



## Erfolgsfaktor (Soft) Skills

# Testen – Teil 1: Beziehungsschwierigkeiten zwischen Entwicklern und Testern

Manuel Arnold

Die Softwarekrise, die Mitte der Sechzigerjahre ausgerufen wurde und bis heute anhält, ist eng mit dem Thema „Soft Skills“ verbunden. Die Informatikbranche hat beim Entwickeln und Testen von Individualsoftware bis heute den „Stein der Weisen“ nicht gefunden. Die wenigsten Projekte scheitern allerdings aufgrund technischer Schwierigkeiten, dazu sind die Softwareentwickler einfach viel zu gute Problemlöser. Die meisten Projekte scheitern in erster Linie aufgrund von Kommunikationsschwierigkeiten zwischen den beteiligten Parteien im Projekt. In diesem ersten Artikel der Reihe über Soft(ware) Skills des Testens soll beschrieben werden, wie man Beziehungsschwierigkeiten zwischen Entwicklern und Testern lösen kann.

## Soziale Kompetenzen

Man unterscheidet allgemein zwischen fachlicher (Hard Skills) und sozialer Kompetenz (Soft Skills) von Mitarbeitern. In dieser kleinen Serie geht es speziell um die für das Testen von Softwaresystemen notwendigen sozialen Kompetenzen. Im ersten Teil wird erklärt, wie sich die Kommunikation zwischen Entwicklern und Testern verbessern lässt. Eine Verbesserung kann dadurch erreicht werden, dass man Wahrnehmungs- und Rückmeldungsprozesse in kleinen Schritten verändert. Ein Tester sollte beispielsweise einem Entwickler Softwarefehler sensibel kommunizieren.

Im zweiten Teil wird es um zentrale Soft Skills für das Testen gehen, also darum, welche sozialen Kompetenzen ein Tester braucht, um seine Tests erfolgreich zum Abschluss zu bringen. Der abschließende dritte Teil wird sich mit der Zusammenstellung von Entwickler- und Testteams beschäftigen.

## Kommunikationsprobleme beim Code-Review

Vorausgestellt sei, dass die angeführten Beispiele rein fiktiv sind und in keiner Weise mit dem Berufsalltag des Autors zu tun haben bzw. hatten. Vielmehr sind sie bewusst überzeichnet, um klarer die wesentlichen Kriterien einer gelungenen Kommunikation herausarbeiten zu können. Die Beispiele sollen dabei helfen, die zwischenmenschlichen Phänomene transparent und anschaulich zu machen.

Jeder weiß, dass sich Menschen oft wegen einfacher Kleinigkeiten in die Haare kriegen. Auf den ersten Blick erscheint dies irrational und unerklärlich. Folgendes Beispiel aus der Berufswelt des Entwicklers und Testers soll zur eigenen Klarheit und daraus resultierend zur Verbesserung der eigenen Kommunikation befähigen. Ein Tester führt mit einem Entwickler ein Code-Review durch. Dabei möchte der Tester dem Entwickler den Sinn des defensiven Programmierens verdeutlichen. Beim offensiven Programmierstil in Listing 1 wird gleich die Variable `isValid` auf `true` gesetzt und dann im Ablauf unter Umständen

```
public boolean validate() {
    boolean isValid = true;
    if(...) {
        isValid = false;
    }
    ...
}
```

Listing 1: Offensives Programmieren

```
public boolean validate() {
    boolean isValid = false;
    if(...) {
        isValid = true;
    }
    ...
}
```

Listing 2: Defensives Programmieren

den auf `false`. Anhand dieses Listings entwickelt sich folgender Dialog.

**Tester:** Generell sollte beim Programmieren nicht die optimistische Annahme getroffen werden, sondern die pessimistische. Das führt zur defensiven Programmierung. Es ist besser, wenn man grundsätzlich von einem Fehlerfall ausgeht und nur, wenn mindestens eine Bedingung erfüllt ist, das Ergebnis auf `true` setzt. Es ist wesentlich sicherer, grundsätzlich von eigenen Fehlern auszugehen und im defensiven Stil zu programmieren.

