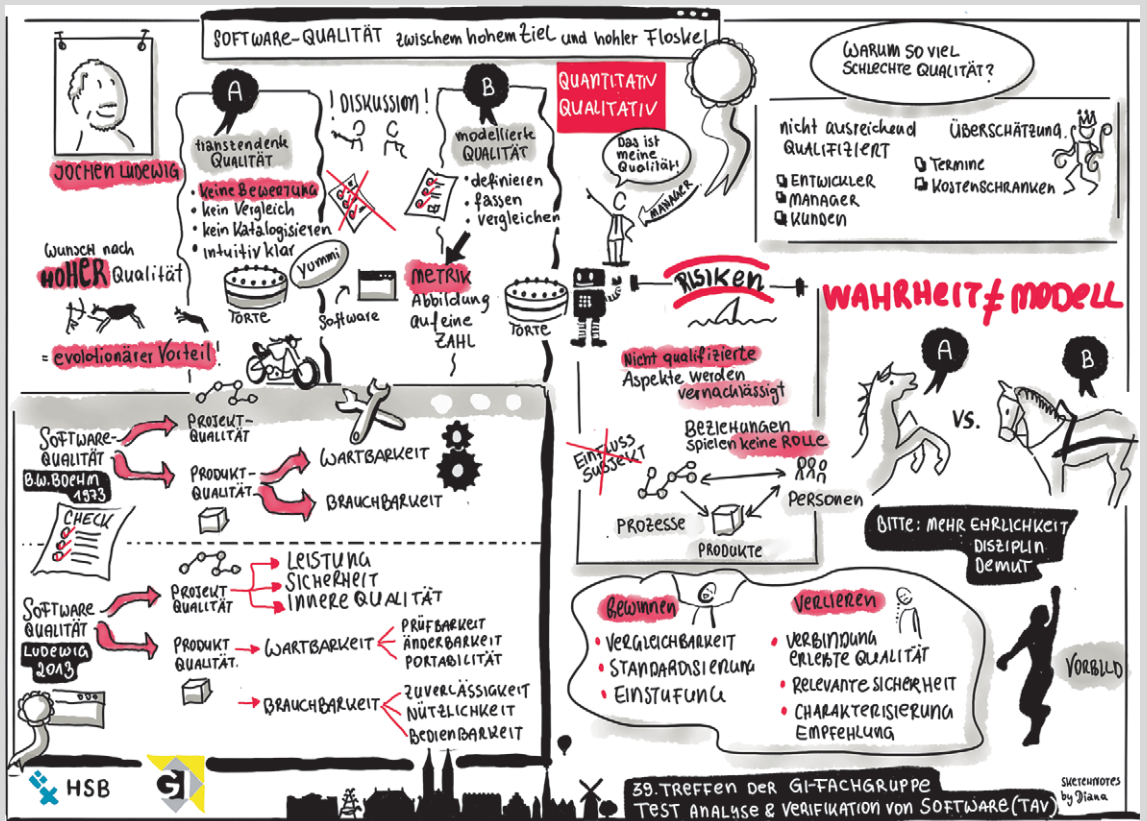


Softwarequalität zwischen hohem Ziel und hohler Floskel

Jochen Ludewig



*Qualität ist ein schwieriger, mehrdeutiger Begriff.
Dieser Beitrag beleuchtet ihn von verschiedenen Seiten.
Schließlich zeigt er, dass es sich lohnt, gute Qualität zu liefern.*

1 Was soll dieser Beitrag vermitteln?

Dieser Beitrag soll zeigen,

- dass wir es bei der Qualität mit zwei sehr verschiedenen Begriffen zu tun haben, der transzendenten und der modellierten Qualität;
- dass wir alle uns eigentlich für die transzendente Qualität interessieren; wir verwenden aber oft die modellierte, weil nur diese zu definieren, zu fassen und zu vergleichen ist;
- dass höhere Qualität sowohl billiger als auch angenehmer ist als schlechte Qualität;
- dass gute Qualität trotzdem kein Selbstläufer ist.

