

Jenseits des Tellerrands – Teil 12: Wenn alles verkehrt läuft

Time's Arrow

■ VON FRANK WESTPHAL UND JOHANNES LINK

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.

Hinterher ist man immer schlauer, sagt der Volksmund. Im vorerst letzten Teil der Serie drehen wir den Pfeil der Zeit einfach um, und plötzlich ergibt so manches Sinn, was uns vorher als Abstrusität in der Softwareentwicklung erschien.

Johannes Link
(john.link@gmx.net)



Das Projekt begann mit einer puren Katastrophe. Zwei Tage nach Projektbeginn platzte unsere Kundin aufgeregt ins Büro – ich hatte gerade erst Bekanntschaft mit unserem Projektleiter Archie gemacht und saß mit ihm über dem Projektplan bei einer Tasse Kaffee zusammen. Ich erinnere mich noch, dass wir beide angesichts des engen Projektzeitplans völlig verzweifelt waren. Doch unsere Kundin wusste unsere Unsicherheit in dieser Minute auszunutzen. Sie knallte die Tür auf und schrie uns puterrot ins Gesicht: „Wir sehen uns vor Gericht.“ Archie entgegnete gelassen: „Kriegen Sie mal keinen Herzinfarkt, Frau Customley. Wir haben alles unter Kontrolle!“ Worauf sie kreidebleich wurde, „Packen Sie Ihre Sachen!“ kreischte und rückwärts aus der Tür war. „Ein dicker Anfang“, dachte ich still bei mir. Denn sie sollte Unrecht haben: Wir hatten wirklich alles im Griff. Und: Es sollte noch besser laufen in Zukunft. Viel besser ... Alles, was wir tun mussten, war

abwarten und ausreichend Kaffee in die Tassen spucken.

Weil Softwareprojekte immer pünktlich sind

Das Schöne an Softwareprojekten ist nämlich, dass man stets genau weiß, wie viel Zeit man benötigt. Unser Projekt wird z.B. am 30. November fertig sein, soviel ist gewiss; denn Softwareprojekte sind so pünktlich wie Heiligabend, nie zu spät und einem einfachen Countdown folgend: Erst haken wir Meilenstein Nummer vier ab, dann Nummer drei, zwei, eins, fertig! Teile und herrsche, ganz simpel. Der Nachteil ist, dass einem die Projekte oft sehr viel teurer zu stehen kommen, als man schätzen würde. Viele, viele Millionen lassen wir IT-Dienstleister es uns kosten, Individualsoftware für unsere Kunden entwickeln zu dürfen. Und tatsächlich kennen wir die genauen Zahlen bereits zu Beginn, wenn wir einen Brief mit der exakten Endsumme erhalten.

Zum Glück lassen die meisten Kunden vor Projektbeginn jedoch mit sich handeln. Häufig gibt es durch einen Gerichtstermin noch einen Rabatt – unter der Voraussetzung, dass wir mit der Software pünktlich sind (nie ein Problem!) und ein paar der fachlichen Anforderungen geflissentlich ignorieren. Weniger Geld zu verdienen stimmt die Kundenseite allerdings ziemlich übel. Softwareentwicklung ist und bleibt eben ein hartes Pflaster. Mir ist es ein absolutes Rätsel, paradox wie alles abläuft.

Nehmen wir zum Beispiel die anfängliche Projektphase: Die ersten drei Wochen kümmern sich alle Mann ganz hektisch um nichts anderes als „Bug-Fixing“. Der Name ist etwas irreführend, geht es doch eigentlich um Sabotage, mit dem Ziel, die Projektarbeit künstlich zu bremsen. Aber das darf man nicht laut sagen und schon gar nicht im Projektplan – eher ein völlig unrealistisches Projektprotokoll – erwähnen. Anschließend beginnt die Weiterentwicklung, in deren Verlauf wir unzählige Bug-Reports an unsere Qualitätssicherer und die Fachabteilung des Kunden austeilen mit der Aufgabe, die beschriebenen Softwarefehler nachzustellen. Sind alle Fehler gefunden, dann nehmen wir ihnen die Anwendung wieder weg und erklären den Beta-Test als beendet. Im anschließenden Alpha-Test haben wir ein „Capture & Replay“-Werkzeug eingesetzt, das wir jedoch schnell wieder für viel Geld verkauft haben, weil es nicht viel taugte. Natürlich stieg uns daraufhin sofort der Kunde aufs Dach, da wir uns eigtl. mehr um Javadoc und externe Dokumentation kümmern sollten. Kaum hatten wir damit angefangen, änderten sich die Prioritäten erneut. Welch ein Chaos ...