Obwohl der Tester viele stichhaltige Argumente dafür bringt, dass defensives Programmieren in jedem Fall sicherer ist als offensives, kann er den Entwickler nicht überzeugen. Auch die Begründung, warum in seinem Fall eine `ArrayList` besser sei als eine `HashMap`, kann der Tester nicht mehr plausibel machen. Der Entwickler meint, der Konflikt liege daran, dass er Spring- und Hibernate-Experte sei und der Tester keine Ahnung davon habe. Der Tester kontert: „So wie du programmierst, so habe ich programmiert, als ich 15 Jahre alt war“.

Beide haben nun das Gefühl, beim anderen nicht anzukommen und gegen eine Wand zu reden.

Watzlawicks zweites Kommunikationsaxiom kann eine mögliche Erklärung für dieses jedem bekannte Phänomen geben: „Jede Kommunikation hat einen Inhalts- und Beziehungsaspekt“ (vgl. [Watz69], S. 56), wobei der Beziehungsaspekt stark den Inhaltsaspekt bestimmt. Jede zwischenmenschliche Kommunikation findet auf zwei Ebenen statt: auf einer Verstandesebene (Stichwort: Backend) und auf einer Gefühlsebene (Stichwort: Frontend). Der Inhaltsaspekt einer Nachricht vermittelt die Daten (wie bspw. Daten in einer Datenbank), der Beziehungsaspekt weist an, wie die Daten aufzufassen sind (wie bspw. das Frontend). Deshalb ist die Gefühlsebene die wichtigere Ebene von beiden. Voraussetzung für einen inhaltlichen Verständigungsprozess ist eine störungsfreie Beziehung. Vergleiche hierzu Ruth Cohns Grundsatz [Cohn92] für die Arbeit in Gruppen: Störungen haben Vorrang vor der inhaltlichen Arbeit.

Allgemein gilt, dass Störungen auf der Gefühlsebene für gewöhnlich „nur“ als Störungen auf der Sachebene wahrgenommen werden. Dies bedeutet, dass Störungen auf der Sachebene oft verlagerte Beziehungsstörungen sind.

Das Gespräch zwischen Entwickler und Tester verlief im oben genannten Beispiel nach einem unreflektierten Reiz-Reaktionsschema ab. Die Wahrnehmung löst Gefühle und bestimmte meist automatisierte Reaktionsmuster aus. Abbildung 1 soll verdeutlichen, dass jede Kommunikation eine Inhalts- und eine Beziehungsebene hat. Dies kann zu einem ungewollten, weil destruktiven und emotionalen Reiz-Reaktionsmechanismus führen.

## Kleiner Schritt – große Wirkung: die Macht des Wahrnehmungsprozesses durch überprüfen der eigenen Wahrnehmung

Zwischen Reiz und Reaktion liegt ein kleiner, scheinbar kaum wahrzunehmender Raum, der es ermöglicht, nicht automatisch zu reagieren. Er lässt eine bewusste Entscheidung zu. Wegen der Interferenzen von Inhalts- und Beziehungsebene kann eine Äußerung nie eindeutig interpretiert werden.

Es gibt verschiedene Möglichkeiten, eine Aussage zu überprüfen: Ansprechen der mehrdeutigen Signale, Rückmeldung über die Wirkung der Aussage oder konkrete Frage nach der Bedeutung.

Abbildung 2 soll verdeutlichen, wie ein gelungener Wahrnehmungs- und Rückmeldungsprozess Klarheit bringen und eine Eskalation verhindern kann.

Der Kontakt zwischen Menschen wird stets zuerst auf der Gefühlsebene aufgenommen und nicht auf der inhaltlichen Ebene. Bei Code-Reviews, bei Zielvereinbarungen oder bei anderen Besprechungen ist es äußerst hilfreich, die Kommunikation mit seinen Gefühlen zu beginnen.