2 »Qualität« und andere Wörter

Ich befasse mich:

- mit Qualität im Sinne der DIN EN ISO 9000:2015-11, der gültigen Norm zum Qualitätsmanagement, als *»Grad, in dem ein Satz inhärenter Merkmale eines Objekts Anforderungen erfüllt«* (und habe leider keinen Raum, um die *inhärenten Merkmale* und die *Anforderungen* zu diskutieren);
- mit der Qualität der **Software** und die der **Softwareprozesse** (oder auch der **Softwareprojekte**), wie sie in IEEE Std. 610.12-1990 definiert sind: (1) *The degree to which a system, component, or process meets specified requirements.* (2) *The degree to which a system, component, or process meets customer or user needs or expectations;*
- **nicht** mit Männern und Frauen, sondern mit **Menschen**, wahlweise auch mit **Personen**;
- im selben ungeschlechtlichen Sinne auch mit **Entwicklern**, **Managern**, **Klienten** und **Kunden**.

3 Urbedürfnis Qualität

Unsere Vorfahren, die in der kalten Altsteinzeit um ihr Überleben kämpfen mussten, hatten wahrlich Probleme genug. Darum überrascht es, dass es unter ihnen Menschen gab, die großartige Kunstwerke geschaffen haben. Davon zeugen Höhlen wie die von *Altamira* und *Lascaux*.

Vermutlich hatten die Bilder damals einen kultischen Zweck. Aber dafür hätten einfachere, schematische Abbildungen genügt. Offenkundig wollten die Künstler **etwas Großes schaffen**. Sie haben sich erfolgreich bemüht, höchste Qualität zu liefern.

Wir sehen: Selbst unter sehr schwierigen Bedingungen kann der Wunsch nach Qualität höhere Priorität haben als z.B. der Wunsch nach Ruhe. Anscheinend ist uns ein **Bedürfnis nach hoher Qualität angeboren**. Warum?



Abb. 1 Höhlenmalerei in Lascaux, Frankreich, ca. 16.000 Jahre alt

Weil hohe Qualität bedeutet:

- Der Umhang ist dicht und schützt den Träger vor Kälte.
- Der Speer fliegt ohne Flattern und dringt tief ins Fleisch der Beute.
- Das Messer, mit dem die Beute bearbeitet wird, ist scharf und haltbar.

Also: Hohe Qualität **verbessert die Chancen, zu überleben** und Nachkommen zu haben. Hohe Qualität ist ein **evolutionärer Vorteil**.

Und da uns die Evolution so programmiert hat, dass wir angenehm, schön und gut finden, was das Überleben unserer Art sichert, haben wir **von Natur aus** nicht nur Bedürfnisse nach Wärme, Nahrung und Sexualität, sondern auch **nach hoher Qualität** (und ebenso nach **Schönheit**). Das ist kein Luxus, sondern ein Beitrag zu unserem Überleben. Dass hohe Qualität **außerdem praktische Vorteile** hat, liegt in der Natur der Sache und ist angenehm.

Bitte beachten Sie: Die bislang betrachtete Qualität, die man nach [Garvin 1984] und [Kitchenham & Pfleeger 1996] als **transzendente Qualität** bezeichnet, kennt keine Bewertung der Qualität mit Punkten oder Schulnoten.

Beispiel:

Eine Person, die ein Stück *Schwarzwälder Kirschtorte* verzehrt hat, wird anschließend vielleicht sagen, es sei *sagenhaft* gewesen (oder *ganz furchtbar*). Das war ihr unmittelbarer Eindruck. Sie zieht dazu keine Kenngrößen heran, die den pH-Wert, die Farbe, die chemische Zusammensetzung oder andere Aspekte beschreiben. Natürlich haben diese Attribute aber zum Eindruck beigetragen, meist ohne dass es der Person bewusst war.

Darum taugt die transzendente Qualität nicht zum Vergleich oder zur Katalogisierung – aber sehr gut zur subjektiven Empfehlung. Ich bezeichne die transzendente Qualität als Qualität A.

Robert M. Pirsig hat sich intensiv mit dieser Qualität befasst. Sein Buch *Zen und die Kunst, ein Motorrad zu warten* [Pirsig 1974] war in den späten Siebzigerjahren Kult. Was war seine Botschaft? Können wir etwas damit anfangen?

