daran ändern auch häufig geäußerte Forderungen nach *Soft Skills* (siehe Kasten 1) wie „Kreativer Arbeitsstil mit erstklassigem Zeitmanagement“ oder „Sie sind in der Lage, Ihr Arbeitsumfeld eigeninitiativ auf vortreffliche Weise zu organisieren“ oder „Sie besitzen überdurchschnittliche Kommunikationsfähigkeit“ in Stellenanzeigen für Softwareentwickler nichts¹⁾. Es sind vielmehr – so muss leider konstatiert werden – oft Alibifloskeln, gebraucht aus einem einzigen Grund: weil das heute so üblich ist. Der deutsche, zu „Soft Skills“ nicht ganz deckungsgleiche Begriff „Schlüsselqualifikationen“ ist einfach seit seiner Prägung Mitte der 70er Jahre im Bewusstsein der Arbeitgeber und damit fester Bestandteil der Anforderungen an neue Mitarbeiter.

Soft-Skill-Forderungen als Alibifloskeln

Dass *Soft Skills* allerdings oft nur als Alibifloskeln gebraucht werden, ist leicht an drei Kriterien festzumachen:

- Sofern ein Fortbildungsprogramm im ausschreibenden Unternehmen existiert, enthält es in der Regel allerlei Maßnahmen zur Erhöhung technischer/fachlicher Kompetenzen (*Hard Skills*), aber keine bzw. nur marginale zur (Weiter-)Entwicklung von *Soft Skills*.
- Auf Nachfrage kann nicht in einfachen klaren Worten erklärt werden, wie die Kompetenz eines Bewerbers im Hinblick auf die geforderten *Soft Skills* gemessen werden sollen. Was sind z.B. Kennzeichen für „erstklassiges Zeitmanagement“? Wie ist ein „auf vortreffliche Weise“ organisiertes Arbeitsumfeld zu erkennen?
- Es existiert keine klare Vorstellung davon, wo und wann *Soft Skills* von Bewerbern überhaupt erworben worden sein könnten. Lernt man einen „kreativen Arbeitsstil“ während der Fachinformatiker-Ausbildung? Ist es die

¹⁾Die Formulierungen entstammen durchschnittlichen Online-Ausschreibungen für Programmiererpositionen.

▶ die autoren



Ralf Westphal
(E-Mail: ralfw@ralfw.de)
ist für die .NET-Plattform Autor, Berater, Trainer und Sprecher auf vielen Entwicklerveranstaltungen. Darüber hinaus ist er „Visual Developer Solution Architect“ MVP.



Renate Klein
(E-Mail: renate@klein-personaltraining.de) hält Seminare, Workshops und Vorträge und arbeitet als Coach zu den Themengebieten Leadership, Rhetorik, Power Verkauf, Teamentwicklung.

Aufgabe eines Universitätsstudiums, „überdurchschnittliche Kommunikationsfähigkeit“ zu vermitteln?

Welchen anderen Schluss lassen unklare Vorstellungen von *Soft Skills* und Ignoranz in Bezug auf Aus- und Weiterbildung zu? Forderungen nach *Soft Skills* sind vielfach leider gerade in kleineren Unternehmen ohne gezieltes Personalmanagement Lippenbekenntnisse.

Und wie sollte es auch anders sein? Den Fordernden ist womöglich nicht einmal ein Vorwurf zu machen, allemal, wenn Sie keine ausgebildeten „Personaler“ sind. Wo hätten Sie denn etwas über die Wichtigkeit von Schlüsselqualifikationen lernen können? Wo hätten Sie denn selbst *Soft Skills* erwerben können? In [BfA] findet sich zwar die folgende Formulierung: „Die Bundesvereinigung der Deutschen Arbeitgeberverbände beschreibt den idealen Hochschulabsolventen so: 51% seiner Fähigkeiten sind fachlicher Art, 49% sind Schlüsselqualifikationen.“ Aber in den Curricula der Ausbildungsinstitute von Berufsschule bis Universität drückt sich das nicht aus. Beispielhaft seien der Bachelor-Studiengang Informatik der Universität Hamburg, der Studiengang Angewandte Informatik der Berufsakademie Karlsruhe und eine Fachinformatikerausbildung herausgegriffen:

- An der Universität Hamburg lässt sich ein Bachelor der Informatik in einem dreijährigen Studium erwerben, in dem nur 12,5% des Lehrplans *Soft Skills* gewidmet sind.
- In Karlsruhe lässt sich der Dipl. Inf. (BA) in einem dreijährigen Studium erwerben, von dem nur 3,75% auf *Soft Skills* entfallen.