So könnte der Anfang für den Code-Review lauten: „Dein Code hat mir sehr gut gefallen, weil er sehr klar, leicht verständlich und leicht wartbar ist. Deinen Code fand ich an vielen Stellen einfach brilliant! Einige Dinge könnte man zur Fehlerreduzierung noch verbessern wie bspw. defensiver zu programmieren!“ So weiß der Entwickler gleich, woran er ist und wie eine Aussage zu verstehen ist.

Zu Beginn einer zwischenmenschlichen Kommunikation fragt sich jeder Gesprächspartner immer zuerst, meist aber unbewusst und unreflektiert, ob das Gegenüber ihn verletzen oder unterstützen wird [Merl06].

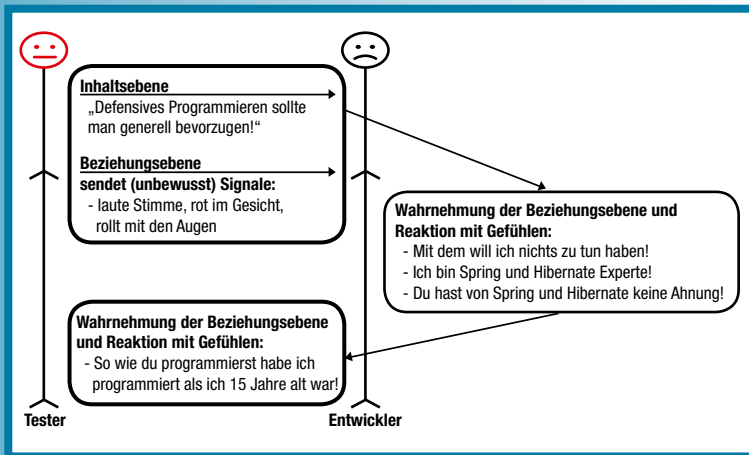


Abb. 1: Verschmelzung von Reiz und Reaktion

## Problemlösung mit und ohne Soft(ware) Skills

### Was würde vermutlich in der Realität passieren?

In einem Briefwechsel stellt Albert Einstein die Frage an Sigmund Freud: Wie ist der Zusammenhang zwischen Gerechtigkeit und Macht? Freud antwortet biologisch, indem er meint, entscheidend sei nicht die Frage nach Gerechtigkeit, letztlich sei es wie im Tierreich, dass sich dort wie auch bei den Menschen immer der Mächtigere durchsetze. Gerechtigkeit spiele bei Entscheidungen keine Rolle [EiFr33].

Genauso könnte es auch in diesem Fall sein. Obwohl bekannt ist, dass der Entwickler seit Jahren gute Arbeit abliefert, kann es sein, dass er laut Projektleiter das Projekt verlassen soll. Der Tester wird vermutlich aufgrund seiner Position der Stärkere sein und, wenn er sich mit dem Projektleiter gut versteht, dann steigen seine Chancen noch einmal, dass er sich gegenüber dem Entwickler durchsetzen kann.

### Wie hätten sich Entwickler und Tester verhalten müssen, um ein konstruktives Gespräch zu ermöglichen?

Damit ein solches Gespräch erfolgreich verlaufen kann, muss es strukturiert werden. Schulz von Thun ist der Auffassung, dass eine bewusste Unterscheidung gemäß dem folgenden Schema eine zentrale Voraussetzung dafür ist, denn nur so bekommt der Empfänger einer Nachricht Klarheit darüber, welche Anteile er selbst an der Kommunikation hat (vgl. [Schu81], S. 72 – 75):

- ▼ **Wahrnehmen**, was der Sender einer Nachricht sagt.
- ▼ **Interpretieren** der Signale und der Aussagen des Senders, das heißt, den Wahrnehmungsinhalten wird eine Bedeutung zugeschrieben, die mit den subjektiven Erfahrungen, Erwartungen und Bedürfnissen verbunden ist.
- ▼ **Reaktion mit seinen eigenen Gefühlen**: Weiß der Empfänger einer Nachricht, was das Gesagte bei ihm auslöst, kann er seine (unbewussten) Signale, welche er in einem Gespräch sendet, bewusster steuern.

Dieses Schema ist eine wichtige Voraussetzung, um angemessen Feedback geben zu können.

