

Softwarequalität als ökonomische Notwendigkeit

# Unerschütterliches Qualitäts- bewusstsein

*Qualität ist wichtig! Das sagen alle  
Stakeholder eines Softwareprojektes.  
Die Frage ist nur: Was ist Qualität?*

**G**erald M. Weinbergs Definition lautet ganz simpel [1]: „Quality is value to some person“. Diese Wertschätzung einer Eigenschaft von Software wird gewöhnlich ausgedrückt durch die Bereitschaft, dafür Geld auszugeben.

Das ist genauso natürlich wie simpel – solange der Qualitätsempfänger feststellen kann, ob das Softwareprodukt tatsächlich die gewünschte Qualität hat. Für funktionale Anforderungen ist solche Prüfung leicht möglich. Das Ergebnis ist binär: Entweder kann die Software etwas oder sie kann es nicht. Ein kleinwenig schwieriger ist es für Effizienzanforderungen wie Performance, Sicherheit, Usability usw. Letztlich ist das Ergebnis dort jedoch auch binär: Entweder weist eine Funktionalität die geforderte Effizienz auf oder nicht.

## Wandelbarkeit

Anders hingegen für die Wandelbarkeit des Codes und die Produktivität des Softwareentwicklungsteams. Es ist unzweifelhaft, dass Kunden eher bereit sind, für ein produktives Team Geld auszugeben als für ein weniger produktives. Ebenso erwarten sie, dass der hergestellte Code sich zukünftig leicht an neue und veränderte Anforderungen anpassen lässt. Wandelbarkeit und Produktivität sind also relevante Qualitäten – allerdings von unbekannter Größe. Beide lassen sich nicht durch den Kunden messen. Es gibt keine binäre Antwort auf die Frage „Ist die Software so wandelbar, wie ich als Kunde es mir wünsche?“ Weder kann diese Qualität direkt gemessen werden, noch hat der Kunde eine Vorstellung davon, wie viel Wandelbarkeit er überhaupt bestellen sollte.

Einerseits soll die Qualität der Wandelbarkeit also hoch sein – andererseits ist stets unklar, ob sie es auch ist. Das Ergebnis dieser Situation ist regelmäßig, dass Ressourcen auf Qualitäten fokussiert werden, die leichter bzw. überhaupt zu messen sind.

Angesichts der mangelnden Messbarkeit von Wandelbarkeit und Produktivität muss der Kunde darauf vertrauen, dass die Softwareproduktion selbstständig für sie sorgt. Dazu ist die auch durchaus bereit, soweit sie eine Vorstellung davon hat, wie das zu tun ist. Allerdings gerät das immer wieder aus dem Blick, wenn das Bedürfnis des Kunden nach Funktionalität und Effizienz steigt. Unter dem Druck, leicht messbare Qualitäten herzustellen, knickt die Softwareproduktion ein. Maßnahmen zur Herstellung und Erhaltung von Wandlungsfähigkeit und

Produktivität erodieren bis zur Aufgabe. Statt den Qualitätsweg in Bezug auf diese Anforderungen zu beschreiben, werden Abkürzungen genommen.

### Beispiele für Erosionen

Code Reviews werden aufgegeben, automatisierte Tests werden nicht aufgesetzt, überhaupt wird die Testdichte reduziert, Technologien werden ohne

Produktivität sind nicht „nice to have“, sondern überlebensnotwendig.

Wo Wandlungsfähigkeit und Produktivität erodieren, bricht der Produktionsfluss zusammen. Anforderungen fließen nicht mehr geschmeidig aus dem Backlog in den Code. Es entsteht vielmehr eine wachsende Zahl größerer und kleinerer Verwirbelungen im Produktionsfluss. Das Symptom sind Hilferufe [2] und Nachbesserungen. Einige Hilferufe sind sichtbar – Bug Reports von Anwendern –, weit mehr sind jedoch unsichtbar. Sie tarnen sich als scheinbar unvermeidbarer Austausch zwischen Teammitgliedern, Missverständnisse, Fehler des Kunden. Ebenso werden Nachbesserungen projiziert auf unklare Kunden oder schicksalhafte Umstände von Softwareentwicklung im Allgemeinen.