Geht Zeit, kommt Rat

Rien ne vas plus, nichts geht mehr. Unser Projekt scheint am Boden – kein Wunder nach so vielen Kursänderungen. Der Aufschwung jedoch kommt mit der Big-Bang-Integration unseres Nearshoring-Teilprojekts aus Tschechien. Zwar zahlten die Tschechen auch nicht viel für ihre Mitarbeit, dafür entfernten sie im Laufe der Zeit eine unglaubliche Menge an Code. Und es war auch überhaupt nicht schade drum, denn mit der Codequalität stand es vorher ... naja, Schwamm drüber, wir haben's überstanden. Und die Stimmung beim Abschied der Tschechen war wirklich gut. Gratulation! Solche Helfer bräuchten wir mehr!

Anzeige

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

Danach ging es weiter aufwärts. Aus den Integrationsproblemen vor dem Eingriff der Tschechen hatten wir unsere Lehre gezogen und integrierten fortan sehr viel häufiger. Auch gaben wir den CVS-Branch auf und hatten eine Reorganisation. Da der Kunde seine Change Requests zurückzog, legten wir Entwicklungs- und Wartungsteam zusammen. Ich war ja von Anfang an kein Fan dieser Trennung.

Der Sommer kam, und Archie wurde nach einer Reihe gemeiner politischer Intrigen von heute auf morgen auf die Straße gesetzt. Zum Schluss wusste er selbst nicht mehr über sein eigenes Projekt Bescheid, vergaß alle unsere Namen und verlor seine Zutrittskarte – ein tragischer Fall. Wir hatten den dritten Meilenstein verfehlt, waren neun Wochen zu früh; Riesenaufregung wegen der schlechten Planung; die Geschäftsleitung hielt uns eine Standpauke. Archie's letzte Amtshandlung war die Entlassung von sieben unnützen Entwicklern, die tags drauf von drei Headhuntern abgeholt wurden. Phillip, der neue Projektleiter, war ein Pfundskerl und hatte von Anfang an unser vollstes Vertrauen. Die Geschäftsleitung forderte zwar die Entlassung weiterer Entwickler, doch Phillip hielt uns den Rücken frei. „Neun Frauen bekommen ein Baby nicht in einem Monat“, blieb er standhaft. Womit er Recht behielt, denn schwanger wurde tatsächlich niemand ... Zudem hatten wir ohnehin keine Frauen an Bord.

Niemand fragt hier nach dem Warum

Wir verdummen ... Je länger wir im Projekt sind, desto betriebsblinder werden wir. Probleme, die wir schon längst aus dem Weg geräumt hatten, rissen auf einmal wieder auf. Wir vergaßen alles, was wir schon gelernt hatten. Zu allem Überfluss wurde uns auch noch das ISO-9000-Zertifikat aberkannt. Noch Tage, nachdem der Auditor schon wieder fort war, lief unser Qualitätsbeauftragter wie ein aufgedrehtes Huhn durch die Büros und verriet uns, was die richtigen Antworten gewesen wären. Hätte er sich nur früher bemüht – verdrehte Welt! Wir nahmen daraufhin die Dilbert-Comics von den Bürowänden. Witze übers Projektmanagement waren uns inzwischen vergangen. Wir fassten uns an die eigenen Nasen

und lachten uns über uns selbst schlapp. Humor ist überhaupt der einzige Weg, heil und gesund durch Projekte zu kommen ... ach, was sag ich da, durchs ganze Leben.

Mit nahezu allen Projektmitarbeitern verbindet mich inzwischen eine innige Brieffreundschaft. Phillip zum Beispiel e-mailt mir jeden Abend einen etwas geschönten Statusbericht mit den Arbeitsergebnissen zum Tage. Aufgaben zu delegieren ist mir aber immer noch am liebsten: Man ziehe eine E-Mail aus seinem Papierkorb und sende sie einem anderen Projektteilnehmer zu – ziemlich wahllos, wie mir scheint. Oder: Man lade alle auf dem eigenen Schreibtisch angestaubten oder aus dem Papierwolf gezogenen Dokumente bei seinen Kollegen ab – nicht jedoch, ohne vorher die braunen Ringe mithilfe von Kaffeetassen zu entfernen. „FYI“ steht drauf, Kommunikation ist alles!

