

FABRIK ODER HANDWERK?



„Als sich im vergangenen Jahr der chinesische Staatschef und die Bundeskanzlerin getroffen haben, wurde auch das Thema Industriespionage angesprochen, wie ich im Radio hörte. Hu Jintao verblüffende Antwort war, da müsse Deutschland sich nicht so sehr sorgen. Zwar sei es relativ einfach, die großen Strukturen einer Maschine nachzubauen, aber die Qualität deutscher Maschinen fange nicht erst bei den Ingenieuren, sondern schon bei den Schrauben an. Und bereits dort zeige sich ‚Made in Germany‘ in einer sonst auf der Welt unerreichten Präzision.“

Schöner, als **Michael „Pul“ Paulsen** es am Ende seines Artikels „Von Schrauben und Kartons: Wo Softwarequalität wirklich entsteht“ tut, kann man einen erwünschten Qualitätsstempel „Made in Germany“ auch in der IT wohl nicht umschreiben. Nur: Die deutschen Softwareschrauben scheinen mir nicht von der gewünschten Qualität zu sein. Anders ist eine zentrale und eben auch im Kern vernichtende Aussage aus dem Bericht zur iqnite-2011-Umfrage zum Thema „Testen“ von **Daniel Simon** und **Frank Simon** nicht zu deuten, die besagt:

- Testen ist zu teuer (=Kosten).
- Testen dauert zu lange (=Zeit).
- Testen verhindert nicht das Ausliefern kritischer Fehler (=Qualität).

Mit anderen Worten: Die aktuelle Testpraxis bringt nichts, hält alles auf und verschlingt dazu noch Geld und Personalressourcen. Schlimmer kann es eigentlich nicht mehr kommen, oder?

Und diesen Eindruck bekomme ich auch persönlich in vielen deutschen Softwarehäusern. Fast überall ist ein vom Management adressiertes Hauptproblem die Qualität. Kunden sind unzufrieden, wenn ein neues Release ausgeliefert wird, in den Callcentern des Supports laufen die Drähte heiß, die Liste der offenen Fehler umfasst Tausende von Einträgen, die niemand mehr wirklich überschaut, wesentliche Entwickler-Kapazitäten sind durch Bug-Fixing gebunden usw. Die Liste der unschönen Symptome lässt sich leicht fortsetzen.

Was ist denn eigentlich Softwarequalität? Ist die Software besonders gut, in der wir viele Fehler gefunden (und hoffentlich auch wirklich behoben) haben, oder doch die, in der wir wenige Fehler gefunden haben? Messen wir Kennzahlen wie Testabdeckung oder werten wir lieber die Beschwerdestatistiken aus, um valide Aussagen über die Softwarequalität zu treffen.

Ralf Westphal stellt in seinem Artikel fest: „Was ist das größte Kompliment, das einer Software gemacht werden kann? Ein Feature-Request. Denn wenn ein Kunde ein neues Feature möchte, dann gefällt ihm die Software so gut, dass er sie weiter betreiben möchte und sogar noch mehr damit tun will.“ Dumm ist in der Praxis eigentlich nur, dass dieses „Kompliment“ an den Softwarelieferanten, also der Feature-Request, zumeist in der langen Liste der Fehlertickets landet, weil dort – so die oft enttäuschte Hoffnung – mit höherer Priorität gearbeitet wird und sich damit die Qualitätsstatistiken weiter verschlechtern.

Wie können wir also eine bessere Qualität erreichen? Gegenwärtig gibt es offensichtlich unterschiedliche Strömungen. Den traditionellen Ansatz fasse ich einmal unter dem Begriff „Test Factory“ zusammen. **Johann Gietl** stellt in seinem Beitrag ein gelungenes

Beispiel vor, wie man Prozesse, Rollen und Werkzeuge in einem Nutzungskonzept zur Wirkung bringt. **Andreas Bartsch** und **Stefan Reise** gehen einen weiteren Schritt in diese Richtung, indem sie eine werkzeuggestützte Testplanung vorstellen.

Den Gegenpol zu diesem Ansatz finden wir bei den Vertretern des agilen Testens, das die Verantwortung für Qualität eher wieder zurück an die Entwicklungsteams gibt und auf eine frühe Zusammenarbeit von Spezifikatoren, Entwicklern und Testern setzt. **Markus Gärtner** und **Arne Rook** beschreiben den durchaus mit Überraschungen gespickten Einstieg vom fiktiven Tester Tobias in die Kanban-Hochburg Flow Work Ltd. Hier scheinen tatsächlich viele der Probleme in traditionellen Testbereichen gelöst zu sein. Und doch resümieren die Autoren: „In einer Kultur, in der Fehler und Probleme lieber versteckt werden, wird Kanban folglich nicht ohne weitere Unterstützung funktionieren.“

Oliver Laitenberger erklärt, warum in agilen Projekten Reviews wieder in Mode kommen und warum dieser manuelle QS-Ansatz vor allem zu Lerneffekten bei den beteiligten Menschen, aber auch in der Organisation führt. Auch **Helmut Pichler** und **Harry Sneed** sind auf dem agilen Pfad, wenn sie die „Testergetriebene agile Entwicklung“ fordern, und betonen dabei die Verantwortung des Testers für die unternehmensweite Integration.

André Janus schließlich betont, dass agile Softwareentwicklung in der Praxis mit Continuous Integration eine nützliche Basis und mit den entsprechenden Tools eine gute Infrastruktur für einen kontinuierlichen Messprozess und somit für ein agiles Qualitätsmanagement findet und baut damit wieder die Brücke zu dem prozess-, mess- und werkzeuggetriebenen Ansatz.

Die Clean-Code-Developer-Initiative schließlich, deren Werte **Ralf Westphal** im Interview erläutert, hebt – ebenso wie das Agile Manifest – den Nutzen für den Anwender als wesentliches Qualitätsmerkmal heraus. Sie geht aber noch einen Schritt weiter: „Scrum und Kanban wollen keine Aussage über Codierungspraxis und Codestrukturen machen. Auch dem Agilen Manifest ist das kein Anliegen. Doch irgendwer muss etwas dazu sagen, denn das Codieren erledigt sich nicht von allein und nicht jeder Code hat dieselbe Qualität in Bezug auf die Bedürfnisse des Kunden. Also versucht die CCD-Initiative, die Lücke zu schließen.“

Damit sind wir wieder bei den Schrauben und der sauberen handwerklichen Verarbeitung der deutschen Maschinen. Softwarequalität entsteht substanzvoll wirklich in erster Linie durch handwerklich sauberes und verantwortliches Handeln im Entwicklungsteam. Unterstützt wird das Ganze durch gute Werkzeuge, Prozesse und ein gut organisiertes Testmanagement. Basis für all das ist eine Kultur, in der aufgedeckte Fehler kein Makel sind, sondern eine Chance, um diese für die Zukunft zu verhindern.

Aber letztlich zählt, wie **Ralf Westphal** so treffend feststellt, nur der Effekt: „Erst wenn durch eine Veränderung der Praxis Bedürfnisse des Kunden besser befriedigt werden, ist etwas erreicht.“

Ihr Thorsten Janning