



Unlängst ergab sich für mich die einmalige Gelegenheit, die überaus interessante Biografie eines indischen Java-Architekten kennenzulernen. Der erste Kontakt kam durch einen befreundeten Guru zustande. Ich besuchte den Architekten in seinem kanadischen Entwicklungslabor, wo er sich mir als „Pi“ vorstellte. Diesen Kosenamen erlangte er in seiner Studienzeit, als er sämtliche Klassen der Java UI APIs auswendig aufzählte, ganz zum Erstaunen seiner Kommilitonen. In Java verliebte er sich schon 1996, wie er mir mit strahlenden Augen erzählte. Leider war er wegen seines jugendlichen Leichtsinns recht unschlüssig, welchem UI-Paradigma er seine Aufmerksamkeit schenken sollte. Kurzerhand entschied er sich deshalb, sämtliche UI-Konzepte zu erlernen und, schlimmer noch, sie auch in allen Projekten einzusetzen. Sein Vater, ein erfolgreicher IT-Unternehmer, hatte ihm allerdings eines Tages eindrucksvoll demonstriert, dass schlechte Bedienbarkeit aus Benutzern gefährliche Raubtiere machen kann. Das nahm sich der junge Pi von jenem Tag an sehr zu Herzen.

Verließ Pis Ausbildung anfangs noch recht geradlinig, kam es eines Tages zu einem einschneidenden Ereignis. Wegen der Finanzkrise sah sich sein Vater nämlich gezwungen, eine Dependence in Kanada zu gründen. Schon bald sollte das erste große Kundenprojekt in Montreal starten, für das Pi als Usability-Architekt auserkoren war. Leider erwies sich der französische Projektleiter als inkompetenter und grobschlächtiger Zeitgenosse, dessen Entscheidungen dem Entwicklungsteam bisweilen großes Unbehagen bereiteten. Statt Eclipse, JUnit und FX, hieß es nur

Life of Pi – Projektcrash durch UI

„Kommandozeilen-Tools oder nix“, und statt der Gebote für gutes Design herrschte eine „Basst scho“-Mentalität. So kam es, wie es kommen musste. Das Projekt geriet zuerst in unruhiges Fahrwasser, kam ins Kentern und versank schließlich trotz aller verzweifelten Rettungsversuche.

Während sich das indische Familienunternehmen daraufhin komplett aus dem Projekt zurückziehen musste, konnte sich einzig und allein Pi ins Boot des Folgeprojekts retten. Als er sich aber anschickte, die neue Situation zu analysieren, erschien unvermittelt UIsyphus, seines Zeichens Gott der höllischen Benutzerqualen. Und jedes Mal, wenn Pi fortan den Versuch unternahm, ein tragfähiges UI-Konzept zu entwerfen, stellte sich UIsyphus ihm bedrohlich entgegen. Dem Gott dürstete und hungerte es geradezu nach Entwurfsfallen, Programmierfehlern und undurchdringlichen Bedienkonzepten. Aus Ekel hatte er sogar vorsorglich sämtliche Lektüre über hochqualitativen Entwurf entsorgt. In seiner Not erzählte Pi dem Gott von Antipatterns und anderen Entwicklungskatastrophen, was eine geraume Zeit eine gute Wirkung tat. Während der sternklaren Nächte begann der zunehmend erschöpfte Architekt bald zu fantasieren – er erblickte LED-Bildschirme mit kunterbunten Klicki-Bunti-Oberflächen sowie Anwendungen, die lebenslustig über dem Betriebssystemkern tanzten.

Eines Morgens, als Pi schon jede Hoffnung aufgegeben hatte, strandeten sie in einem Großlabor, das von Hunderten sympathischer und äußerst agiler Entwickler bewohnt war. Zunächst mutete die Szenerie geradezu paradiesisch an, doch schon am ersten Tag erkannte Pi, dass die zahlreichen Entwickler regelmäßig bei Anbruch der Dunkelheit wie Zombies zu ihren Workstations huschten, um im kühlen Neonlicht satanische .NET-Programme zu kreieren. Kurzerhand entschloss er sich deshalb zur Flucht aus diesem vermeintlichen Paradies. Leider fasste auch die nervtötende Gottheit den Beschluss, Pi bei seiner Flucht zu begleiten, weshalb beide auch im weiteren Projektverlauf aneinander gekettet blieben. Nach vielen Personentagen und kurz vor dem endgültigen Burn-out landete das Projekt dank Pis Usability-Expertise an einem wichtigen Meilenstein. Enttäuscht ob seines Misserfolges, entschwand der egozentrische UIsyphus auf Nimmerwie-

dersehen in den Entwicklungsdschungel, ohne sich zu verabschieden oder wenigstens einen Blick zurück zu werfen.

Kurz nach der Rettung – so setzte Pi seine Schilderung fort – gesellten sich zwei McKinsey-Berater zu ihm, mit dem Auftrag, das Projekt ausgiebig zu untersuchen. Doch immer, wenn er ihnen seine Geschichte erzählte, vermittelte ihre Körpersprache Skepsis und großen Unglauben. Um endlich seinen Frieden zu finden, tischte Pi ihnen daher eine alternative Version des Geschehens auf. Anfangs, so begann Pi, habe der besagte Projektleiter sämtliche sinnvollen Verbesserungsmaßnahmen für die Bedienbarkeit unterlaufen und allzu widerspenstige und kritische Mitarbeiter einfach ins kalte Wasser geworfen. Als er aber dann noch den Versuch unternahm, von der Java-Plattform auf .NET zu wechseln, habe Pi schlussendlich die Geduld verloren und seinerseits dafür gesorgt, dass die Geschäftsleitung den Projektleiter unehrenhaft entließ, der daraufhin sofort untertauchte. Als frisch gekrönter Chefarchitekt konnte Pi mit seinem Team eine Software mit herausragender Bedienerfreundlichkeit entwickeln, die sich auf allen Betriebssystemplattformen nutzen ließ. Damit gaben sich die Berater endlich zufrieden.

Als er seine Schilderung beendet hatte, schaute mir Pi tief in die Augen, und wollte wissen, welche der beiden Versionen denn meiner Meinung nach die richtige sei? Ich war unschlüssig, befürchtete aber, die zweite Version entspräche der Wahrheit. Da Pi noch mit Kollegen zu einem Meeting verabredet war, verabschiedete ich mich von ihm, nicht ohne dass er mir zuvor die McKinsey-Projektanalyse in die Hand gedrückt hatte. Die Analyse habe ich später bei mehreren Tassen Java neugierig verschlungen, speziell den Satz: „Vom Projektleiter und Softwarearchitekten Pi, der trotz starken Wellengangs standhaft dem Gotte der höllischen Bedienqualen zu trotzen vermochte“.

Wir wissen freilich nicht, welche Version der Wahrheit entspricht, glauben aber fest daran, dass Sie mit den Beiträgen in dieser Ausgabe viel Freude haben werden. In diesem Sinne wünsche ich Ihnen viel Vergnügen bei der Lektüre.

Ihr Prof. Dr. Michael Stal