Das Buch hat die Form eines (ungewöhnlichen) Romans. Darin geht es um Qualität. Den Hintergrund bildet eine Reise auf dem Motorrad, die der Autor mit seinem Sohn 1968 unternommen hat. Was er im Buch mitteilt (*mir* mitteilt), ist die Bedeutung einer ganzheitlichen Sichtweise (Zen) im ganz normalen Leben, beispielsweise bei der Wartung eines Motorrads. Wer glaubt, dass Zen nur Tempel, Räucherstäbchen und langes Stillsitzen bedeutet, ist überrascht. Für Pirsig ist die Qualität eine fundamentale, intuitiv begründete Kategorie, die über anderen Kategorien (wie Umfang, Gewicht, Preis usw.) steht. Qualität, Wahrheit und Schönheit sind im Kern gleich.

Wenn wir versuchen, die Qualität aufzudröseln, in viele kleine Aspekte zu zerlegen, geht sie uns verloren. Denn die Qualität ist ein Merkmal des Ganzen.

Obviously some things are better than others ... but what's the »betterness«? ... So round and round you go, spinning mental wheels and nowhere finding anyplace to get traction. What the hell is Quality? What is it? [Pirsig 1974, S. 184]

Pirsigs Schriften sind bis heute Gegenstand der Diskussion und Thema weiterer Publikationen (vgl. Webseiten zum Stichwort MOQ = Metaphysics of Quality).

Wir halten fest: Qualität im bisher beschriebenen Sinn (**Qualität A**) ist jedem von uns intuitiv klar, widersetzt sich aber der analytischen Betrachtung. Es ist unmöglich, daran Messungen und rationale Vergleiche festzumachen.



Abb. 2 Robert M. Pirsig auf großer Fahrt mit seinem Sohn Chris, Juli 1968

4 Modellierte Qualität

Wenn wir anfangen, die Qualität zu definieren, zu standardisieren und zu vergleichen, wechseln wir zu einem **konstruierten, modellierten** Qualitätsbegriff (**Qualität B**). Kurz: An die Stelle der transzendenten tritt die **modellierte Qualität**. Tatsächlich besteht dieser Übergang aus zwei Schritten:

Zunächst stellen wir fest, dass die Softwarequalität viele Facetten hat, die wir unterscheiden und definieren müssen, um die Qualität insgesamt in den Griff zu bekommen. Wir entwickeln also eine Taxonomie. Die Softwarequalität gliedern wir in Projekt- und Produktqualität, und diese zerlegen wir weiter, bis uns die Teile einfach genug erscheinen; das sind die **atomaren Qualitäten**.

