



Welcome to the Machine

▶ Bob, der Berater, fühlt sich nicht gerade wohl in seiner Haut, als er im geheiligten Konferenzraum vor den versammelten Vertretern des Managements steht. Ständig wippt er mit dem Körper hin und her, wandert durch den Raum und spielt nervös mit seinem iPhone. Drei Monate zuvor hat ihm die Firmenleitung der JWD (Java World Domination) den Auftrag erteilt, eine genaue Untersuchung darüber zu erstellen, warum deren frisch gegründete System-Engineering-Truppe so viele Projektfehlschläge verzeichnen musste. Eine Spurensuche bei der Entwicklungsabteilung ist natürlich immer höchst unerfreulich. Letztendlich hat Bob aber dank der Mitarbeiter einige sachdienliche und hilfreiche Hinweise zusammengetragen.

Der Stundenzeiger rückt auf 9 Uhr vor. Das große Schaulaufen vor den „Schlupsträgern“ kann beginnen. Let's powerpoint! ...

▼ Im Projekt Pandora sollte eine Armbanduhr für die Live-Überwachung von Gesundheitsdaten wie Pulsfrequenz und Blutdruck entstehen. Um die Stückkosten zu senken, hatte die Projektleitung kurzerhand einen Speicherchip durch eine billigere Variante mit weniger Kapazität ersetzt. Ein genialer Schachzug, so erschien es zumindest den Verantwortlichen. Bei einer anvisierten Stückzahl von 5 Millionen Geräten führte jeder eingesparte Euro schließlich zu einer satten Kostenreduktion. Allerdings sahen sich mangels ausreichenden Speichers die Entwickler nun gezwungen, zusätzliche Maßnahmen in der Software durchzuführen, etwa die ständige Kompression von Daten. So kam es, wie es kommen musste. Durch die notwendige Datenkompression lief die Software deutlich träger, sodass

recht zügig die Forderung nach einem schnelleren Prozessor aufkam. Dummerweise hätte das die Stückkosten erst recht in exorbitante Höhen getrieben, weshalb sich die Projektleitung letztendlich gezwungen sah, wieder den teureren Speicherchip einzusetzen. Durch derlei Störmanöver hatte sich mittlerweile eine Projektverzögerung von einem halben Jahr ergeben, was der Kunde als höchst inakzeptabel betrachtete und den Auftrag, ohne lang zu fackeln, an eine Konkurrenzfirma vergab. Moral von der Geschichte: Falsches Sparen lohnt sich nicht!

▼ Auch das Projekt „jFone 42“ für den Kunden Bulb stand unter einem denkbar schlechten Stern. Um dem ungeheuren Wettbewerbsdruck zu begegnen und das neue Smartphone jFone 42 schnellstmöglich am Markt zu positionieren, schreckte die Entwicklungsleitung selbst nicht vor unkonventionellen Maßnahmen zurück. So etablierten die Verantwortlichen drei separate Teams, je eines für Softwareentwicklung, Hardwareentwicklung und Gehäusedesign. Damit sich diese nicht all zu sehr in die Quere kamen, ließ man die Teams unabhängig arbeiten und minimierte gezielt die Aufwände für Koordination und Integrationstests. Schließlich hatte Bulb bereits die Vorgängermodelle des jFones erfolgreich auf den Markt gebracht. Für die Softwareentwickler hatte man auch noch eine exzellente Simulation der Hardware durch Softwarekomponenten anfertigen lassen. Was sollte da noch schiefgehen? Kurze Zeit später konnte der CEO von Bulb der breiten Öffentlichkeit stolz das neue Wunderwerk präsentieren. Dumm nur, dass die ersten Konsumenten eher ihr blaues Wunder erlebten, als sie feststellen musste, dass ein jFone 42 zwar mit einer beträchtlichen Menge unnützer Gimmicks aufwarten konnte, sich aber bei seiner Hauptaufgabe, dem Telefonieren, als eher zickig erwies. Am Ende kristallisierte sich heraus, dass die Wurzel allen Übels aus fehlender Absprache und unzureichenden Integrationstests bestand. Und dabei geriet dieses winzig kleine, aber unbedeutende Antennenproblem doch glatt aus dem Blickfeld. Manchmal wird aus einer kleinen Mücke eben doch ein Elefant. Moral von der Geschichte: Entwicklerquarantäne funktioniert eben nicht!

Bob illustriert noch Dutzende weiterer Negativbeispiele, und erläutert, welche Schlussfolgerungen sich daraus ergeben. Am Ende der Präsentation herrscht zunächst betretenes Schweigen im Raum. Dann kommt Applaus auf, erst leise, dann immer heftiger, ganz nach dem Motto, „Wir haben verstanden“. Bob wischt sich den Schweiß von der Stirn. Ihm fällt nicht nur ein Stein vom Herzen.

Was lernen wir Entwickler daraus? Systemnahe Softwareentwicklung weist zwar viele Gemeinsamkeiten mit Anwendungsentwicklung auf, aber andererseits auch deutliche Unterschiede. Etwa bezüglich der finanziellen Aspekte, der Koordination von verschiedenen Disziplinen wie Hardware und Software, der weltweiten Mobilität von eingebetteten Geräten mit unterschiedlichen Versionsständen oder hinsichtlich Lebenszyklus & Wartung von Produkten. Wer diese Unterschiede ignoriert oder womöglich auf die leichte Schulter nimmt, provoziert garantiert eine Bruchlandung. Daraus können durchaus außerordentlich teure Fehlschläge resultieren, was nicht nur die Beispiele der Ariane 5 (naive Nutzung der Software aus der Vorgängerversion, 2002) und des Mars Orbiters (Nutzung verschiedener physikalischer Einheiten in verschiedenen Softwaremodulen, 1999) belegen. „Lernen aus Fehlern“ erweist sich also immer als nützlich. Umso besser, wenn man dabei aus den Fehlern anderer lernen kann.

Die aktuelle Ausgabe widmet sich dem Schwerpunktthema Industrielles und Embedded Java. Sie enthält dazu einige praxisrelevante Beiträge namhafter Autoren. Wohl gemerkt, nicht nur aus der Theorie, sondern auch aus der Praxis. Wie sagte Einstein doch so trefflich, „In der Theorie sind Praxis und Theorie gleich, in der Praxis sind sie es nicht“.

In diesem Sinne viel Spaß mit dem vorliegenden Heft

Prof. Dr. Michael Stal

PS: Jede Ähnlichkeit mit lebenden Personen, Unternehmen oder tatsächlichen Ereignissen ist natürlich rein zufällig und völlig unbeabsichtigt!