## Wo Wandlungsfähigkeit und Produktivität

erodieren, bricht der

Produktionsfluss zusammen.

Es entsteht eine wachsende Zahl größerer und kleinerer

Verwirbelungen im

Produktionsfluss. Das

Symptom sind Hilferufe und Nachbesserungen.

Ralf Westphal, freiberuflicher Berater, Projektbegleiter und Trainer für Themen rund um Softwarearchitektur und Teamorganisation

ausreichende Kenntnis eingesetzt, die Teamstärke wird kurzfristig erhöht, Retrospektiven werden ausgesetzt, die Umsetzung einer Anforderung wird unterbrochen zugunsten einer anderen. Das Muster dahinter: Um massiven Forderungen nach mehr Qualität bei Funktionalität und Effizienz nachzukommen, werden zunehmend lokale Optimierungen im Produktionsprozess eingeführt. Einzelne Stationen der Herstellung werden kurzfristig beschleunigt – mit gravierenden negativen Folgen für die Qualität langfristiger Wandelbarkeit und Produktivität.

Lokale und temporäre Optimierungen gehen immer auf Kosten des globalen und langfristigen Optimums. Die Softwareproduktion schneidet sich ins eigene Fleisch, wenn sie Erosionen zulässt. Qualität, oder genauer: innere Qualität, d.h. Wandlungsfähigkeit und

### Lokale Optimierungen

Abkürzungen und lokale Optimierungen zu Ungunsten von Wandelbarkeit und Produktivität insbesondere unter Druck des Kunden auf Funktionalität und Effizienz lassen die Produktionsmaschine heiß laufen. Die Spannung in allen Beteiligten steigt, das führt zu Konflikten, die weitere Ressourcen kosten. Schweißperlen stehen allen auf der Stirn, alle arbeiten hart – nur sorgt die harte Arbeit nicht dafür, dass das Projekt vorankommt. Im Gegenteil: Die Leistung sinkt trotz immer mehr Aufwand.

Das bedeutet, eine Softwareproduktion, für die Maßnahmen zur Herstellung und Erhaltung von Wandlungsfähigkeit und Produktivität nicht unverbrüchlich sind, wird weitaus teurer als notwendig. Das kann nicht im Interesse des Unternehmens und auch nicht des Kunden sein.

Verständlich wird das umso leichter, je mehr Softwareproduktion nicht als „kreatives Chaos“, sondern als Produktionsprozess verstanden wird. Der besteht aus distinkten Schritten, durch die Material fließt. Anforderungen werden in diesem Prozess in Code transformiert. Jeder der Schritte hat seinen Sinn. Jeder der Schritte hat Anforderungen an seinen Input und macht Versprechen über seinen Output. SLAs lassen sich nicht nur für den Betrieb von Software definieren, sondern auch für Softwareproduktionsschritte.

Dass solch ein Prozess nicht zum Wasserfall verkommt, muss am Anfang durch die Größe der zu verarbeitenden Anforderungen gesteuert werden. Dass solch ein Prozess nicht ins Stocken gerät, muss durch unerschütterliches Qualitätsbewusstsein in jedem Verarbeitungsschritt sichergestellt werden. Die Motivation jedes Einzelnen und das Vertrauen des Kunden sind allemal zu hohe Güter, als dass sie durch lokale Optimierungen für kurzfristige Gefälligkeiten aufs Spiel gesetzt werden sollten.

RALF WESTPHAL

#### Leseempfehlungen

- [1] Gerald M. Weinberg, Agile and the Definition of Quality, <http://secretsofconsulting.blogspot.de/2012/09/agile-and-definition-of-quality.html>
- [2] Ralf Westphal, Der dunkle Bruder des Feedback, <http://ralfw.de/2015/12/der-dunkle-bruder-des-feedback/>

WEB-TIPP:  
[www.ccd-school.de](http://www.ccd-school.de)