

INTERVIEW

„Die Zukunft von .NET 5 ist großartig“

Entwickler trainieren oder lieber coden: Thomas Claudius Huber macht beides gern.

Wie passen Skateboard fahren, .NET 5 und TypeScript zusammen? Thomas Claudius Huber ist nicht nur vom Brett mit den vier Rollen, sondern auch von .NET 5, Angular und TypeScript begeistert. Im Interview mit der dotnetpro weiß er aber auf eine Frage keine eindeutige Antwort.

Du entwickelst selbst Software, trainierst aber auch andere Entwickler. Was machst du lieber?

Thomas Claudius Huber: Bei der Frage kann ich mich nicht entscheiden. Das Entwickeln von Software macht mir viel Spaß. Man steht vor einem Problem, analysiert es, überlegt sich, wie man es lösen kann, und arbeitet sich dann schrittweise an die Lösung heran. Ein bisschen ist das Entwickeln für mich wie das Lösen eines Kreuzworträtsels. Das Lehren baut für mich darauf auf. Ich mag es, anderen Entwicklern Konzepte und Methoden beizubringen, meine Faszination zu teilen und meine eigenen beim Entwickeln gesammelten Erfahrungen weiterzugeben. Das Schönste am Lehren ist für mich zu sehen, wie der Funke überspringt und wie andere Entwickler erfolgreich ihre Projekte mit dem vermittelten Wissen umsetzen.

Wie viel Zeit verwendest du darauf, selbst Neues zu lernen?

Huber: Das ist unterschiedlich. Ich habe keinen fixen Zeitblock fürs Lernen reserviert. Aber ich schaue, dass ich ständig den Überblick über die neuen Themen habe. Ich versuche auch, regelmäßig etwas Unbekanntes auszuprobieren. Beispielsweise stehen Q# und Quantum Computing auf meiner Liste. Ehrlich gesagt aber faszinieren mich zu viele Dinge, die von neun bis 17 Uhr Platz hätten. Da ist es eine Herausforderung, eine gute Work-Life-Balance zu finden.

Wie lernst du am liebsten? Lesen oder ausprobieren?

Huber: Ich lese sehr gerne. Einen Artikel, ein Buch, einen Blogpost, eine Dokumentation. Ich stöbere gerne darin, um anschließend kleinere Dinge auszuprobieren und zu experimentieren. Lesen ist für mich die Theorie, Programmieren die Praxis. Für den Lernerfolg braucht es meist beides.

Aber auch das Erstellen von Trainings, Vorträgen, Videos, Büchern und so weiter ist Teil meines eigenen Lernprozesses. Albert Einstein sagte bereits, „wenn du es nicht einfach erklären kannst, hast du es nicht gut genug verstanden“. Zu überlegen, wie ich Sachverhalte gut vermitteln kann, hilft mir selbst auch, Dinge besser und tiefgründiger zu verstehen. Sobald man dem Kollegen oder der Kollegin ein Problem schil-

dert, kommt man selbst auf die Lösung. Genau das passiert auch beim Erstellen von Blogposts, Artikeln, Vorträgen etc., daher ist das zum Lernen sehr effektiv.

Du bist sowohl in der Schweiz als auch in Deutschland unterwegs. Siehst du zwischen den beiden Ländern einen Unterschied in der Akzeptanz von Technologien wie der Cloud?

Huber: Nein. Den Unterschied sehe ich eher zwischen einzelnen Firmen – unabhängig vom Standort. Manche Firmen sind technologisch sehr offen, neugierig, experimentierfreudig und überlegen ständig, wie sie ihre IT noch weiter verbessern können. Andere Firmen investieren nicht viel in das, was bereits läuft, und starten neue Projekte auch mit älteren Technologien, die bereits bewährt und stabil sind. Damit halten sie das Risiko gering, auf ein falsches Pferd zu setzen. Beides hat irgendwo seine Berechtigung. Bei der Cloud steigt die Akzeptanz stark an. Für viele Firmen ist es bereits klar, dass die Cloud gut funktioniert und in vielen Fällen besser als der eigene Server ist.