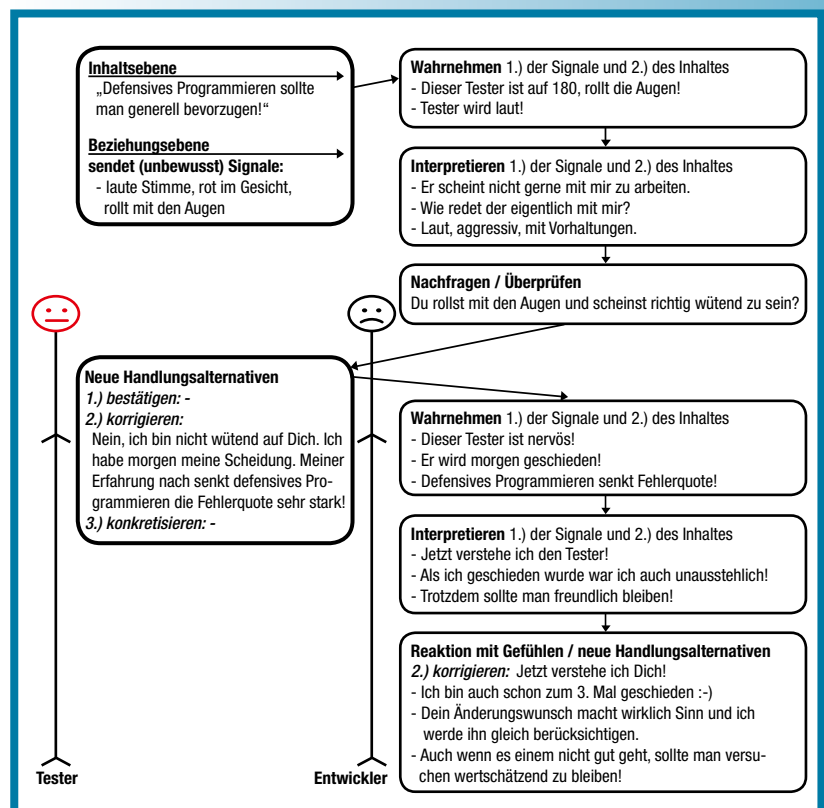


Abb. 2: Prinzip der kleinen Schritte: Sensibilisierung für Wahrnehmungs- und Rückmeldungsprozesse

## Lösung des Konflikts

Ist die Beziehung zwischen Tester und Entwickler gestört, ist ein Code-Review nicht sinnvoll. Es ist wenig hilfreich zu sagen, dass Tester und Entwickler sachlich bleiben sollen. Probleme auf einer Ebene müssen immer auf der gleichen Ebene gelöst werden.

Der erfahrene und sozialkompetente Teamleiter erfährt, dass der Projektleiter einen Softwareentwickler aus dem Projekt nehmen möchte. Daraufhin werden der Entwickler und der Tester zu einem Gespräch mit dem Teamleiter eingeladen. Im Gespräch stellt sich heraus, dass der Tester das Gefühl hat, dass der Entwickler seine Vorschläge nicht ernst nimmt und ihn dies sehr ärgert. Der Entwickler meint, die beleidigende Art, wie der Tester mit jedem im Team umgeht, sei unerträglich. So lasse er mit sich nicht reden. Anschließend gibt es eine Besprechung mit dem gesamten Projektteam, bei der die Entwickler meinen, dass der Tester zwar fachlich wirklich firm sei, aber permanent alle Entwickler im Team beleidige.

Nun ist der Tester geschockt. In den nächsten Tagen merkt man die Bemühungen des Testers, sein Verhalten umzustellen. Durch das Gespräch mit dem Teamleiter wird aus der Gruppe ein Team, in dem die Beziehungen untereinander geklärt sind und es für alle Teammitglieder klare Grenzen gibt. So wird das Klima im Projektteam immer besser.

Es kann der falsche Eindruck entstehen, dass die Kommunikation durch „nettes Getue“ funktioniere. Die Basis für ein Gespräch ist aber die Kongruenz zwischen Denken und Fühlen mit dem Gesagten (vgl. [Schu81], S. 116 f.).

