

Jenseits des Tellerrands: IT-Berufsethos oder Verantwortung in der Softwareentwicklung

# In der Supervision

■ VON HENNING WOLF UND STEFAN ROOCK



Softwareentwicklung ist weit mehr als Programmierung, und Programmierung ist weit mehr als das bloße Beherrschen von Programmiersprachen und Werkzeugen. Die Artikel dieser Serie greifen wichtige Themen, aktuelle Fragen und auch grundsätzliche Probleme auf, beleuchten diese und stellen „die gängigen Antworten“ immer wieder in Frage. Subjektivität ist dabei keineswegs verpönt, sondern das notwendige Salz in der technischen Einheitssuppe.

Als Entwickler ziehen wir uns gerne auf den neutralen Standpunkt zurück: Das Management entscheidet, und wir setzen um. Wir sind lediglich für die korrekte Realisierung der Managemententscheidungen verantwortlich, bei den restlichen Folgen waschen wir unsere Hände in Unschuld. Doch so einfach lassen uns die Autoren des folgenden Artikels nicht entkommen.



Johannes Link  
(john.link@gmx.net)

Welche Vorstellung von seiner Rolle als Softwareentwickler hat eigentlich ein Softwareentwickler? Welche Vorstellung hat ein IT-Manager? Und was halten die voneinander? Um uns diesen Fragen zu nähern, lauschen wir mal bei zwei Supervisionsgesprächen.

## Der IT-Manager bei der Supervision

IT-Manager: „Meine Entwickler treiben mich noch in den Wahnsinn. Sie halten Termine nicht ein und liefern Software, die nur aus Fehlern zu bestehen scheint.“

Supervisorin: „Was haben Sie getan, dass die Softwareentwickler das tun?“

IT-Manager: „Was ich getan habe? Nichts! Ich bin doch nicht das Problem. Die Softwareentwickler weigern sich einfach, richtig zu arbeiten. Eigentlich müsste ich meine Entwickler herschicken, damit die mal lernen, wie man sich richtig verhält.“

Supervisorin: „Wie verhält man sich denn Ihrer Meinung nach richtig?“

IT-Manager: „Es kann doch nicht angehen, dass die mir erst eine Woche vor der Projekt-Deadline sagen, dass sie noch mindestens vier weitere Wochen Zeit brauchen!“

Supervisorin: „Haben Sie das Ihren Entwicklern so auch schon mal gesagt?“

IT-Manager: „Nein, aber das ist doch wohl klar!“

Supervisorin: „Wer setzt denn die Termine?“

## Jenseits des Tellerrands

- Teil 1: Das nächste Java?
- Teil 2: Typisierung in Java und darüber hinaus
- Teil 3: Warum gibt es eigentlich Softwareentwicklungsprojekte?
- Teil 4: Softwarearchitektur: eine kritische Bestandsaufnahme
- Teil 5: Textuelle domänenspezifische Sprachen
- Teil 6: Moderne System- und Abnahmetests
- Teil 7: Warum nutzen wir die Mittel zur Softwareverbesserung nicht, obwohl wir sie kennen? Eine Diskussion
- Teil 8: Java-Bashing: Unsere liebste Plattform ist zu kompliziert geworden
- Teil 9: Rich Client vs. Web Client
- Teil 10: Wie viel Abstraktion brauchen wir?
- Teil 11: Berufsethos, Qualität, Rollenverständnis
- Teil 12: Time's Arrow: ein Projekt von hinten

IT-Manager: „Ich setze die Termine gemeinsam mit den Kunden.“

Supervisorin: „Erklären sich denn die Entwickler explizit einverstanden mit den Terminen, und sind die Termine für die Entwickler realistisch?“

IT-Manager: „Ach, Jammern gehört doch bei Entwicklern zum Geschäft! Wenn die nicht immer unter Strom gehalten werden, dann bauen die bloß Frameworks, die keiner braucht. Die halten alle meine Schätzungen für unrealistisch. Ich habe noch nie erlebt, dass ein Entwickler mit einer Aufgabe eher fertig gewesen wäre.“

Supervisorin: „Es ist kaum verwunderlich, dass Ihre Entwickler sich nicht einlassen können auf Schätzungen, die sie nicht selbst vorgenommen haben. Wie sollen sie denn Verantwortung für etwas übernehmen, das sie im Detail noch nicht kennen?“

