



Fehler schlucken

Als ich den Raum betrat, war niemand zu sehen. Erst leise Atemzüge verrieten die Frau unter dem Tisch, der in der Mitte des Zimmers stand.

Ich setzte mich mit dem Rücken an die Wand auf den Boden und hatte nun freien Blick auf Hanna R., Junior-Entwicklerin bei einem Consulting-Unternehmen.

Mitgenommen sah sie aus, getrieben, verfolgt, mit wirren Haaren. Immer wieder zuckte sie zusammen. Sie sei im Customer-Service für die Behebung von Fehlern zuständig, erzählte sie mit krächzender Stimme. Das mache ihr tatsächlich mehr Spaß, als neue Funktionen zu implementieren. „Das hat etwas von Detektivarbeit, Sherlock Holmes, verstehen Sie?“, flüsterte sie verschwörerisch. Schon immer sei das ihre liebste Lektüre gewesen: Detektivgeschichten.

Sie war exzellent in ihrem Job. Keiner fand Fehler so schnell wie sie. Vor allem aber gab sie sich nicht damit zufrieden, nur schnell die Symptome beispielsweise mit einem try-catch-Block zu eliminieren.

„Swallowing an error“, einen Fehler zu schlucken, kam für sie nicht infrage.

Alles ging gut, bis zu diesem einen Fehler. Die Einträge einer Dropdown-Box waren wild durcheinandergewürfelt. „Kleine Geschichte“, dachte sie und machte sich sofort an die Arbeit. Doch die Software wehrte sich und sie konnte den Bug nicht finden. Um 22 Uhr gab sie auf, programmierte die Einträge in richtiger Reihenfolge fest in den Code. Morgen würde sie ...

Doch am Morgen brach die Hölle über sie herein. Denn plötzlich war die Datenbank inkonsistent. Der Kunde brüllte, ihr Chef schrie, sie ärgerte sich und machte sich an die Arbeit, die Datenbank wieder in Ordnung zu bringen. Das gelang aber nicht, denn sie hatte bei alldem vergessen, dass die Fehler-Nichtbehebung immer noch im Code stand. So fixte sie Inkonsistenzen, während an anderer Stelle neue entstanden. Nach zwei Tagen Durcharbeiten ohne Schlaf wachte sie erst wieder in dieser Einrichtung auf. „Der Fehler ... er verfolgt mich ... großes Spaghettimonster hilf ...“, faselte sie, als sie zu sich kam.

Ich weiß nicht warum mir in diesem Moment der Witz von dem stotternden Knecht einfiel, der die Hühner abends noch schnell in den Stall treiben will. Eine Henne widersetzt sich erfolgreich seinen Anstalten, sie ins Innere zu bugsieren. Schließlich schimpft der Knecht: „W-wenn du jetzt nicht r-reingehst, d-dann haue ich d-dich kaputt-putt-putt-putt-putt“ – woraufhin alle wieder herauskommen. Merke: Schnell dauert in der Softwareentwicklung doppelt so lange wie langsam.

Viel Spaß mit der dotnetpro

Tilman Börner
Chefredakteur dotnetpro



Tobias Meier

stellt Ihnen Dapr vor, mit dem sich verteilte Anwendungen entwickeln lassen (S. 26)



Torsten Zimmermann

zeigt Ihnen, wie Sie Daten für Tests auf einfache Weise erzeugen können (S. 48)



Carl Klein

weiß, wie Sie BrowserStack einsetzen, um Apps auf mehreren Browsern zu testen (S. 106)