## Ursachen für Kommunikationsstörungen

Ein Grund dafür, dass es bei der Nachrichtenübermittlung immer wieder zu Komplikationen kommt, kann darin bestehen, dass Sender und Empfänger einer Nachricht einen anderen Aspekt einer Nachricht wahrnehmen.

Jede Nachricht hat nicht nur einen Sach- und einen Beziehungsaspekt, sondern zudem einen Appell- sowie Selbstoffenbarungsaspekt:

- ▼ **Sachaspekt** (blau in Abb. 3): Wie ist der Sachverhalt zu verstehen?
- ▼ **Beziehungsaspekt** (gelb): Wie steht er zu mir? Welche Art der Beziehung möchte er aufnehmen?
- ▼ **Selbstoffenbarungsaspekt** (grün): Was ist das für ein Mensch? Wie geht es ihm?
- ▼ **Appellaspekt** (rot): Was soll ich denken, fühlen und tun?

Ein kleines Beispiel erläutert mithilfe des Kommunikationsquadrats, wie eine Kommunikationsstörung entstehen kann. Ein Tester macht eine Fehleroffenbarung:

*Tester* (schüttelt den Kopf! Hat einen roten Kopf und brüllt:)  
Es funktioniert nicht! Bringen Sie das in Ordnung!

*Entwickler* (wahrnehmen und interpretieren):

- ▼ **Sachinformation**: Es gibt einen Fehler.
- ▼ **Beziehungsinformation**: Der Tester denkt wohl, dass ich nur mangelhafte Qualität abliefern. Es ist unglaublich, wie dieser Mensch sich aufführt.
- ▼ **Appellaspekt**: Ich soll den Fehler schnell beheben!
- ▼ **Selbstoffenbarung**: Der Tester ist vermutlich wütend, da in der Produktion ein Fehler aufgetreten ist, der ihm als Tester hätte auffallen müssen. Ich bin mir nicht sicher, ob der Tester über genügend Soft Skills verfügt, um diese Tätigkeit ausüben zu können.

Nachdem der Entwickler seine Wahrnehmung interpretiert hat und gegebenenfalls seine Interpretation durch Nachfragen beim Tester überprüft, sind folgende Reaktionen des *Entwicklers* denkbar:

- ▼ **Sachinformation**: Ich kümmere mich um den Fehler! Sobald ich den Fehler gefunden habe, melde ich mich.
- ▼ **Beziehungsinformation**: Fehler sind ärgerlich für alle Beteiligten. Aber Fehler können niemals 100-prozentig vermieden werden. Die Zusammenarbeit mit dem Tester würde mir leichter fallen, wenn er in stressigen Situationen, nämlich bei der Inbetriebnahme einer neuen Software, etwas gelassener wäre und zumindest nicht schreien würde.
- ▼ **Appellaspekt**: Der Ton macht die Musik – das weiß der Tester auch. Auch wenn ein Fehler in meinem Programmcode sein sollte, ist das keine Art, so rücksichtslos mit Menschen zu reden und einem die Würde zu nehmen.
- ▼ **Selbstoffenbarung**: Auf Dauer kann ich mit ihm nicht zusammenarbeiten, wenn er über so wenig Einfühlungsvermögen verfügt. Er sollte m. E. in puncto Soft Skills noch sehr viel lernen.

Wie diese überspitzt dargestellte Situation zeigt, können die Reaktionen sehr unterschiedlich ausfallen, je nachdem mit welchem „Ohr“ die Fehleroffenbarung wahrgenommen wird. Das Modell des Kommunikationsquadrats erklärt auch, warum Menschen etwas ganz anderes verstehen als das, was eigentlich gemeint war. Es liegt an der freien Wahl des Empfängers, auf einen Aspekt einer Nachricht zu reagieren. Dies macht die Kommunikation so störungsanfällig.

Weitere Störquellen sind oft einseitige Hörgewohnheiten auf einem der vier „Ohren“. Problematisch ist auch, wenn das Gehörte ständig vor dem Hintergrund eines (verzerrten) Selbstbildes wahrgenommen wird, wie z. B. „ich bin OK und du bist nicht OK“ [PoSch05].