IT-Manager: „Aber das muss ich doch auch! Ich verhandle nach Erfahrungswerten einen Aufwand und einen Preis mit dem Kunden, und dafür übernehme ich dann auch die Verantwortung. Da ist es doch nicht zu viel verlangt, wenn auch die Entwickler ihren Teil der Verantwortung tragen.“

Supervisorin: „Hier liegt vielleicht genau das Problem: Der Aufwand, den Sie für die Preisermittlung aus Erfahrungswerten heranziehen, muss ja eben nicht direkt etwas mit dem Aufwand zu tun haben, den Ihre Entwickler brauchen, oder?“

IT-Manager: „Nee, hat er auch nicht. Leider. Wenn ich Entwickler sowas schätzen lasse, dann brauchen die ganz lange dafür, die verlieren sich in Details, und schon sind die Aufwände hoch, meist höher als der Markt hergibt.“

## Supervision

Supervision ist eine Coachingform, die die Qualität beruflicher Arbeit verbessern soll. Dazu werden Einzel- oder Gruppensitzungen vom Supervisor (meist mit psychologischer Ausbildung) angeleitet, in denen häufig die Interaktion und Kommunikation mit Kunden, Vorgesetzten, Mitarbeitern oder Kollegen thematisiert wird. Der Supervisor unterstützt den Supervisanden bei dessen persönlicher Weiterentwicklung, um in zukünftigen Interaktions- und Kommunikationssituationen bessere Ergebnisse zu erzielen.

Supervisorin: „Aber die unterschiedlichen Granularitätsebenen dürften Sie doch nicht verwundern: Sie sind Manager, die Entwickler arbeiten doch auf einer anderen Detailebene als Sie! Wie gut sind denn die Entwicklerschätzungen auf der Detailebene?“

IT-Manager: „Ja, im Detail sind Entwickler ziemlich verlässlich. Je kleiner die Anforderung, umso verlässlicher. Aber eigentlich immer relativ konstant um 50 Prozent zu niedrig in der Schätzung im Verhältnis zum tatsächlichen Aufwand.“

Supervisorin: „Prima, damit müsste sich doch arbeiten lassen. Dann lassen Sie doch möglichst kleine Anforderungen schätzen und verdoppeln das Ergebnis.“

IT-Manager: „Tja, da muss ich mal drüber nachdenken ...“

Supervisorin: „Was brauchen denn die Entwickler, um für ihre Arbeit Verantwortung übernehmen zu können? Was meinen Sie?“

IT-Manager: „Klare Ansagen, klare Anforderungen, klare Rahmenbedingungen.“

Supervisorin: „Und wer ist verantwortlich, diese Klarheit herzustellen?“

IT-Manager: „Das bin ich, das ist mein Job. Das mache ich ja auch im Großen und Ganzen.“

Supervisorin: „Bedeutet im ‚Großen und Ganzen‘, dass Sie meinen, man könnte das noch besser machen?“

IT-Manager: „Sicherlich, ich könnte das wohl noch besser machen.“

Supervisorin: „Was hindert Sie daran, es besser zu machen?“

IT-Manager: „Ich weiß nicht genau, warum die Entwickler mit den für mich klaren Rahmenbedingungen nicht zurechtkommen.“

Supervisorin: „Dann fragen Sie doch erstmal Ihre Entwickler, was die denn genau brauchen, um sich verlässlich und verantwortlich auf gemeinsame Termine oder Ziele einzulassen.“

## Szenenwechsel: Der Softwareentwickler bei der Supervision

Entwickler: „Mein Chef ist irre. Er setzt uns utopisch knappe Deadlines und zwingt uns dadurch, schlechte Qualität abzuliefern. Später regt er sich auf, dass die Software so viele Fehler enthält und die Wartungskosten explodieren.“

Supervisor: „Wie macht Ihr Chef es denn, dass Sie diese unhaltbaren Termine für sich akzeptieren?“

Entwickler: „Er zwingt mich dazu. Er sagt, bis zum 1. August *muss* die Software unbedingt fertig sein.“

Supervisor: „Und sie beginnen dann mit der Programmierung?“

Entwickler: „Genau. Der Chef wird dann schon sehen, was er davon hat.“

Supervisor: „Hm. Aber Sie haben doch jetzt den Stress.“

Entwickler: „Ja, weil der Chef einfach nicht versteht, vor welchen Schwierigkeiten wir als Entwickler stehen. Wenn er sich ändern würde, das wäre schön ...“ (Entwickler guckt verträumt) „Aber das werde ich wohl leider nicht mehr erleben.“

