



Jeden Tag besser

Okay, ich bekenne:
Ich bin Heimwerker. Und am liebsten
arbeite ich mit Holz.

Wenn man in Metall eine Bohrung einen halben Millimeter zu weit links platziert, passen die zu verschraubenden Teile nicht mehr ineinander. Holz zwingt man in so einem Fall schon noch zusammen.

Wahrscheinlich war ich deshalb auch schon immer der Softwerker. Fehler lassen sich einfacher beheben. Eine kleine Änderung im Quellcode, und schon ragt das Label nicht mehr in die Textbox.

Hardware sollte funktionieren. Punkt. Unter dem Schreibtisch herumkrabbeln, um Netzstecker und LAN-Kabel zu platzieren? Furchtbar.

Am allerliebsten baue ich Werkzeuge selbst – für die Holzbearbeitung und für die Arbeit. Eine Gehrungsführung für die Tischkreissäge, ein Lineal für die Oberfräse, eine Verwaltung für die Artikel, eine Möglichkeit, Texte aus Word in ein spezielles XML-Format zu exportieren, um sie dann in das Publishing Tool zu importieren. Check.

Lieber zwei Wochen Tool programmieren als fünf Mal dieselben stupiden Tätigkeiten ausführen.

Auch die Vorgehensweisen gleichen sich. Projekte beginnen immer mit einem Spike, einem Proof of Concept der zentralen Technologie. Dann wird geplant – und der Spike einfach meist weiterentwickelt. Großes Pfui. Selbstbelügend beschwichtige ich mich dann, dass nach dem Spike-Projekt eine neue Entwicklung einsetzen wird: sauber durchgeplant und mit 100 Prozent Testabdeckung. Nur – dazu kommt es nie. Denn das ehemalige Spike-Projekt tut, was es soll, und die Zeit ist eh immer viel zu knapp.

Und dann fühle ich mich schlecht, weil ich das, was ich jeden Tag in die dotnetpro bringe, nur zum Teil lebe. Agil? Ja, so arbeite ich immer. Kundenorientiert? Natürlich – ich bin mein bester Kunde. Kurze Feedback-Schleifen? Selbstverständlich. Aber Tests? Selten. Architektur? Nur in meinem Kopf. Und dann stelle ich mir die Frage, ob ich der Einzige bin, der so arbeitet, und ehrlich gesagt kann ich mir das nicht vorstellen. Wie viele Programme es wohl auf der Welt gibt, die schlicht funktionieren – nur unter die Haube darf keiner gucken?

Aber ich versuche, jeden Tag besser zu werden. Der Weg ist das Ziel. Ein zusätzlicher Test hier, eine Skizze, die das grobe Schema der Architektur zeigt, dort.

Worin ich schon richtig gut bin, ist die Pfadfinderregel: Verlasse den Platz sauberer, als du ihn vorgefunden hast. Refaktorisieren gehört bei jedem Entwicklungsschritt dazu. Ein Hoffnungsschimmer.

Tilman Börner
Chefredakteur dotnetpro



Tobias Richling

setzt automatische Builds
inklusive Unit-Tests mit TFS
und VSTS auf (S. 30)



Syrian Hadad

zeigt, wie sich .NET-Core-
Anwendungen einfach testen
lassen (S. 100)



Marek Böttcher

verbindet die Assistentin
Alexa mit Azure und ASP.NET
Core (S. 106)