Ein weiterer Grund für Kommunikationsstörungen kann darin bestehen, dass sich der Tester entwertend und zynisch verhält, indem er fragt: „Passiert Ihnen dieser Fehler ständig?“ Mit Carl Rogers Kommunikationsansatz des aktiven Zuhörens [Rog83] kann man subtil entwertende Formulierungen, die höflich und freundlich vorgebracht werden, ein Ausfragen sowie ein Bloßstellen, d. h. einem Menschen die Würde nehmen, als falsche Kommunikation identifizieren.

## Mitteilung von Fehlern

Um als Zuhörer besser zu verstehen, welchen Aspekt einer Nachricht ein Sender meinen könnte, bzw. um als Sender einen Aspekt einer Nachricht bewusst hervorzuheben, kann die Transaktionsanalyse viel zur eigenen Klarheit beitragen. Das Modell der Transaktionsanalyse beruht auf drei verschiedenen, sogenannten „Ich-Zuständen“:

- ▼ **Das Eltern-Ich**: Jeder Mensch hat seine Eltern verinnerlicht. In der Kommunikation äußert sich das in der sorgenden Haltung oder in der Kritik, welche einem Gesprächspartner entgegengebracht wird.



Abb. 3: Das Kommunikationsquadrat, aus [Schu]



- ▼ Das *Erwachsenen-Ich*: In diesem Zustand kann ein Gesprächspartner weitestgehend sachlich und objektiv kommunizieren. Das Gegenüber wird gleichwertig, respektvoll und sachlich-konstruktiv behandelt.
- ▼ Das *Kind-Ich*: So wie die Eltern verinnerlicht wurden, so lebt auch das Kind im Menschen weiter. Manchmal antwortet das Kind so frei, wie sich der Sender einer Nachricht gerade fühlt, manchmal rebellisch oder ein anderes Mal angepasst.

## Zusammenhang der Aspekte einer Nachricht mit der Transaktionsanalyse

Wie man in Abb. 4 sieht, besteht ein direkter Zusammenhang zwischen den Aspekten einer Nachricht und der Transaktionsanalyse. Der Beziehungsaspekt einer Nachricht steht in



Abb. 4: Zusammenhang der Aspekte einer Nachricht mit der Transaktionsanalyse

engem Zusammenhang mit der Selbstoffenbarung. Die Selbstoffenbarung meint in der Transaktionsanalyse, dass Inhalt (Erwachsenen-Ich) und Gefühl (Kind-Ich) beim Gesagten echt bzw. kongruent sein müssen, sonst wird dies beim Hörer als seltsam und unangenehm wahrgenommen.

## Besonderheit der Kommunikation zwischen Entwickler und Tester

Watzlawicks fünftes Axiom hilft, die Besonderheit der Kommunikation zwischen Entwicklern und Testern zu beschreiben: „Zwischenmenschliche Kommunikationsabläufe sind entweder symmetrisch [M.A. gleichwertig] oder komplementär [M.A. über- oder untergeordnet], je nachdem, ob die Beziehung zwischen den Partnern auf Gleichheit oder Unterschiedlichkeit beruht“ (vgl. [Watz69], S. 70).

Ist die Position zweier Mitarbeiter gleich hoch, dann nennt man dies „spiegelbildliche oder symmetrische“ Kommunikation. Die Partner sind sowohl in Stärke und Schwäche und jeder anderen Hinsicht ebenbürtig. Ist die Position zweier Mitarbeiter dagegen über- oder untergeordnet, dann nennt man dies eine „ergänzende oder komplementäre“ Kommunikation. Ein Kommunikationspartner spielt die Rolle des Überlegenen der andere des Unterlegenen.

Beim Testen besteht per se eine komplementäre Beziehung, da der Tester die Arbeit des Entwicklers kontrolliert. Der Tester wird durch seine Tests mit großer Wahrscheinlichkeit Fehler finden und dies dem Entwickler mitteilen.