Supervisor: „Wahrscheinlich nicht. Er wird sich nicht von selbst ändern, und Sie können ihn auch nicht ändern. Sie können nur eines tun: Ihr eigenes Verhalten ändern. Was könnten Sie persönlich also tun, damit Sie nicht wieder in so eine Situation kommen?“

Entwickler: „Kündigen.“

Supervisor: „In anderen Firmen kommt sowas nicht vor?“

Entwickler: „Leider doch, die Chefs scheinen alle irgendwie grenzdebil zu sein.“

Supervisor: „Was können Sie also tun?“

Entwickler: „Ich sage ja schon immer, dass das so nicht zu schaffen ist mit den Terminen.“

Supervisor: „Wie sagen Sie das denn genau? Ich bin jetzt mal Ihr Chef und sage: ‚Am 1. August *muss* die Software fertig sein!‘“

Entwickler: „Das ist kaum zu schaffen, Chef.“

Supervisor: „Sie sagen ‚kaum‘? Wissen Sie, was Ihr Chef dann verstanden hat? Vermutlich, dass es zu schaffen ist, wenn auch anstrengend. Warum sagen Sie nicht, dass es *nicht* zu schaffen ist.“

Entwickler: „Das will der Chef ja nicht hören. Manchmal kommt er rein und sagt: ‚Was gibt es Neues? Aber bitte nur gute Nachrichten!‘“

Supervisor: „Wie fühlt sich so ein Spruch für Sie an?“

Entwickler: „Hm, ich werde nicht ernst genommen.“

Supervisor: „Wenn Ihr Chef Sie nicht ernst nimmt, dann nehmen Sie sich doch

bitte jedenfalls selbst ernst. Ich spiele noch mal Ihren Chef: „Am 1. August *muss* die Software fertig sein!“

Entwickler: „Chef, das ist unmöglich. Das kann ich nicht schaffen.“

Supervisor (als Chef): „Na, kommen Sie. Letztes Mal haben Sie es doch auch hingekriegt. Das sieht jetzt schlimmer aus, als es ist.“

Entwickler: „Nein, ich bin mir sicher, dass ich es nicht schaffen kann. Und letztes Mal haben wir das bezahlt mit ziemlich vielen Fehlern in der Software.“

Supervisor (als Chef): „Was schlagen Sie denn vor?“

Entwickler: „Wir können die wichtigsten Funktionen bis zum 1. August fertig stellen und die dann auch als erste Ausbaustufe des Systems ausliefern. Weitere Ausbaustufen können danach fertig gestellt werden.“

Supervisor: „Das klingt doch nach einem möglichen Kompromiss. Glauben Sie, dass Ihr Chef sich auf sowas einlassen würde?“

Entwickler: „Vielleicht, aber wahrscheinlich nicht.“

Supervisor: „Probieren Sie es bitte aus.“

### Verantwortung in Softwareprojekten

Wer hat denn nun die Schuld an der ganzen Misere? Wer kann etwas ändern? Oder sind alle den Umständen ausgeliefert und müssen sich damit abfinden? („Wenn es dafür wenigstens ausreichend Schmerzensgeld gäbe“, sagen alle.) Jeder ist verantwortlich für alles, was er tut (und auch für alles, was er sagt oder nicht sagt).

So sind die IT-Manager verantwortlich dafür, dem Kunden Termine zu versprechen, die nicht gehalten werden können, und die Softwareentwickler sind verantwortlich dafür, diese utopischen Termine (vielleicht murrend) zu akzeptieren. Die Softwareentwickler sind weiterhin dafür verantwortlich, wenn sie sich entscheiden, wegen des Termindrucks die Qualität zu opfern und damit auch für den Zustand des Systems beim Kunden.

Übrigens: Verantwortung ist etwas anderes als Schuld. Mit Schuld assoziiert man etwas Negatives, und die Schuldigen werden bestraft, damit sie nicht nochmal Straftaten begehen. Verantwortung kann man tragen, z.B. wenn man sich für eine

bestimmte Art und Weise *entschieden* hat. Für seine Fehler die Verantwortung zu übernehmen deutet nicht auf Schwäche hin, sondern auf breite Schultern. Fehlentscheidungen, die jemand verantwortet, gehören daher auch nicht bestraft. Stattdessen muss man aus ihnen für die Zukunft lernen.

