Quick and dirty

Er war neidisch auf die Menschen, die hinter großen Namen standen. Menschen, die die Softwareentwicklung geprägt hatten.

Die ihre Marke hinterlassen hatten. Ob Robert C. Martin, Eric Evans, Ken Schwaber oder Martin Fowler: Sie alle waren bekannt. Ihre Namen standen in Wikipedia-Einträgen und wurden vielfach bei Vorträgen genannt. Wollte man als Organisator einer Konferenz einen davon als Sprecher engagieren, musste man tief in die Tasche greifen. Auch er wollte endlich etwas Bleibendes in der Softwareentwicklung hinterlassen. Etwas, das man zitieren würde.

Die Frage war nur, mit was. Mitnichten hatte er ein Open-Source-Projekt wie das von Linus Torvalds aufgesetzt. Auch Node.js war nicht ihm, sondern Ryan Dahl eingefallen. Tief in Gedanken tippte er an einem Code.

„Und da – zwischen einem foreach und der öffnenden geschweiften Klammer, hatte er die zündende Idee."

Er würde eine neue Vorgehensweise entwickeln. Statt Test-Driven Development mit Red-Green-Refactor würde er Sputter, Improve, Check, Keep erfinden, was so viel bedeutete wie hinrotzen, verbessern, prüfen, behalten.

Diese Idee war durch seine Arbeitsweise so naheliegend. Denn wenn er Software entwickelte, musste diese sofort arbeiten. Er konnte nicht Monate auf die fertige Version warten, sondern sie musste so schnell wie möglich funktionieren. Deshalb rotzte er den Code erst einmal hin (Sputter). Wenn das herauskäme, würde er in einem Schauprozess zu sechzehn Jahren VBA-Programmierung verurteilt. Aber das war eine andere Geschichte. Was da in den Computer floss, hatte so überhaupt nichts mit Clean Code zu tun. Es gab keine Tests, der Code war miserabel aufgeteilt, die Klassen viel zu groß, endlos lange Methoden verwirrten den Leser.

Aber das Hinrotzen war nur der erste Schritt. Da er nicht durchgängig an der Software arbeitete, sah er den Code durch die Neueinstiege jedes Mal mit anderen Augen. Und das führte zu Reue und Verbesserung (Improve).

Er extrahierte Funktionen, änderte Variablennamen und schrieb Tests (Check). Das Resultat war zwar nicht Clean, doch er erhielt schnell Funktionalität bei schließlich gutem Code, mit dem er leben konnte (Keep) – Weltklasse. Mit Sputter, Improve, Check, Keep würde er Softwaregeschichte schreiben.

Doch das Akronym aus den Anfangsbuchstaben (SICK) sprach eher gegen die Methode. Hätte er außerdem Recherche betrieben, hätte er bemerkt, dass diese Idee schon andere hatten [1]. Es würde wohl nichts werden mit der Berühmtheit.

Viel Spaß mit der dotnetpro wünscht Ihnen

Tilman Börner
Chefredakteur dotnetpro

[1] Programming in the twenty-first century, www.dotnetpro.de/SL2401Edi1

Christian Havel

greift auf eine Datenbank über das Muster Active Record zu, das Vor- und Nachteile hat (S. 26)

Fabian Hügle

ersetzt die WPF mit dem Cross-Plattform-Framework Avalonia (S. 34)

Daniel Basler

beweist, dass Machine Learning sogar mit Microsoft Excel machbar ist (S. 126)