



Legacy-Horror

Seitdem Stefan Z. an den von Franz K. [1] empfohlenen Treffen der anonymen Entwickler teilnimmt, geht es ihm wesentlich besser.

Ruhig kann er die eineinhalb Stunden auf seinem Stuhl sitzen, den anderen zuhören oder selbst von seinem Schicksal erzählen. „Corona war der Auslöser“, begann er nach einem Monat regelmäßiger Besuche der Treffen. „Nach leichtem Fieber und etwas Husten ging es mir wieder gut.“ Er war ausgeruht, musste aber noch ein paar Tage isoliert zu Hause bleiben. Zum ersten Mal seit Langem hatte er damit Zeit. Ohne Deadline begann er zu programmieren. Er schrieb kurze Erweiterungsfunktionen und fand perfekte Namen für jede einzelne. Eine lange nicht mehr erlebte Euphorie stieg in ihm auf.

Ermutigt durch den Erfolg öffnete er den Code einer schon zehn Jahre alten Anwendung. Er baute seine neuen Funktionen ein. Das klappte überraschend gut. Im Quellcode der alten App stieß er dann auf eine 20-Zeilen-Funktion. Längst hatte er vergessen, wie sie funktionierte, fand aber schnell heraus, was sie machen sollte. Angesporn von seinen jüngsten Erfolgen fing er an, die Funktion zu renovieren. Nach einer knappen halben Stunde stand die neue Version. Die ersten Tests waren ermutigend: Beide Funktionen lieferten dieselben Ergebnisse. Aber je mehr Testläufe absolviert wurden, desto häufiger gab es Abweichungen.

„Seltsam“, murmelte er und machte sich an die Fehlersuche im neuen Code. Aber er fand keinen Bug.

Irgendwann wurde ihm klar, dass der Fehler keineswegs im neuen Code steckte – sondern im alten. Die Funktion hatte zwar offenbar lange Zeit zufriedenstellend funktioniert, aber war bei den zurückgegebenen Werten nicht in jedem Fall verlässlich gewesen. Deprimierend. Während er so über die Irrungen alten Codes sinnierte, verkürzte er die neue Funktion auf nur zwei Zeilen Code. Das hob kurzzeitig seine Stimmung wieder.

Aber dann: „Was mag nur für überflüssiger Code in der Anwendung stecken“, murmelte er zu sich selbst. Welche Fehler steckten darin, die nie jemand bemerkte, bis eines Tages die falsche Menge Kaffee bestellt würde, weil eine Funktion falsche Werte zurückgab. Konnte er mit diesem Wissen überhaupt noch zur Arbeit gehen? Vielleicht sollte er den Arbeitgeber wechseln? Aber was, wenn der neue Arbeitgeber neben den neuen auch ältere, umfangreichere Proben sehen wollte? Wenn aus dem alten Code nach kurzem Schütteln die Fehler nur so herauspurzelten? Verstrickt in gedankliche Endlosschleifen wurde Stefan Z. immer nervöser, unsteter und fahriger. Glücklicherweise gab Franz K. ihm den Tipp mit der Selbsthilfegruppe.

Inspiziert durch eigene Programmierarbeit und -erfahrung und durch die Geschichte von Franz K.

Bernhard Lauer
Freier Mitarbeiter und Gast-Editorial-Schreiber dotnetpro

[1] Tilman Börner, *Viele Welten*, dotnetpro 11/2022, Seite 3,
www.dotnetpro.de/A2211Edi



Dennis Doomen

bringt mithilfe von Pulumi und C# Anwendungen in die Amazon Cloud (S. 8)



Hendrik Lösch

zeigt zusammen mit Attila Bertok, wie Sie langlebige Softwarearchitekturen schaffen (S. 28)



Robert Mühsig

erleichtert Ihnen den Umgang mit CSS durch das Framework Tailwind CSS (S. 46)