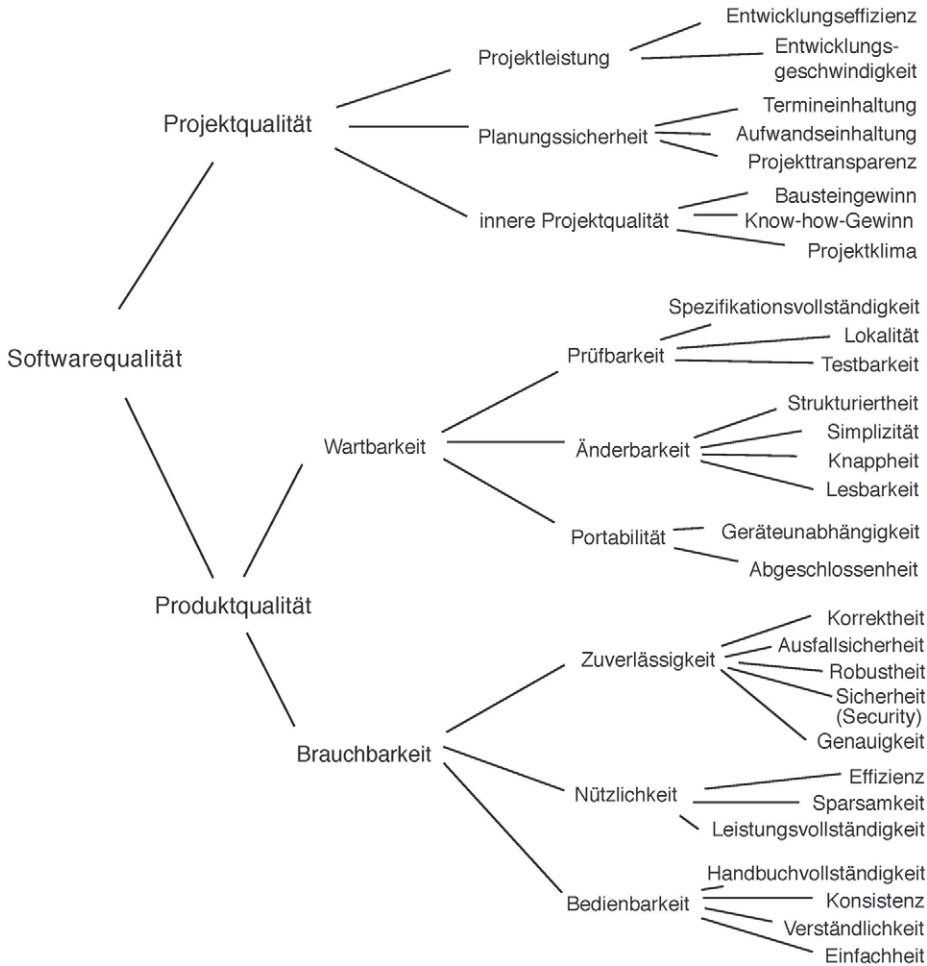


Abb. 3 Eine Taxonomie der Softwarequalität

Die Taxonomie in Abbildung 3 (ähnlich der in [Ludewig & Lichter 2013]) basiert ursprünglich auf Arbeiten von [Boehm et al. 1973]. [ISO/IEC25000] bietet ein ähnliches **Qualitätsmodell** an.

Man darf hier nicht dem Irrtum erliegen, dass diese (oder eine andere) Taxonomie *wahr* sei: Das ist nur ein Modell und wie alle Modelle mehr oder minder nützlich. Darum legen wir die Taxonomie auch so an, wie es für uns praktisch ist, also als Baumstruktur. Natürlich kann man die Begriffe auch anders wählen und anordnen. Beispielsweise kann man diskutieren, wie die Prozessqualität einzuordnen ist, die in diesem Modell nur implizit in der Projektqualität steckt.

Die Zerlegung der Softwarequalität in (letztlich) atomare Qualitäten hat den Vorteil, dass wir über die relative Bedeutung der atomaren Qualitäten nachdenken und – vor allem in der Spezifikation – Aussagen darüber machen, welches

Gewicht, welchen Rang jede einzelne Qualität haben soll. Und wir können dazu Anforderungen formulieren. Auch wenn diese meist nicht formal oder quantifiziert sind, ist der Fortschritt gegenüber den üblichen vagen (oder ganz fehlenden) Aussagen beträchtlich.

Wenn wir ein »hartes«, quantifiziertes Urteil anstreben, müssen wir festlegen, wie den atomaren Qualitäten Zahlenwerte zugeordnet werden können und wie daraus ein Wert für die höheren Knoten des Qualitätsbaums abgeleitet werden kann; als einfachste Lösung können wir die Werte der untergeordneten Knoten addieren. Das liefert aber oft keine sinnvollen Ergebnisse, weil verschiedene Qualitäten ganz unterschiedliches Gewicht haben; insbesondere gibt es K.-O.-Kriterien, die erfüllt sein müssen, um einer Software überhaupt einen Nutzen zuzumessen zu können. Wir kennen das schon aus der Bewertung von Schulaufsätzen: Wenn das Thema verfehlt ist, kann kein Vorzug des Texts die Note 5 verhindern.

Wenden wir diese Überlegungen wieder auf die bereits oben missbrauchte Schwarzwälder Kirschtorte an: Wir wollen nun einen Roboter bauen, der Torten beurteilen kann. Dieser muss mangels menschlicher Urteilskraft viele einzelne Messwerte erheben und daraus ein Urteil synthetisieren. Ob das Urteil plausibel ist, prüfen wir, indem wir das Urteil des Roboters (Qualität B) mit unserem eigenen (Qualität A) vergleichen. Kaum jemand wird, wenn sich die Urteile widersprechen, dem Roboter Recht geben. Anders verhält es sich, wenn der Roboter einen schädlichen Inhaltsstoff erkennt, den wir mit unseren Sinnesorganen nicht wahrnehmen können. Der Vergleich ist also nur sinnvoll, wenn wir und der Roboter dieselben Facetten der Qualität prüfen können und auch prüfen.

