



Trashy End

Er hatte mit sich gerungen. Hatte getüftelt und gekniffelt. Hatte Unmengen an Code geschrieben, umstrukturiert und verworfen.

Lange Kommentarabschnitte zeugten von Phasen des Ausprobierens. Der Code löste zwar das Problem nicht, wurde aber doch als bewahrenswert bewertet. Vielleicht könnte er als Fallback oder Inspiration für später dienen, würde der neu eingeschlagene Weg nicht zum Erfolg führen. Schließlich war nicht nur Zeit in das Tippen geflossen, sondern auch die Überlegungen und die Fehlersuche hatten lange gedauert.

Die Zahl der Commits ging inzwischen in die Hunderte, und doch gab es noch immer viele Dateien, Elemente und Controls, die eigentlich schon lange gelöscht werden hätten müssen. Schließlich wären sie ja im Repository gespeichert.

Doch Vorsicht bewahrt die Porzellankiste beim Transport durch das Elefantengehege: Lieber den Code behalten und mitschleppen, als einmal einen Kniff zu verlieren, für den er viele Stunden gebraucht hatte, ihn herauszufinden. Außerdem hatte er Angst:

In welchem Commit wäre der gesuchte Abschnitt enthalten? Wie würde er ihn je wiederfinden?

Auf dem ihm unbekanntem Domänenterrain kam er nicht sehr schnell voran, wie er zugeben musste. Besser gesagt: katastrophal langsam. Dafür baute sich aber ganz allmählich die Anwendung auf.

Wobei die Rückschritte nicht ohne waren. Gerade eben hatte er ein Control wieder in die Wüste geschickt, das er über Stunden gepöppelt hatte. Aber es passte nicht in die Anwendung, es sperrte sich, war von keiner passenden Natur, wie er dazu sagte.

Geflüssentlich ignorierte er, dass die vielen auskommentierten Code-Stellen den Blick auf den wesentlichen, den aktiven Code verstellten. So passierte es, dass er Code dann doch zweimal entwickelt hatte, weil ihm erst nach dem Schreiben der zweiten Version auffiel, dass der Code schon einmal bei früheren Versuchen entstanden war. Er stand irgendwo im Urwald der Kommentare.

Aber irgendwann würde er fertig sein mit der Anwendung. Irgendwann würde er die letzten unnötigen Zeilen löschen können und das Programm ganz stolz an den Auftraggeber übermitteln. Der würde zufrieden sein und ihn loben und nach kurzer Zeit Erweiterungswünsche an ihn herantragen.

Er würde zufrieden sein. Zumindest kurze Zeit. Denn nur wenige Stunden nach der Lieferung der Software würde er erkennen, dass er das ganze Problem wesentlich einfacher mit wesentlich weniger Zeilen Code hätte lösen können.

Viel Spaß mit der dotnetpro

Tilman Börner
Chefredakteur dotnetpro



Dr. Holger Schwichtenberg

hat einen Narren an Vue.js gefressen und stellt die Version 3 vor (S. 8)



Marc Armbruster

weiß, wie Sie mithilfe eines Indexers auf Datenmengen in einem Objekt zugreifen (S. 42)



Andreas Aschauer

baut mit dem AI Builder Machine-Learning-Algorithmen in Ihre App ein (S. 70)