Es ist also kein Wunder, wenn Wunsch und Realität weit auseinander klaffen: Unternehmen wünschen sich Mitarbeiter mit *Soft Skills*, die den *Hard Skills* im Grunde in nichts nachstehen, doch niemand weiß, wo sie erworben werden könnten. In den üblichen Aus- und Fortbildungsinstitutionen der Branche stehen sie nicht auf dem Programm (sic!). Vereinzelt gibt es zwar Angebote, z. B. bei der oose GmbH das Training „Kommunikation und Moderation in IT-Projekten für Entwickler und Analytiker“ (vgl. [oos06]) oder bei der

Als *Soft Skills* werden solche Kompetenzen und Persönlichkeitszüge bezeichnet, die zur Ausübung eines Berufes nötig oder hilfreich sind, aber eben nichts direkt mit Fachdomäne und ihren Techniken zu tun haben, die durch so genannte *Hard Skills* definiert werden.

In Stellenausschreibungen häufig geforderte *Soft Skills* sind Teamfähigkeit, Kommunikationsfähigkeit, Selbstständigkeit, Belastbarkeit, Führungskompetenz oder Kreativität. Die Literatur kennt jedoch Dutzende weitere (vgl. [Mor], [Wik]), die sie in unterschiedliche Kategorien

- soziale Kompetenzen
- kommunikative Kompetenzen
- personale Kompetenzen
- Führungskompetenzen
- Umsetzungskompetenzen
- mentale Kompetenzen

entlang mehreren Dimensionen anordnet (vgl. [Wik]), wie z. B.

- erlernbar/kaum oder nicht erlernbar
- äußerlich/innerlich
- beruflich/privat verwertbar
- global bzw. interkulturell/regional bzw. lokal

Nach einer DIHK-Studie zu Erwartungen von Unternehmen an Hochschulabsolventen, in der 2.154 Unternehmen befragt wurden (vgl. [DIHK]), sind *Soft Skills* „heute entscheidender denn je“ beim Einstieg in den Beruf.

Kasten 1: Was sind „Soft Skills“?

Professional Developer College GbR das „Rhetoriktraining für Softwareentwickler“ für erfolgreiche technische Präsentationen (vgl. [Pro]). Aber das sind punktuelle Vorstöße, die (noch) keine breite Akzeptanz gefunden haben.

Aber vielleicht erwarten Unternehmen ja, dass *Soft Skills* außerhalb der Ausbildung erworben werden. Setzen Sie eventuell darauf, dass ein neuer Mitarbeiter einfach schon beim vorherigen Arbeitgeber „erstklassiges Zeitmanagement“ gelernt hat? Das ist aber eher unwahrscheinlich, denn Fortbildung wird branchenweit klein geschrieben. Oder sind *Soft Skills* so grundlegend, dass darauf gehofft werden darf, sie würden während der Kindergarten- oder Schulzeit erworben? Auch das trifft nicht zu, wie die Ergebnisse der Pisa-Studie, jedermanns eigene Erfahrungen und die Lehrpläne an deutschen Schulen unmissverständlich klar machen sollten. Was bleibt noch? Eigentlich kann daher nur die Familie der Ort sein, an dem Schlüsselqualifikationen quasi mit der Muttermilch eingeflößt werden. Angesichts steigender Scheidungsraten – die sicherlich negativ korrelieren mit *Soft Skill* Konfliktfähigkeit – und des Erfolgs der „RTL Super Nanny“ besteht auch hier kaum Hoffnung.

Es kann also nur konstatiert werden, dass *Soft Skills* zu den *Hard Skills* zwar als fast gleichwertig erachtet werden (vgl. [DIHK], [BfA]), dass sich aber niemand für eine dieser Wertschätzung angemessene Vermittlung verantwortlich fühlt. Das lässt nur zwei Schlüsse zu:

- Sind *Soft Skills* doch nicht so wichtig, wie allgemein behauptet? Dann lohnt es sich nicht, Schulungsaufwand in sie zu investieren.

- Oder lassen sich *Soft Skills* ganz einfach, quasi im Vorbeigehen erwerben, sodass sie keiner expliziten Ausbildung bedürfen?

