



Von der Wiege bis zur Bahre

großer und komplexer Unternehmensanwendungen lauern fast überall Fettnäpfchen und Fallen. Ist anfangs auch nur wenig Sand im Getriebe, so münden diese Projekte dennoch oft in einem Desaster, vielleicht nicht unbedingt als Totalschäden, aber als Notfälle für die Intensivstation oder zumindest als Kandidaten für eine ausgiebige Reha-Maßnahme.

Dabei starten die Projekte meistens mit großem Enthusiasmus und einer gehörigen Prise Selbstüberschätzung. Ich denke exemplarisch an die Aussage eines britischen Ingenieurs vor rund einem Jahrhundert: „Nicht einmal Gott kann die Titanic zum Sinken bringen“. Als weiterer Misserfolgskfaktor ist das „Problem der Fürstentümer“ auszumachen, bei dem Organisationseinheiten sich nur um ihre eigenen Belange kümmern und Schutzwälle gegen die anderen Projektbeteiligten errichten. In einem großen Enterprise-Projekt hat das letztendlich zu einer katastrophalen Architektur, inkompatiblen Anwendungen und zum technischen Overkill dank Integration sehr vieler (unnötiger) Technologien und Produkte geführt. Das Ganze gepaart mit ineffizienter und ineffektiver Kommunikation sowie fehlendem Kompetenzaufbau – fertig war das Rezept zum Unglücklichsein. Bei der Crime Scene Investigation konnten wir noch zahlreiche weitere Beweise sichern und gerichtsarchitektonisch verwerten. Dass es freilich kaum bewährte Standardlösungen und -methoden für „einzigartige“ Großprojekte gibt, steht auf einem anderen Blatt.

Sollte ein Projekt misslingen, ist Demenz, Verstecken oder Verdrängen daher der falsche Weg. Stattdessen muss eine Post-Mortem-Analyse erfolgen, um ähnliche Fehlentscheidungen in Zukunft zu vermeiden. Das gilt durchaus auch für die mehr oder weniger desaströsen Fehler anderer Projekte. Oder wie es Henry Petroski ausführlich in einem seiner Bücher zu erläutern pflegt, *„Learning from Failure“* sollte man als Selbstverständlichkeit und nicht als lästige Pflicht betrachten.

„Bill Blake – Was hanged by mistake.“
(Grabinschrift aus Colorado, USA)

Happy Birthday

Zwei Jahrzehnte sind vergangen, seitdem die Erfolgsgeschichte von Java ih-

ren Anfang nahm. Am 23. Mai 1995 betreten John Cage von Sun Microsystems und Marc Andreessen, Mitgründer von Netscape, gemeinsam die Bühne der SunWorld Conference und stellen dem erstaunten Publikum offiziell die Java-Technologie vor. An diesem denkwürdigen Tag begann der „Siegesszug“ der Java-Applets – an Java für Embedded- und Enterprisesysteme oder für Desktopanwendungen war damals noch nicht zu denken. Spätestens das Erscheinen der konkurrierenden Microsoft-.NET-Technologie Ende des Jahres 2000 ließ sich als untrügliches Indiz für den durchschlagenden Erfolg der Java-Technologie betrachten.

Nur kurze Zeit nach Javas Geburt trugen JavaSPEKTRUM für den deutschsprachigen und Cousin Java REPORT für den englischsprachigen Markt ihr Scherlein zur weltweiten Verbreitung und Akzeptanz der Plattform bei.

Im Gegensatz zum Chefredakteur schaut diese auch heute noch nicht alt aus, sondern gilt als etablierte, reife und innovative Technologie für viele Anwendungsbereiche. Nicht zuletzt erwiesen sich die Entwicklung alternativer JVM-Programmiersprachen à la Groovy, Clojure oder Scala und die schier unbegrenzte Verfügbarkeit von Entwicklungswerkzeugen, Literatur und Schulungen als Erfolgsgarantien für das Java-Universum. Inzwischen gilt die Plattform nicht nur als eine Technologie, sondern fast schon als Lebensphilosophie. Und da das Gute stets unter dem Evolutionsdruck des Besseren steht, dürfte uns Java noch einige Jahre begleiten. Schließlich gehört der Erfolg den Tüchtigen.

In diesem Sinne, „Happy 20th Anniversary, Java!“

Das Redaktionsteam von JavaSPEKTRUM wünscht Ihnen ein erfolgreiches, glückliches, sonniges und gesundes neues Jahr 2015. Ebenso viel Freude beim Lesen unseres Fachmagazins.

Für Sie aus feinsten Zutaten kredenzt, präsentieren wir hiermit die erste Ausgabe im noch jungfräulichen Jahr.

Ihr ...

Michael Stal

Rest in Peace

*“In memory of Anna Hopewell -
Here lies the body of our Anna
Done to death by a banana
It wasn't the fruit that laid her low
But the skin of the thing that made her go.”*
(Grabinschrift aus den USA)

Erinnern Sie sich noch an Taligent (*Talent + Intelligent*), das amerikanische Joint Venture von Apple und IBM in den späten Achtzigerjahren des letzten Jahrhunderts, dessen Ziel in der Entwicklung des objektorientierten Betriebssystems Pink bestand? Nein, aber doch sicher an die Softwareentwicklung für die Ariane 5-Rakete mit ihrem buchstäblich explosiven Höhepunkt? Oder an das deutsche Lkw-Mautsystem Toll Collect, das seinen Dienst nach zahlreichen Geburtswehen und mit deutlicher Verspätung aufnehmen konnte? Ganz zu schweigen von der deutschen Gesundheitskarte, mit ihrem schon für 2006 geplanten Siegeszug? Von vielen anderen Beispielen haben Sie wahrscheinlich noch nie gehört: beispielsweise von der Softwaremodernisierung des U. S. Internal Revenue Service, die 1997 nach Kosten von 4 Milliarden (!) US-Dollar zu Grabe getragen wurde. Oder das Projekt Advanced Automation Systems der U.S. Federal Aviation Administration, das mit rund zwei Milliarden US-Dollar Schaden Schiffbruch erlitt. Natürlich gibt es zahlreiche Leichen im Keller fast aller Entwicklungsabteilungen, aber wir wollen uns nicht weiter als „Grabschänder“ betätigen.

Wenn wir uns also über Projekte wie den neuen Berliner Flughafen oder die Hamburger Elbphilharmonie echauffieren, sollten wir auf das laute Geräusch zerberstender Scheiben im eigenen Glashaus achten, um nicht in die Scherben zu treten. Gerade bei der Entwicklung