Beide – IT-Manager und Softwareentwickler – haben die Möglichkeit, aus dem Teufelskreis, den viele euphemistisch „Business“ nennen, auszubrechen. Nein! Sie haben nicht die Möglichkeit, sie haben die *Pflicht*, auszubrechen. Der IT-Manager, der dem Kunden einen Termin verspricht, den seine Entwickler nicht halten können, *lügt* seinen Kunden an. Der Entwickler, der dem IT-Manager nicht laut und deutlich mitteilt, dass der Termin nicht gehalten werden kann, *lügt* seinen Manager an. Macht der Entwickler während des Projektverlaufs den Kunden nicht auf den unhaltbaren Termin aufmerksam, *lügt* er den Kunden an.

### Berufsethos

1. Jeder ist verantwortlich für alles, was er tut.
2. Jeder ist verantwortlich für alles, was er denkt.
3. Jeder ist verantwortlich für alles, was mit ihm geschieht.

Zu 1.: Wenn Entwickler Qualität opfern, um einen Termin zu halten, so ist das ihre Entscheidung, und sie sind dafür verantwortlich. Das gilt auch, wenn Manager es so angeordnet haben. In diesem Fall geben die Entwickler die Verantwortung nämlich nicht vollständig ab, wenn auch der Manager hier mitverantwortlich ist.

Zu 2.: Wenn IT-Manager über ihre Softwareentwickler denken, sie seien faule Tunichtgute, ist das nicht die objektive Wahrheit. Es ist die Entscheidung des IT-Managers, so zu denken und mit dieser Denkweise möglicherweise konstruktive Lösungswege zu versperren.

Zu 3.: Wenn ein Entwickler in ein Projekt mit unhaltbaren Terminen gesteckt wird, ist es seine Entscheidung, das zuzulassen. Es gibt immer eine Alternative – im schlimmsten Fall die Kündigung – aber häufig helfen auch schon deutlich weniger drastische Maßnahmen.

Der dritte Punkt ist in seiner Allgemeinheit – kontrovers und objektiv gesehen – wahrscheinlich sogar falsch. Aber diese Sichtweise ist sehr nützlich. Sie holt uns aus der Opferrolle, in der wir so gerne in Selbstmitleid zerfließen. Wenn jemand heute keinen Arbeitsplatz findet, mag das an den Umständen liegen und nicht an ihm selbst. Mit dieser Sichtweise hat der Arbeitssuchende aber nur eine Möglichkeit: resignieren und auf bessere Umstände hoffen. Der Arbeitssuchende, der sich selbst in der Verantwortung sieht, sucht nach Lösungen. Kann ich meine Bewerbungsstrategie ändern? Kann ich Qualifikationen erwerben, die meine Chancen erhöhen?

### Aufruf zur Revolte?

Müssen Sie jetzt also auf die Barrikaden gehen und als möglicherweise einziger Aufrechter in ihrem Unternehmen die Kündigung riskieren? Das müssen Sie mit sich selber ausmachen. Sie müssen sich *entscheiden*. Sie können sich dafür entscheiden, alles so zu lassen, wie es ist. Es ist Ihre Entscheidung und Ihre Verantwortung. Sie dürfen sich dann aber nicht darüber beschweren, wenn sie unter irrsinnigen Deadlines und anderen Dilbert-artigen Auswüchsen leiden müssen.

Lügen verbietet sich, das können Sie auch dem IEEE Code of Ethics [1] entnehmen, den ihr Chef bestimmt auch unterstützen würde. Was überhaupt zu der Idee führt, man könnte in der Firma, im Team inklusive Chef ja mal vereinbaren, wie man es halten will mit der Zusammenarbeit.



Dipl.-Inform. **Henning Wolf** ist Geschäftsführer von akquinet agile. Er verfügt über mehrjährige Erfahrung aus agilen Softwareprojekten als Projektleiter, Entwickler, Kundenberater und XP-Coach. Darüber hinaus hat er zahlreiche Artikel und Tagungsbeiträge über agile Softwareentwicklung verfasst und ist Buchautor. Er geht seit Jahren zur Supervision, das obige Gespräch wurde aber nicht mit ihm geführt.



Dipl.-Inform. **Stefan Roock** ist Senior-IT-Berater beim selben Unternehmen und verfügt über mehrjährige Erfahrung aus agilen Softwareprojekten als Projektleiter, Entwickler, Kundenberater, Entwicklerberater und XP-Coach. Darüber hinaus hat er zahlreiche Artikel und Tagungsbeiträge über agile Softwareentwicklung verfasst und ist Autor mehrerer Bücher.

### ■ Links & Literatur

- [1] IEEE Code of Ethics:  
[www.ieee.org/about/whatis/code.xml](http://www.ieee.org/about/whatis/code.xml)