Man tut, was man am besten kann, nicht, was am besten zu tun wäre

Was ich an meinen beiden Bürokollegen – der eine studiert noch, der andere ist Freelancer – bewundere, ist ihre Fähigkeit, Klarheit im Code und in der Anwendung zu schaffen. Noch am Tag zuvor sieht unser GUI wie ein Flickenteppich aus; niemand findet sich in der Anwendung zurecht, und auch der Code ähnelt einem Kafka-Roman. Solche Codestellen krepeln die zwei in wenigen Tagen vollständig um. Danach ist nicht nur der Code einfach und klar strukturiert, sondern auch die überflüssigen Features sind verschwunden. Klar, dass die beiden alle paar Tage neue Problemstellen vorgesetzt bekommen.

Der Verdienst anderer Kollegen ist mehr als zweifelhaft. Da gab es zum Beispiel so ein seltsames, aber extrem dynamisches Duo: Frank Testfall und Johannes Verbindung. Sie begannen ihr zweimonatiges Zwischenspiel mit einer wunderbar strukturierten Codebasis, um diese dann Schritt für Schritt vollständig aus dem System zu entfernen. Kein Wunder, dass der Respekt, den wir zu Beginn für diese Eigenbrödler hatten, schlussendlich in kritischer Verachtung endete. Ich muss zugeben, dass sich hier unser oberster Boss prophetisch erwies, als er auf Phillips Äußerung „Aber die beiden waren schneller und erfolgreicher als alle anderen zusam-

men!“ antwortete: „Pair Programming ist doch rausgeworfenes Geld!“ Warum die beiden dann trotzdem am nächsten Tag auf den besten Teil unseres Codes losgelassen wurden, ist mir bis dato ein Rätsel.

Subtrahiere null von null und du erhältst null

Ach ja, ich habe mich noch gar nicht vorgestellt: Bis vor kurzem war ich der so genannte Architekt des Teams. Faktisch bestand meine Aufgabe darin, die vielen Diagramme und Modelle, die es von Anfang in unserem Projekt gab, Schritt für Schritt aus dem Repository zu entfernen und schließlich ganz zu löschen. Das mag sich zunächst unsinnig anhören, bei näherer Betrachtung ergibt es jedoch Sinn. Denn die Diagramme wurden von niemandem außer mir selbst jemals angeschaut. Und eines der zentralen Feng-Shui-Prinzipien lautet, dass man Dinge, die man ein ganzes Jahr lang nicht benötigt hat, loswerden soll. Ich denke, ich liege mit meiner Vermutung recht gut, dass Softwareentwicklung viel mit fernöstlichen Weisheiten und Lehren gemein hat.

In meiner Rolle als Architekt durfte ich mir auch immer wieder Codeteile zu Gemüte führen. Codeteile, die mir seltsam vertraut vorkamen, so als würde ich später noch nähere Bekanntschaft mit ihnen schließen. Meist geschahen diese Inspektionen unmittelbar nachdem ich ein entsprechendes UML-Diagramm endgültig weggeworfen hatte; auch das mit gutem Grund, denn Parallelen zwischen Code und Modell waren nur oberflächlicher Natur.

Du musst gemein sein, um nett zu sein

Letzte Woche hat man uns die hochauflösenden Flachbildschirme weggenommen und uns stattdessen zentnerschwere Flimmerkisten vor die Nase gesetzt. Anschließend gingen unsere ergonomischen Bürosessel im Tausch gegen Nullachtfüß-zehn-Sitzmöbel verloren. Alles aus dem Sperrmüllcontainer vorm Haus gefischt; muss man sich mal vorstellen. Einfach widerlich ... doch die Firma war noch jung und brauchte offensichtlich das Geld, um dem Kunden die erste Rate zahlen zu können.

Übrigens hat mich Phillip mittlerweile als Architekt beerbt. „Jetzt kannst du endlich zeigen, was du drauf hast!“, sagte er

mir; und das kurz bevor unser Boss den offensichtlich ironisch gemeinten Satz brachte: „Euer Projekt braucht einen eigenen Projektleiter, ich kann das nebenher nicht mehr schaffen.“ Ja und dann übernahm er selbst die Rolle des Projektleiters. Management by Sarcasm.