Auch wenn es hier und da Vertreter der Position geben mag, *Soft Skills* seien so weich, dass ihr Erwerb jedem mühelos in der Freizeit wie von selbst gelingen sollte, so kann doch angenommen werden, dass diese Meinung nicht wirklich vorherrscht. Die schlichte Forderung nach ihrem Besitz in Stellenausschreibungen spricht dagegen. Denn was explizit gefordert wird, ist gewöhnlich eben keine Selbstverständlichkeit und somit auch nicht im Vorbeigehen zu erwerben.

Es gibt also nur eine Erklärung für die eklatante Unterrepräsentation von *Soft Skills* in den Aus- und Weiterbildungslehrplänen der Softwarebranche sowie auf dem Fortbildungsradar der Unternehmen: *Soft Skills* sind am Ende doch nicht so wichtig für Softwareentwickler. DeMarco drückt diesen Glauben prägnant so aus: „Ein Programmierer braucht eben weniger Einfühlungsvermögen [...]“ (vgl. [DeM01]). Was zählt, sind also harte technisch-fachliche Kompetenzen; echte Programmierer verstehen etwas von Objektorientierung, Applikations-Servern, WSDL und dem „Impedance Mismatch“ beim RDBMS-Zugriff. Alles andere ist nebensächlich oder ergibt sich schon. Man hat es – die Teamfähigkeit, die Selbstorganisation, die Konfliktkompetenz, das systemische Denken – oder man hat es eben nicht. Und wenn nicht, dann ist das auch nicht so schlimm. Es soll ja schließlich Software produziert und kein Beziehungsringelreihen aufgeführt werden. Oder?

Das ist nicht nur eine naive, sondern sogar eine erfolgsfeindliche Haltung. Sie läuft den Anforderungen an die heutige Produktion von Software zuwider.

Softwareentwicklung ist Wissensarbeit

Warum ist Gleichgültigkeit den *Soft Skills* gegenüber erfolgsfeindlich? Weil sie nicht der Natur der Tätigkeit „Softwareentwicklung“ entspricht. Die Fehleinschätzung des Stellenwerts von *Soft Skills* ist das Resultat einer überkommenen Vorstellung, wie Programmieren „funktioniert“. Diese Vorstellung wurzelt in der Mitte des letzten Jahrhunderts und in klassischen Ingenieursdisziplinen. Sie ist das Relikt einer Zeit, in der Wissen zentral in den Universitäten „produziert“ wurde, Ingenieure nach klaren Anforderungen Software entworfen und dann auch noch implementiert haben und Arbeit in hierarchischen Strukturen organisiert war.

Natürlich war auch damals die Programmierung schon etwas anderes als die Schuhfabrikation oder das Bäckerhandwerk. Sie war nicht nur wissensbasiert, sondern sogar wissensintensiv. Auch hat sich Softwareentwicklung immer schon von anderen wissensintensiven Arbeiten wie dem Maschinenbau oder der Elektrotechnik unterschieden. Der hohe Abstraktionsgrad – Werkzeuge und Materialien sind nicht wirklich greifbar – hat sie von Anfang an in die Nähe der Mathematik gerückt. Dennoch haftet der Softwareentwicklung der Geruch des Handwerks bzw. traditioneller Industrien an.

Historisch betrachtet ist das auch durchaus verständlich. Letztlich müssen bzw. mussten auch andere Branchen um ihre „Entstaubung“ kämpfen und sich neu erfinden. „Lean

Production” in der Automobilindustrie ist hier nur ein Stichwort. Und auch die Softwareentwicklung hat sich von überkommenen Vorstellungen befreit und ist agil geworden.

Andererseits ist es aber schon verwunderlich, dass *Soft Skills* immer noch so einen geringen Stellenwert haben, denn die Frage, ob denn Programmierung überhaupt eine Ingenieursdisziplin sei oder nicht vielmehr eine Kunst, ist ja sehr alt und keineswegs allgemein akzeptiert entschieden. Das heißt, allemal die Kreativität, die jeder Künstler für sich reklamiert, sollte ganz oben nicht nur auf der Anforderungsliste aller Stellenausschreibungen stehen, sondern auch ihre allgegenwärtige Förderung – sei es durch Schaffung eines inspirierenden Arbeitsumfeldes oder explizite Fortbildung. Doch selbst das ist nicht der Fall. Die vorherrschenden Arbeitsbedingungen in Softwareprojekten lassen sich eher mit dem Attribut „kreativitätstötend“ versehen. Der schiefe Mangel an Spielräumen (vgl. [DeM01]), die Kreativität braucht, um sich zu entfalten, ist dafür die erste Ursache. Wo „Experiment“, „Fehler“ und „Lernen“ nicht zum Vokabular gehören oder negativ besetzt sind, weil sie keinen unmittelbaren Kundennutzen produzieren, da kann die Kreativität, die Künstler doch brauchen, nicht blühen.