Alle Qualitätsmodelle folgen diesem Prinzip. Sie können wie in Abbildung 3 **qualitativ** oder (wie beim Geschmacksroboter) **quantitativ** sein.

Allerdings sind mit Qualität B **Risiken** verbunden:

- Qualitätsaspekte, die sich kaum definieren oder noch weniger quantifizieren lassen, bekommen weniger Aufmerksamkeit und werden folglich vernachlässigt.
- Insbesondere die Beziehungen zwischen Personen, Prozessen und Produkten spielen keine Rolle. Tatsächlich haben sie aber große Bedeutung.
- Der Einfluss des Subjekts (des Betrachters) wird kaum oder gar nicht sichtbar, obwohl die Qualität oft im Auge des Betrachters liegt. Das steckt schon im plattdeutschen Sprichwort:

» *Wat den Eenen sin Uhl is den Annern sin Nachtigall.* «

In allen Fällen ist die Modellierung vom Versprechen begleitet, dass die mithilfe des Modells ermittelte Qualität B stark korreliert ist mit der erlebten, empfundenen Qualität A.

5 Was gewinnen, was verlieren wir mit der modellierten Qualität?

Wir gewinnen

- die Vergleichbarkeit,
- die Möglichkeit zur Standardisierung,
- eine objektive, absolute Einstufung.

Wir verlieren

- die unmittelbare Verbindung zur erlebten Qualität, damit auch den Blick auf denjenigen, der die Qualität erlebt (das Objektive verdrängt das – wichtigere – Subjektive),
- die Sicherheit, dass die festgestellte Qualität relevant ist,
- die Einfachheit einer griffigen, klaren Charakterisierung oder Empfehlung.

Beispiel 1: Kaufempfehlung versus Warentests

Die Stiftung Warentest prüft regelmäßig Produkte gegen ein (den jeweiligen Waren angepasstes) Qualitätsmodell.

Das funktioniert ganz gut, wenn die Waren einander ähnlich sind. Außenseiterprodukte sind so aber nicht fair zu beurteilen.

Beispiel 2: Bewertung von Abschlussarbeiten

Wir haben über Jahre versucht, Diplomarbeiten nach einem Punkteschema zu bewerten. Ein Ergebnis war, dass die Bewertung manchmal brauchbar war, oft aber weit neben dem Eindruck lag, den die Arbeit auf uns, die Leser, machte. Das heißt: Die Punktbewertung erfasst nicht immer, was wir intuitiv wichtig und richtig finden.

6 Technical Debt

Wer ein Haus kaufen oder bauen will, aber das dafür notwendige Geld nicht hat, muss eine Hypothek aufnehmen. Solange die Hypothek nicht getilgt ist, werden Zinsen fällig. Oft stellen Tilgung und Zinszahlung den Schuldner vor große Probleme.

Ward Cunningham hat – in einer Nebenbemerkung – das Konzept der Hypothek als Metapher für die Verpflichtungen verwendet, die die Hersteller einer Software bewusst oder unbewusst eingehen, wenn sie nicht perfekte Software ausliefern [Cunningham 1992]. Die Zinsen sind die Kosten, die die Software verursacht, weil sie nicht den Anforderungen entspricht, also weniger brauchbar oder weniger wartbar ist, als sie sein sollte. Der Tilgung entspricht der Aufwand, den man treiben muss, um die Mängel zu beseitigen.

»Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite ... The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt. Entire engineering organizations can be brought to a stand-still under the debt load of an unconsolidated implementation, object-oriented or otherwise.« [Cunningham 1992]

Wer beim Autofahren eine Geschwindigkeitsbeschränkung missachtet, aber dabei nicht erwischt wird, hat keine Nachteile zu befürchten. Die Chance, folgenlos Fehler zu machen, gibt es aber in der Softwareentwicklung nicht. Und alle Lasten müssen abgetragen werden, bevor die Organisation wie ein überschuldeter Staat unter der Schuldenlast zusammenbricht, weil alles verfügbare Geld durch Zinsen und Tilgung gebunden ist.

