



# Für die Ewigkeit

Wie lange lebt Software? Wie oft wird sie beerdigt, weil es inzwischen bessere, schnellere und effizientere Technologien gibt?

Wenn man sich die rund 20 Jahre ansieht, die .NET mittlerweile auf dem Buckel hat, dann kann man nur konstatieren: Das Leben einer Software ist kurz. Denn in den 20 Jahren kamen nicht nur viele Updates für das .NET Framework heraus, sondern mit .NET Core beziehungsweise .NET 5 auch komplett neue Plattformen. Die Mehrzahl der Projekte, die mit .NET Framework 1.0 entwickelt wurden, wurden wohl inzwischen auf eine spätere Version migriert, komplett neu geschrieben oder schlicht weggeworfen.

Trotzdem, es gibt sie noch, die alten Systeme, die unverwüstlich ihren Dienst tun. Und ich spreche jetzt nicht von COBOL-Programmen bei Banken. Auch im .NET-Umfeld sind immer noch Windows-Forms-Anwendungen in Betrieb und müssen gepflegt werden.

**„Verglichen aber mit der Software, die seit 1977 durchs All fliegt, ist die Lebensspanne von .NET-Anwendungen ein Witz.“**

Die Sonden Voyager 1 und Voyager 2 wurden in dem Jahr im Abstand von rund zwei Wochen gestartet und mithilfe der Gravitation diverser Planeten beschleunigt aus dem Sonnensystem katapultiert. Vier Jahre sollten Hard- und Software funktionieren. Aber alles war viel robuster und langlebiger als gedacht. Und so hat die Bodenstation nach 45 Jahren immer noch Kontakt zu den beiden Sonden, die fleißig Daten sammeln und übermitteln. Rund 20 Stunden braucht das Signal inzwischen von der Sonde zur Erde oder in der anderen Richtung. Die Sonden senden mit einer Leistung von nur 22 Watt. Wenn das Signal auf der Erde ankommt, ist es so schwach, dass die NASA eine 70-Meter-Antenne verwenden muss, um es überhaupt empfangen zu können.

Stellen Sie sich nun vor, Sie müssten für eine 45 Jahre alte Software einen Patch schreiben, diesen nach allen Regeln der Kunst testen und ihn dann in einem Gerät einspielen, auf das Sie keinen physischen Zugriff haben. Zwar haben Sie hier auf der Erde die neueste Technik zur Verfügung und können den Computer der Sonde auf dem Notebook simulieren. Doch sollte etwas schiefgehen, ist die Sonde unter Umständen verloren.

Wenn Sie also mal wieder unzufrieden mit Visual Studio oder .NET sind, dann denken Sie an das gerade beschriebene Szenario.

Viel Spaß mit der dotnetpro wünscht Ihnen

Tilman Börner  
Chefredakteur dotnetpro



**Dennis Doomen**

hat 30 Best Practices gesammelt, mit denen Sie besser mit Unit-Tests umgehen (S. 62)



**Alexander Genne**

bringt native Programmierung und Webtechnologien gewinnbringend zusammen (S. 74)



**Tobias Richling**

zeigt Ihnen, wie Sie eine Anwendung in einen Kubernetes-Cluster verteilen (S. 86)