Aber vielleicht sind Softwareentwickler ja auch keine Künstler, sondern Ingenieure. Dann müssen sie auch nicht so kreativ sein, oder? Jeder Elektrotechniker oder Maschinenbauer würde das sicherlich zurückweisen und Kreativität für sich reklamieren. Aber letztlich klingt Ingenieur immer noch „handfester“ als Softwareentwickler. Nicht zuletzt ist dafür eine außergewöhnlich heterogene Ausbildungslandschaft verantwortlich, denn obwohl „echte“ Informatiker, also Softwareentwickler mit Universitätsabschluss, hoch im Kurs stehen, gibt es doch kein einheitliches Berufsbild und der Hintergrund professioneller Softwareentwickler ist breit gefächert. Vom Autodidakten und Umschüler über den Fachinformatiker und Berufsakademieabsolventen bis zum Dipl.-Inf. reicht das Spektrum. Pointiert ließe sich vielleicht sagen: Wenn früher arbeitslose Philologen Taxifahrer wurden, dann werden sie heute Programmierer. Zehnder spricht sogar davon, dass Informatikberufe von Quereinsteigern dominiert würden (vgl. [Zeh03]).

Letztlich ist es jedoch für das Verständnis der Wichtigkeit von *Soft Skills* egal, ob Programmierung nun Ingenieurstätigkeit oder Kunst oder Ingenieurskunst ist. Auch der Ausbildungshintergrund der Softwareentwickler ist unerheblich. Vielmehr ist es entscheidend zu erkennen, wie Softwareentwicklung hier und heute funktioniert. Und das lässt sich in einem Satz zusammenfassen: *Softwareentwicklung ist Wissensarbeit*. Sie ist nicht nur wissensbasiert und wissensintensiv, sondern ist Wissensarbeit. Das heißt, Wissen wird

- „kontinuierlich revidiert,
- permanent als verbesserungsfähig angesehen,
- prinzipiell nicht als Wahrheit sondern als Ressource betrachtet [...] und
- untrennbar mit Nichtwissen gekoppelt [...], sodass mit Wissensarbeit spezifische Risiken verbunden sind“ (vgl. [Wil98])

Vor dem Hintergrund des hohen Abstraktionsgrades von Software liegt der Grund für ihre Form der Wissensarbeit in

einem fundamentalen Erkenntnisproblem: Für die Softwareentwicklung gibt es keine absolute, erkennbare Wirklichkeit. Es gibt keine absoluten Anforderungen des Kunden, die mit etwas Aufwand erkannt werden können. Es gibt keinen absoluten Markt der Wettbewerber, der einfach nur analysiert werden muss. Es gibt keine absolute Menge von Technologien oder Konzepten, die einmal erlernt werden müssen. Kunde, Markt, Werkzeuge und Materialien sind vielmehr in ständiger Bewegung und die Wahrnehmungsorgane einzelner Entwickler oder Projektteams sind unvollkommen.

Zu keinem Zeitpunkt kann ein Programmierer ausrufen „Jetzt weiß ich genau, was der Kunde will!“ oder „Jetzt nehme ich meinen Werkzeugkasten in- und auswendig!“. Ganz im Sinne des Konstruktivismus (vgl. [Wat95]) kann jeder Einzelne und jedes Team immer nur ein mehr oder weniger passendes Modell von den Anforderungen, vom Markt oder von den technischen Möglichkeiten haben. Statt eines „Jetzt weiß ich es aber“ gibt es nur ein „Bisher bin ich noch an keine Grenze meines Modells/Wissens gestoßen“.

Wissensarbeit, d.h. Softwareentwicklung, ist also Arbeit mit dem und am Vorläufigen. Das bedeutet, sie muss sich ständig vergewissern, ob sie noch „auf Kurs“ ist bzw. ob ihre Modelle noch passend sind. Ihr geht es nicht um Wahrheit, sondern um Viabilität. Und das bedeutet, Softwareentwicklung muss ständig handeln und wahrnehmen, was in ihrer Umwelt „einfach so“ passiert, und wahrnehmen, was ihre Handlungen bewirken.