Weil ich Entwickler bin, entwickelt sich alles, was ich tue, schon irgendwie

Tatsächlich bin ich jetzt beim eigentlichen Kern der Softwareentwicklung gelandet: Ich darf programmieren – getreu dem Motto: Architect also implements. Was mir als Architekt noch ganz einfach erschien, wird nun zu einer echten Tüftelei: Welche Zeilen des Codes sind ausreichend unwichtig oder gar überflüssig, sodass ich sie entfernen kann. Lustigerweise stelle ich am Ende meist fest, dass der ganze Code weg muss; aber solche Entscheidungen darf man ja nicht leichtfertig treffen!

Manche Kollegen gehen die gleiche Aufgabe übrigens völlig anders an. Zunächst löschen sie – wie mir scheint willkürlich – irgendwelche Stellen im Code, um anschließend festzustellen, dass unsere Unit-Tests nicht mehr laufen. Klar, um das zu zeigen sind die Tests ja da! Der Hammer: Nun löschen sie einfach die fehlschlagenden Testfälle und alles *scheint* wieder gut – kann man sich *noch* unprofessionelleres Vorgehen vorstellen?

Als Programmierer willst du vor allem eines: den Code jede Minute ein kleines bisschen besser hinterlassen, als du ihn selbst vorgefunden hast: Kaizen – das ist die höchste aller Tugenden, und darin waren wir Weltmeister [Anm.: Kaizen ist eine japanische Managementphilosophie und steht für die unaufhörliche Veränderung zum Besseren]. Unser Code nahm mit jedem Tag eine immer einfachere Form an. Schlecht riechende Codeteile wurden radikal ausgerottet. Unser Streben nach nahezu buddhistischer Einfachheit trug uns sogar so weit, dass wir unsere Drittanbietersoftware regelmäßig downgraden konnten, weil wir uns von den angebotenen Features frei gemacht hatten. Wir brauchten all das einfach nicht mehr.

Würde die IT-Branche einen Architekturpreis vergeben, ich bin mir sicher, Phillip hätte ihn bekommen. Er war so stolz auf sein Machwerk, dass er es für die

Nachwelt konservieren musste. Während er also wochenlang an seinen Kraftpunktfolien schnitzte, gingen wir noch ein letztes Mal durch die Anforderungsdokumente. Eigentlich wollten wir nur noch abschließend sicherstellen, dass wir wirklich zu 100 Prozent verstanden hatten, was sich der Kunde wünschte. Die Dokumente spiegelten jedoch keinesfalls wider, was wir zuvor in monatelanger Kleinstarbeit entwickelt hatten. Wir spielten unsere Unsicherheit herunter und hofften, dass es niemandem auffallen möge.

Der Kunde liebt uns, er liebt uns nicht

Und tatsächlich sollte unser partnerschaftliches Verhältnis mit dem Kick-off-Meeting seinen offiziellen Höhepunkt erreichen. Nachdem wir gemeinsam mit dem Kunden durch die Kneipen gezogen waren, überreichten wir ihm zeremoniell die finale Projektdokumentation. Das soll's gewesen sein!? Schon eine Woche später verkaufte die Firma sowohl den

Application Server (Software wie Hardware) als auch unsere Modellierungs- und Entwicklungswerkzeuge. Wir hatten auch wirklich genug Ärger damit!

Ich gehe seitdem zur Hochschule und studiere Informatik. Phillip und die anderen ehemaligen Kollegen habe ich nie wieder gesehen. Vielleicht ist das auch gut so. Fremd geworden, wie wir uns am Ende waren.



Frank Westphal ist freier Softwareentwicklungsscoach, Berater und Trainer. Seine Hauptschwerpunkte sind XP, Lean und TDD. Mit dem Tonabnehmer bietet er einen Podcast zum Thema Agile Softwareentwicklung.

Derzeit programmiert er Web 2.0-Anwendungen mit AJAX und Ruby on Rails. Kontakt: www.frankwestphal.de.



Johannes Link ist Softwareentwickler, Bücherleser und Buchschreiber. Er arbeitet bei der 100world ag. Johannes dankt allen Autoren der Serie für ihr Engagement und den Lesern des *Java Magazins* für Geduld und Feedback. Sein Webtagebuch findet sich auf jlink.blogger.de.

Links & Literatur

[1] „Time's arrow or the Nature of the Offence“ – genialles Buch von Martin Amis (Vintage, 2003)

Anzeige