Inzwischen versucht man, die Schuld zu quantifizieren und damit den Entscheidungen der Entwickler eine rationale Grundlage zu geben. Das funktioniert für messbare Codemängel (*Code Debt*, z.B. mit SonarQube¹), aber nicht für die Mängel anderer Dokumente, beispielsweise die Architektur (*Design Debt*). Damit besteht die Gefahr, dass der Code weiterhin überbewertet wird und sehr wichtige andere Dokumente wie die Spezifikation weniger Aufmerksamkeit bekommen.

7 Qualität für Bürokraten

Manager und andere Menschen, die nicht direkt mit der Wertschöpfung befasst sind, laufen Gefahr, die ursprüngliche Qualität A aus den Augen zu verlieren. Für sie *ist* Qualität B *die* Qualität.

Dann wird das *Modell* des Softwareprozesses oder der Software zum *Selbstzweck*. Den Softwareentwicklern werden Qualitätskriterien vorgegeben, die nur der Qualität B nützen. An die Stelle des **guten Produkts** tritt das **gut bewertete Produkt**, statt hoher Qualität werden Kennwerte befohlen und durchgesetzt.

Beispiel:

Softwareprozesse werden auf CMMI oder auf ISO 9000 getrimmt. Sinn und Nutzen sind perdu, aber das Kriterium ist erfüllt.

Achtung, das drückt keine Geringschätzung für CMMI aus. Oft ist nur die Reihenfolge verkehrt: Wenn nicht die Verbesserung, sondern das Etikett im Vordergrund steht, geht der Sinn verloren. Diesen Effekt gibt es nicht nur in der Softwaretechnik, sondern in allen Bereichen, wo Inkompetenz regiert, also besonders oft in der Politik: Gebote und Verbote, Steuerlasten und Steuervergünstigungen, die ursprünglich einem sinnvollen Ziel dienen sollten, werden vom Ziel abgekoppelt und damit zur Spielwiese für Lobbyisten und Wahlkämpfer.

1. <http://www.sonarqube.org>

Und die Mitspieler passen sich an: Der Schadstoffbetrug bei VW und anderen Firmen bediente die Metrik-Gläubigen so, wie sie es verdient haben. Dem Umweltschutz haben die Prüfvorschriften nicht genützt, sondern geschadet. Wir können daraus folgern: Jedes Qualitätsmodell bedarf der Diskussion, ob, für wen und wie weit die dargestellten Aspekte und ihre Bewertung tatsächlich zur Qualität A beitragen.

8 Das Streben nach Qualität als Grundhaltung

Der Kunde einer Schreinerwerkstatt muss warten. So beobachtet er, wie der Schreiner die Rückseite einer Schublade sorgfältig bearbeitet.

»Warum machen Sie sich damit so viel Mühe?«, fragt er ungeduldig den Schreiner. »Das sieht doch keiner.«

»Doch«, antwortet der Schreiner, »ich.«

Gute Qualität zu produzieren, ist außerordentlich befriedigend, wie gutes Essen oder guter Wein. Wer sich gute Qualität leisten kann, sei es als Produzent oder als Konsument, ist zu beneiden.

Und: Letztlich ist die gute Qualität billiger. Die sauber gearbeitete Schublade wird später weder klemmen noch auseinanderfallen. Aber es sieht so aus, als ob diese Schreinermeister aussterben.

Ich sehe (am Rechner und an der Software, aber auch an meinem Auto und an vielen anderen Gebrauchsgegenständen), dass uns viele, viele neue Funktionen (»Features«) angeboten werden, während die Qualität sinkt. Und wir verstehen die Produkte nicht mehr.

Beispiel 1:

Vor vierzig Jahren garantierte SIEMENS der Deutschen Bundespost für die Telefonanlagen Ausfallraten in der Größenordnung von 10 s in 10 Jahren. Heute ist der Ausfall des Kommunikationsnetzes kaum noch ein Thema.