Soft Skills sind das Fundament der Softwareentwicklung

Wissensarbeit ist damit unablässige Interaktion. Interaktion aber ist fach- und technikenunabhängig. Sie ist vielmehr die zentrale Kunst, zu deren Beherrschung *Soft Skills* nötig sind. Die Beherrschung von *Soft Skills* definiert geradezu die Interaktionsfähigkeit eines Softwareentwicklers bzw. eines Teams. Damit sind *Soft Skills* das Fundament jeder zeitgemäßen Softwareentwicklung. Sie bilden den Rahmen für die Programmierung und sind ihr vorgelagert. Ohne *Soft Skills* keine Softwareentwicklung, denn ohne *Soft Skills* kein Wissenserwerb. Und wo Wissen schon vorhanden ist, wird es ohne *Soft Skills* schal oder kann sich nicht voll entfalten.

Den Programmierer, der in einem klar abgesteckten Rahmen auf der Basis seines (einmal) erworbenen Wissens einfach kreativ klare Anforderungen umsetzt, gibt es nicht mehr – wenn es ihn denn je gegeben hat. Der Programmierer als Einzelkämpfer ist tot. Softwareentwicklung ist eine Teamanstrengung. Software entsteht im Dialog: im Dialog mit dem Kunden, den Anwendern, den Kollegen, dem Vorgesetzten, den Mitarbeitern, den Technologieanbietern.

Soft Skills können insofern auch als Kooperationskompetenzen aufgefasst werden. Sie machen die Interaktionen der Wissensarbeit geschmeidig im Hinblick auf die gemeinsame Erreichung eines Zieles.

Soft Skills im Einsatz

Software ist das Ergebnis von Kooperation, Interaktion und Dialog. Die Kommunikationsachsen sind vor allem:

- Kunde-Softwareteammitglied
- Teammitglied-Teammitglied (*Peer-to-Peer*)
- Teammitglied-Vorgesetzter (Hierarchie)
- Teammitglied-Technologieanbieter
- Teammitglied-Markt

Entlang diesen Achsen muss kommuniziert und geführt werden. Die Endpunkte der Achsen müssen Entscheidungen umsetzen können und personale wie mentale Kompetenzen haben.

Wo früher vor allem nach IQ für wissensintensive Arbeit gefragt wurde, sind heute für die Wissensarbeit genauso *EQ* (*Emotionale Kompetenz*, dazu gehören Empathie oder Menschenkenntnis), *SQ* (*Sinnorientierung*, d. h. Sinnstiftung, Motivierungsvermögen) und *PQ* (*Philosophische Kompetenz*, dazu gehören kritisch-systemisches Denken oder der Umgang mit Unbestimmtheit) wichtig.

Wissensarbeit geht nicht mehr von einer zentralen Wissensproduktion und (einmaligen) Vermittlung von oben nach unten aus, sondern von der ständigen Wissensschöpfung in zwischenmenschlicher Interaktion. Im Kern der Softwareentwicklung steht daher – bei all ihrer Wichtigkeit – nicht die Beherrschung von Tools und Technologien, sondern der Umgang von Menschen mit Menschen.

Kein Werkzeug (Entwicklungsumgebung, Profiler, RDBMS) kann bedient, kein Material (API, Daten) verarbeitet werden, ohne dass nicht zuerst ein *Soft Skill* zum Einsatz gekommen ist. Am Anfang stehen Lese- und Lernkompetenzen. Ohne sie gibt es keinen Wissenserwerb und keine Werkzeug- und Technologiekompetenz.

Kurz darauf folgen Stressbewältigungsvermögen, Motivierungsvermögen, Zeit- und Selbstmanagement. Ohne sie kann keine Arbeit stattfinden, auch nicht allein.

Dann kommen Selbstvermarktungsfähigkeit, Überzeugungsvermögen, Verhandlungsgeschick, Präsentations- und Rhetorikkompetenz. Ohne sie gibt es keinen Job und keinen Verkauf von Ideen und Produkten.

Und schließlich folgen Empathie, Moderationskompetenz, Menschenkenntnis, Delegationskompetenz, systemisches Denken sowie Konflikt- und Kritikfähigkeit. Ohne sie gibt es keine Zusammenarbeit mit Kollegen, Vorgesetzten und Kunden.

Soft Skills sind also die Grundlage aller fachlich-technischen Arbeit. Sie fangen nicht erst beim Verkauf, Projektmanagement oder expliziten Führungsaufgaben an, sondern durchziehen die gesamte berufliche und private Tätigkeit.