Neuerdings beschäftigst du dich auch mit TypeScript und Angular. Warum?

Huber: Mit AngularJS konnte ich mich nicht anfreunden. Aber als im Jahr 2016 Angular 2 rauskam, hat mich das Konzept rund um die komponentenbasierte Entwicklung fasziniert. Auch fand ich gut, wie Dependency Injection direkt in das Framework integriert wurde. Ich habe festgestellt, dass viele Dinge ähnlich zu WPF sind: Data Binding, Komponenten/UserControls, Direktiven/Markup Extensions, Pipes/ValueConverter und so weiter.

Auch eine Angular-Komponente als solche besteht aus einer View (HTML) und einem ViewModel (Komponenten-Klasse). Somit war/ist Angular für mich das WPF für Web-Anwendungen. Ich habe später in ersten Kundenprojekten mit Angular 2 gearbeitet und anschließend mit Thomas Gassmann und Thomas Bandixen ein zweitägiges Angular/TypeScript-Training für Trivadis geschrieben.

TypeScript ist für mich das C# für die JavaScript-Welt, denn ich kann statisch typisierten Code schreiben, der die modernsten JavaScript-Features nutzt. Am Ende kommt reines JavaScript heraus. Ich war und bin so begeistert von TypeScript, dass ich dazu im Februar 2017 ein Buch geschrieben habe. Mir fällt als Entwickler kein Grund ein, warum ich ein JavaScript-Projekt ohne TypeScript starten sollte.

Wie schätzt du ein, dass .NET Core jetzt WPF und WinForms unterstützt?

Huber: Das finde ich super, da .NET Core (und .NET 5) die Zukunft ist. Somit profitieren diese beiden wichtigen UI-Frameworks weiterhin von den neuesten Innovationen im Bereich .NET. Man darf beispielsweise nicht vergessen, dass Sprachfeatures von C# 8.0 unter .NET Framework 4.8 nicht unterstützt werden. Mit der Portierung von WPF und WinForms auf .NET Core ist somit auch sichergestellt, dass sich für diese Anwendungen die modernsten C#-Features nutzen lassen.

Darüber hinaus gibt es sehr viele produktive WPF- und WinForms-Anwendungen. Viele meiner Kunden starten neue Projekte mit WPF oder WinForms. Das ist durchaus in Ordnung, auch wenn wir momentan einen Hype rund um Cross-Platform und Web haben. Man darf nicht vergessen, dass es zahlreiche Unternehmen gibt, in denen die Mitarbeiter an einem Schreibtisch mit Windows-Rechner und zwei Bildschirmen sitzen. Da spielen Cross-Platform und Web keine Rolle. Und genau da haben WPF und WinForms ihre Daseinsberechtigung. Eine Web-Anwendung bietet dann lediglich beim Deployment einen kleinen Vorteil.

Glaubst du, dass Microsoft auch für Linux und MacOS Oberflächentechnologien für .NET Core bereitstellen wird?

Huber: Wenn man sich anschaut, was man mit .NET Core 3.1 alles machen kann, dann ist eine Oberflächentechnologie zum Entwickeln von Cross-Platform-Desktop-Apps so ziemlich das Einzige, was mir fehlt. Derzeit ist Microsoft dran, die UWP-UI-Technologie von Windows 10 zu entkoppeln und als eigenständiges UI-Framework unter dem Namen WinUI bereitzustellen. WinUI soll noch dieses Jahr erscheinen. Dabei gibt es bereits eine Open-Source-Initiative namens UNO Platform, welche WinUI-Anwendungen mit Xamarin und WebAssembly für Android, iOS und das Web lauffähig macht. Doch ich denke, dass da noch mehr kommen wird und spannende Zeiten auf .NET-Entwickler warten.

Du hast für die Developer Week 20 den Track zu .NET zusammengestellt. War das eine besondere Herausforderung?