Beispiel 2:

Viele Jahre habe ich als Mailprogramm Eudora verwendet, ein einfaches, klares Programm, bis es unter einem neuen Betriebssystem nicht mehr lief. Seitdem verwende ich Apple-Mail. Das bietet jede Menge *Bells & Whistles*, aber keine wohldefinierte Funktionalität. Ich weiß bis heute nicht, welches Informationsmodell dem Programm zugrunde liegt, und habe laufend Probleme.

Kurz: Gute Qualität konkurriert mit dem Bemühen, *schnell vielen vieles* zu bieten.

9 Warum sehen wir viel schlechte Qualität?

Wenn Qualität ein Bedürfnis, ein Lebensmittel jedes Menschen ist, stellt sich die Frage, warum wir so viel schlechte Qualität sehen. Ich vermute, dass die schlechte Qualität vor allem zwei Gründe hat:

■ Erstens sind viele Menschen für ihre Aufgaben nicht ausreichend qualifiziert. Das kann bedeuten:

- Sie machen ihre Arbeit nicht so gut, wie sie sollten.
- Sie hindern andere daran, gut zu arbeiten.
- Sie üben Druck aus, schlecht zu arbeiten.

Diese Beschreibungen lassen sich in unserem Metier (sehr grob!) den Rollen Entwickler, Manager und Kunde zuordnen.

- Viele **Entwickler** haben ihren Beruf nicht gelernt und/oder sind unter dem Druck ihrer Arbeitsbedingungen gegenüber Qualitätsfragen indifferent geworden. Sie machen ihren Job. Wenn sie die Situation durchschauen, igeln sie sich mit Zynismus ein. Das erspart ihnen die Selbstkritik.
 - **Manager** verstehen oft wenig von Software, treffen aber trotzdem Entscheidungen darüber, wie sie entwickelt und bearbeitet werden soll. Dabei stehen kurzfristige Vorteile (vermeintliches Sparen an der Qualität zulasten der Technical Debt) im Vordergrund.
 - **Kunden** verstehen ihre eigenen Interessen nur unzureichend. Sie nutzen ihre Macht, um niedrige Preise, späte Änderungen der Anforderungen und kurze Lieferfristen durchzusetzen – und schaden sich damit selbst, weil sie sich so Nachteile einhandeln, die nicht offensichtlich, aber sehr schädlich sind.
- Zweitens ist allen gemeinsam die Tendenz, sich zu überschätzen. Vielleicht ist diese Schwäche unser größtes Problem: Wären sich alle Beteiligten ihrer Beschränktheit bewusst, so wären sie vor- und umsichtiger. Sie würden öfter Rat suchen, sich stärker um präzise Formulierungen der Anforderungen und Ziele bemühen und viel mehr prüfen. Dass sich jemand bewusst entscheidet, schlechte Qualität zu produzieren, dürfte selten sein. Sehr oft aber werden Termine und Kostenschranken akzeptiert, die es faktisch nicht zulassen, gute Qualität zu liefern.

Wenn ein Koch versucht, aus altem Gemüse und nicht mehr ganz frischen Fischen in 30 min ein Festessen zuzubereiten, wird es den Gästen nicht schmecken, und der Koch wird seinen Job bald verlieren; wenn ein Softwareentwickler das Gleiche tut, erscheint das normal. Es mangelt uns an guten Köchen wie an Feinschmeckern.

10 Wir Qualitätshüter

Stanislaw Jerzy Lec, 1909–1966 (aus: Unfrisierte Gedanken):

Das Gewicht eines Problems wird brutto notiert. Wir sind darin inbegriffen.

Vermutlich wimmelt es unter den Lesern dieses Beitrags von Leuten, die die Qualität mit dem ganz großen Löffel gegessen haben. Gut so! Dann können wir uns ja darauf verlassen, dass sie (Sie!) nur Software hoher Qualität entwickeln, dass sie Texte verfassen mit präzise zitierten Quellen und zwingenden Schlussfolgerungen auf der Basis solider Empirie, vorbildlich in Rechtschreibung, Grammatik und Interpunktion.