Komplexe Systeme wie heutige Software lassen sich nicht in Isolation von wenigen nach Wasserfall-Manier realisieren. Sie bedürfen des ständigen Austauschs vieler Beteiligten. Sie sind das Ergebnis von Wissensarbeit. Und damit hängt ihre Qualität vom Grad der Kooperationskompetenzen der Beteiligten ab. Ist der gering, ist der Misserfolg vorprogrammiert. Inwiefern sich also die notorische Softwarekrise und der immer wieder festgestellte hohe Prozentsatz fehlgeschlagener Softwareprojekte auf einen Mangel an *Soft Skills* zurückführen lassen, ist daher eine interessante Frage.

Fazit

Wenn nun nicht von der Hand zu weisen ist, dass *Soft Skills* so essenziell für die Softwareentwicklung sind, dann sollte

auch auf der Hand liegen, dass ihr Erwerb keine Nebensache sein kann. Der Stellenwert von *Soft Skills* in Aus- und Weiterbildung muss deutlich angehoben werden. *Soft Skills* müssen in der Fortbildung gleichberechtigt neben *Hard Skills* stehen. Sie bedürfen eines Entwicklungsplans wie alle fachlich-technischen Kompetenzen. Überhaupt muss der ständige Kompetenzausbau, das arbeitslebenslange Lernen, eine höhere Priorität bekommen bzw. überhaupt auf den Radarschirm genommen werden.

Eine Sensibilität dafür kann die Lektüre von [DeM01], [Vig06] oder [Mor06] stärken. Letztlich kommt es aber darauf an, dass jeder einzelne Entwickler, jeder Softwarearchitekt, Projektleiter, Abteilungschef und Unternehmer in der Branche die Entwicklung von *Soft Skills* plant und auch umsetzt. *Soft Skills* können sich nicht (nur) aufgrund der Initiative einzelner zum Nutzen des Ganzen entfalten. Die Qualität von Softwareprodukten steigt nicht, wenn ein Entwickler an seiner Kommunikationskompetenz feilt. *Soft Skills* entfalten ihren Nutzen nur, wenn sie integraler Bestandteil der Unternehmensstrategie sind. *Soft Skills* müssen gelebt werden: vom Unternehmenslenker bis hinunter zum Programmierer.

Die Qualität von Software ist nicht vordringlich vom Einsatz neuester Tools und Technologien abhängig, sondern von den Menschen, die sie entwickeln. Und Menschen sind weit mehr als ihre Fachkompetenz. In Anlehnung an Hartmut von Hentig, einem Vordenker moderner Pädagogik, gilt daher: Die wichtigste Ressource eines Entwicklers ist seine Persönlichkeit. Wer in sie nicht investiert, riskiert Qualitäts- und Erfolgseinbußen. ■

Literatur & Links

[BfA] Bundesagentur für Arbeit, Schlüsselqualifikationen, siehe: www.wege-ins-studium.de/qualifikationen.html

[DeM01] T. DeMarco, Spielräume. Projektmanagement jenseits von Burn-out, Stress und Effizienzwahn, Hanser Fachbuch 2001

[DIHK] DIHK, Fachwissen ist nicht alles – Persönlichkeit ist gefragt, DIHK-Studie zu Erwartungen von Unternehmen an Hochschulabsolventen, siehe: www.dihk.de/inhalt/informationen/news/meldungen/meldung_005580.main.html

[Mor] A. Moritz, Soft Skills – Seminare Training Coaching, siehe: www.soft-skills.com/

[Mor06] A. Moritz, Felix Rimbach, Soft Skills für Young Professionals, Gabal 2006

[oos06] oose Innovative Informatik GmbH, [SKM] Kommunikation und Moderation in IT-Projekten für Entwickler und Analytiker, siehe: www.oose.de/Soft_Skills_SKM.htm

[Pro] Professional Developer College GbR, Rhetoriktraining für Softwareentwickler, www.prodevcollege.de/produkte-sprecher/training.html

[Vig06] U. Vigerschow, B. Schneider, Soft Skills für Softwareentwickler, dpunkt 2006

[Wat95] P. Watzlawick, Wie wirklich ist die Wirklichkeit?, Piper 1995

[Wik] Wikipedia, Soft Skills, siehe: de.wikipedia.org/wiki/Soft_skills

[Wil98] H. Wilke et al., Wissensarbeit in intelligenten Organisationen, 1998, siehe: www.uni-bielefeld.de/presse/fomag/uni1B_pdf/wilke_s20_24.pdf

[Zeh03] C.A. Zehnder, Wer sind denn diese Informatiker?, 2003, siehe: www.inf.ethz.ch/personal/zehnder/informatikerwerInfkCAZ-04.pdf