Kommunikationspsychologisch ist es dagegen sehr sinnvoll, wenn auf der Beziehungsebene ein gleichwertiges Verhältnis besteht. Die Gründe dafür sind: Erstens haben beide Kollegen eine gleichwertige berufliche Qualifikation und beide verfolgen das gleiche Ziel, nämlich ein Produkt abzuliefern, das so gut wie möglich werden soll. Nur wenn Entwickler und Tester menschlich auf einer Ebene stehen, entsteht ein gutes Arbeitsklima.

Zweitens ist der Tester immer wieder von Informationen des Entwicklers abhängig, wie bspw. der Stand in puncto Implementierung eines Anwendungsfalles ist oder wie er ein Requirement konkret realisiert hat. Beim Testen arbeitet der Tester an vielen Stellen Hand in Hand mit dem Entwicklerteam

zusammen. Ein letzter wichtiger Grund ist, dass die meisten Entwickler ihr Wissen und ihre Motivation je nach Betriebsklima der Firma in unterschiedlichem Maße zu Verfügung stellen.

## Verschiedene Ansätze der Übermittlung von Fehlern

Beim Vermitteln von Testergebnissen können unterschiedliche Ansätze verwendet werden [PeRi02]:

- ▼ *Formloser Ansatz*: Hier meldet der Tester seine Testergebnisse mit formlosen Mitteln wie mit der mündlichen Aussage: Klappt nicht! Was fehlt ist die Dokumentation, welche die Details des Zustandes der Software offen legt. So ist dieser Ansatz für den Tester und das Management wenig hilfreich.
- ▼ *Negativer Ansatz*: Der Tester meldet nur schlechte Nachrichten. Dies ist dann sinnvoll, wenn der Tester nicht möchte, dass eine Software in Produktion gehen soll. Der Nachteil dieses Ansatzes ist, dass er nicht objektiv ist und die Entwickler nicht motiviert.
- ▼ *Zuckerbrot (Täuschungs-)Ansatz*: Der Tester berichtet nur, was die Software bereits kann, und spielt die negativen Ergebnisse seiner Tests herunter. Der Grund kann Angst vor dem Management sein und dass er seine Freunde im Projektteam nicht mit der Realität konfrontieren möchte. Dieser Ansatz täuscht Menschen über den wirklichen Zustand eines Programms hinweg. Beim Testen geht es immer darum, eine Realitätsannäherung zu erzielen und seinen Kollegen eine mögliche Enttäuschung zuzumuten, an der sie aber letztlich wachsen können.
- ▼ *Objektiver Ansatz*: Dieser Ansatz vermittelt alle Fakten, die laut IEEE 829 [STD] in einem Testprotokoll vorhanden sein müssen oder vorhanden sein können. Die Informationen können auch in unregelmäßigen Abständen mitgeteilt werden.
- ▼ *Prozessorientierter Ansatz*: Dieser Ansatz beinhaltet den objektiven Ansatz und erweitert diesen mit Prozessen, wann der Bericht erstellt und kommuniziert wird. Der prozessorientierte Ansatz kann bspw. wöchentlich zeigen, wie hoch die Testabdeckung mit Unit-Tests ist oder wie viele schwere Fehler noch nicht gefixt sind.

Beim Projektabschluss kann ein Meta-Gespräch [ViSch07] mit allen Projektmitarbeitern sehr nützlich für die weitere Zusammenarbeit sein. Folgende Fragen sind diesbezüglich denkbar: Was hat mir in puncto Testen gefallen? Was hat mir in puncto Testen nicht gefallen? Diese Rückmeldungen sollte der Tester sehr ernst nehmen.

Der prozessorientierte Ansatz ist am besten geeignet, regelmäßig und verlässlich Testergebnisse zu kommunizieren, damit die notwendigen Schlussfolgerungen gezogen werden können. Beispielsweise kann diskutiert werden, welche wichtigen Bugs schon gefixt sind und welche Bugfixes als nächstes anstehen. Für den Tester bedeutet dieser Ansatz, dass er am effizientesten ist, wenn er einerseits bei den Fakten bleibt und seine Befunde klar kommuniziert, gleichzeitig Interesse an der Person des Entwicklers, an seinen Fehlerarten und an seiner fachlichen Weiterentwicklung ausdrückt. Eine kollegial-wertschätzende Grundeinstellung und Authentizität erhöhen die Effizienz.