Denn es wäre doch sehr peinlich, die Gretchenfrage in aktualisierter Form beantworten zu müssen:

Nun sag, wie hast du's mit der Qualität? Du bist ein herzlich guter Mann, allein ich glaub, du hältst nicht viel davon.

11 Zusammenfassung

- Für jeden von uns Einzelnen ist und bleibt die Qualität A, die transzendente Qualität, ein zentraler Aspekt unseres Lebens.
- Wenn wir aber über die Qualität kommunizieren, sie domestizieren wollen, müssen wir sie in beschreibbare, möglichst messbare Aspekte zergliedern. Das führt zur Qualität B, der modellierten Qualität.
- Die transzendente Qualität ist das Wildpferd, das uns begeistert, sich aber weder reiten noch vor den Wagen spannen lässt. Die modellierte Qualität ist das Arbeitspferd, das uns ernährt. Natürlich hilft es uns nicht, wenn es die wesentlichen Stärken seiner wilden Verwandtschaft (vor allem Kraft, Ausdauer und Intelligenz) verloren hat, als es gezähmt wurde. Mit anderen Worten: Die modellierte Qualität bedarf stets der Kontrolle und Bestätigung durch die transzendente Qualität.
- Oft bekommen wir nicht die Qualität, die wir brauchen, wünschen, in Auftrag gegeben oder einfach nur erwartet haben. Eine wichtige Ursache dafür ist, dass wir die Probleme der Softwarebearbeitung unter- und uns überschätzen. Das gilt gleichermaßen für Kunden, Entwickler, Projektleiter und Manager. Mehr Ehrlichkeit, Disziplin und Demut wären gut. Das hat Dijkstra gemeint, als er vom demütigen Programmierer sprach [Dijkstra 1972]:

We shall do a much better programming job, provided that we approach the task with a full appreciation of its tremendous difficulty, provided that we stick to modest and elegant programming languages, provided that we respect the intrinsic limitation of the human mind and approach the task as Very Humble Programmers.

Aber Disziplin und Demut verlangen von uns nicht, der Bürokratie zu dienen. Und: Die Protagonisten der Qualitätspartei sollten stets mit gutem Beispiel vorangehen!

Literatur

Alle URLs wurden im Dezember 2016 überprüft.

- [Boehm et al. 1976] Boehm, B. W.; Brown, J. R.; Lipow, M.: Quantitative Evaluation of Software Quality, ICSE 1976: 592-605.
- [Cunningham 1992] Cunningham, W.: The WyCash Portfolio Management System, 1992, <http://c2.com/doc/oopsla92.html>.
- [Dijkstra 1972] Dijkstra, E. W.: The humble programmer. Turing Lecture. Commun. ACM 15 (1972), 10: 859–866.
- [Garvin 1984] Garvin, D.: What does 'product quality' really mean? Sloan Management Review, Fall 1984, S. 25-45.
- [ISO/IEC 25000] ISO/IEC 25000 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. Deutsch: System und Software-Engineering – Qualitätskriterien und Bewertung von System- und Softwareprodukten (SQuaRE) – Leitfaden für SQuaRE. DIN-Normenausschuss Informationstechnik und Anwendungen (NIA), 2014. Vgl. ISO/IEC 25000 in Wikipedia.
- [Kitchenham & Pfleeger 1996] Kitchenham, S.; Pfleeger, S. L.: Software quality: the elusive target. IEEE Software, 13(1), 1996, S. 12-21.
- [Ludewig & Lichter 2013] Ludewig, J.; Lichter, H.: Software Engineering – Grundlagen, Menschen, Prozesse, Techniken. 3. Aufl., dpunkt.verlag, Heidelberg, 2013, ISBN 978-3-86490-092-1.
- [Pirsig 1974] Pirsig, R.: Zen and the Art of Motorcycle Maintenance: An Inquiry into Values (ZAMM), first published in 1974; William Morrow & Company. Eine der deutschen Ausgaben: Zen und die Kunst ein Motorrad zu warten. Ein Versuch über Werte. Mit einem Nachwort des Autors: Zehn Jahre nach Erscheinen der ersten Ausgabe. Fischer-Taschenbuch-Verlag, Frankfurt 2005.
Siehe auch: MOQ = Metaphysics of Quality! (<http://moq.org/>)