Huber: Ja. Es gab sehr viele Einreichungen und alle haben mir sehr gut gefallen. Das Ganze auf sechs Sessions zu reduzieren, ist keine einfache Aufgabe. Ich habe mir die Fragen gestellt: Was müssten .NET-Entwickler aus einem .NET-Track im Jahr 2020 mitnehmen? Was würde ich selbst von einem .NET-Track erwarten?

Ich habe mir über diese Fragen den Kopf zerbrochen, die eingereichten Sessions sortiert, gefiltert, evaluiert, priorisiert, wieder alles verworfen und nochmals sortiert, gefiltert, evaluiert und priorisiert. Am Ende bin ich nach viel Arbeit zum Programm für den .NET-Track gekommen. Ich glaube, dass das Programm das beinhaltet, was im Jahr 2020 für .NET-Entwickler wichtig und spannend ist. Ich freue mich schon sehr auf diesen Tag. Allen .NET-Entwicklern/Entwicklerinnen, die Sessions eingereicht haben, kann ich nur Danke sagen. Und für jene, die leider nicht angenommen wurden: Es waren alles tolle Vorschläge. Reicht auch beim nächsten Mal eure Sessionideen ein!

● Thomas Claudius Huber

Thomas ist Microsoft MVP im Bereich Windows Development und Senior Principal Consultant und Partner der Trivadis AG. Als Trainer, Berater und Entwickler ist er in den Bereichen C#, TypeScript, WPF, UWP, Angular und Azure unterwegs. Da ihn Benutzeroberflächen schon seit seinem Informatikstudium faszinieren, setzt er sich seit der ersten WPF-Version mit der UI-Programmierung mit XAML auseinander. Heute setzt er neben XAML-UIs begeistert Anwendungen mit TypeScript und Angular um. Thomas ist Autor verschiedenster Bücher, darunter das umfassende Handbuch zur Windows Presentation Foundation und das Buch „Getting Started with TypeScript“. Seine persönliche Webseite finden Sie unter www.thomasclaudiushuber.com.



Bild: Thomas Claudius Huber

Zum Schluss: Was fällt dir zu folgenden Begriffen ein?

Skateboard: Ich habe als Fußballspieler mit 20 Jahren mit dem Skateboarden aufgehört und mit 38 Jahren wieder angefangen. In der Theorie bin ich Profi (lacht). In der Praxis bin ich davon weit entfernt. Skateboarden heißt für mich auf jeden Fall Spaß, Freiheit und Abschalten.

Blazor versus Angular: Ich halte Blazor und Angular für sehr gute Frameworks und bin Fan von beiden. Aber als .NET-Entwickler zieht es mich mehr zu Blazor hin. Hier gefällt mir insbesondere, dass einem auch die JavaScript-Welt offensteht und man somit das Beste aus beiden Welten nutzen kann.

Node.js: Unglaublich und beeindruckend, wie weit es JavaScript geschafft hat. Allerdings setzen meine Kunden und ich im Backend auf .NET. Ich habe mit Express und Node.js experimentiert und REST APIs gebaut. Aber am Ende des Tages gefallen mir .NET und C# besser.

Agil: Agile Methoden sollte jeder Entwickler kennen und verstehen. Allerdings zählt auch hier der gesunde Menschenverstand. Wichtig ist, dass das Team einen agilen Prozess nutzt, der für das Team gut funktioniert. Ob das dann 100 Prozent Scrum-konform ist oder nicht, das ist nicht entscheidend.

Zukunft von .NET 5: Die Zukunft von .NET 5 ist großartig. Microsoft hat mit der Einführung von .NET Core vieles richtig gemacht: Cross-Platform, hochperformant, weitestgehend das .NET, das wir alle kennen und lieben. Und .NET-5-Vereinheitlichung ist der richtige Weg. Endlich gibt es von Xamarin über ASP.NET Core bis WPF eine einheitliche Base Class Library (BCL). Die Zukunft gehört der Long-Term-Support-Version .NET 6, die über drei Jahre unterstützt wird (<https://github.com/dotnet/core/blob/master/roadmap.md>). ■