Für die Zusammenarbeit zwischen Entwickler und Tester ist es hilfreich, wenn der Entwickler auch Verständnis für die Aufgabe des Testers entgegenbringt – nämlich gemeinsam an der Verbesserung eines Produktes zu arbeiten und die Arbeit des Testers als Entwicklungschance für sich selbst erlebt. Das Verständnis des Entwicklers kann durch das Kennenlernen der Tätigkeit des Testers gesteigert werden. Verschiedene Firmen



verlangen, dass die Softwareentwickler auch die Zertifizierung für den Certified Tester, Foundation Level [ISTQBCT] ablegen.

## Ausblick

Im nächsten Teil der Serie soll beschrieben werden, welche sozialen und ethischen Kompetenzen die Voraussetzung dafür sind, damit ein Testspezialist seine Arbeit erfolgreich ausführen kann. Dieser Artikel wird den notwendigen Brückenschlag von der Kommunikationspsychologie hin zu den neuesten Untersuchungsergebnissen der Hirnforschung darstellen.

## Literatur und Links

**[Cohn92]** R. C. Cohn, Von der Psychoanalyse zur themenzentrierten Interaktion, 11. Auflage, Klett-Cotta, 1992

**[EiFr33]** A. Einstein, S. Freud, Warum Krieg?, Wien, 1953, S. 19 f. oder aus <http://www.netzwerk-regenbogen.de/einstfre021224.html>, Zugriff am 8.6.2011

**[ISTQBCT]** International Software Testing Qualifications Board, Certified Tester, [http://de.wikipedia.org/wiki/ISTQB\\_Certified\\_Tester](http://de.wikipedia.org/wiki/ISTQB_Certified_Tester)

**[Merl06]** H. Merl, Über das Offensichtliche oder:

Den Wald vor lauter Bäumen sehen, Krammer-Verlag, 2006

**[PeRi02]** W. E. Perry, R. W. Rice, Die 10 goldenen Regeln des Softwaretestens, mitp-Verlag, 2002, S. 193 - 195

**[PoSch05]** M. Poser, K. Schneider, Leiten, Lehren und Beraten: Fallorientiertes Lehr- und Arbeitsbuch für Pflegemanager und Pflegepädagogen, Verlag Huber, Bern, 2005, S. 213

**[Rog83]** C. Rogers, Die klientenzentrierte Gesprächspsychotherapie, Fischer-Verlag, 1983

**[Schu]** Kommunikationsquadrat, aus

[http://www.schulz-von-thun.de/index.php?rex\\_img\\_type=bild\\_gross\\_textbereich&rex\\_img\\_file=bild3.jpg](http://www.schulz-von-thun.de/index.php?rex_img_type=bild_gross_textbereich&rex_img_file=bild3.jpg), Zugriff am 8.6. 2011

**[Schu81]** F. Schulz von Thun, Miteinander reden 1. Störungen und Klärungen, Rowohlt, Reinbek, 1981

**[STD]** IEEE 829 Standard for Software Test Documentation, [http://de.wikipedia.org/wiki/IEEE\\_829](http://de.wikipedia.org/wiki/IEEE_829)

**[ViSch07]** U. Vigerschow, B. Schneider, Soft Skills für Softwareentwickler, Dpunkt-Verlag, 2007, S. 71 f.

**[Watz69]** P. Watzlawick u. a., Menschliche Kommunikation. Formen, Störungen, Paradoxien, 4. unveränderte Aufl., Verlag Huber, Bern, 1969



**Manuel Arnold** hat im Erstberuf Lehramt studiert mit den Fächern Pädagogik, DV/Informatik und Psychologie (Beratung). Interesse an Soft Skills ergibt sich aus dem beruflichen Werdegang.  
E-Mail: [arnold4711@gmail.com](mailto:arnold4711@gmail.com)