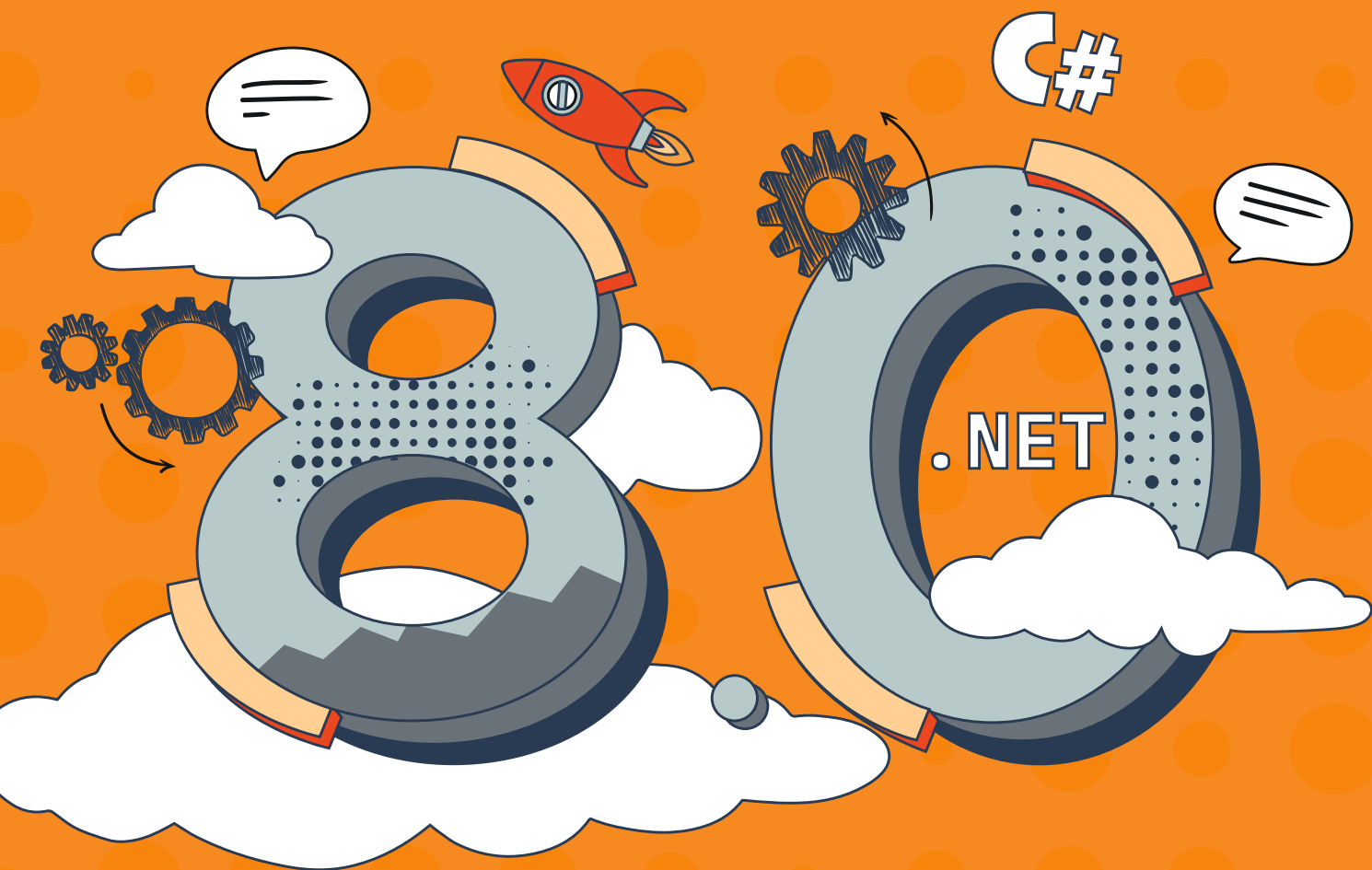


We Are Developers!

Frühjahr – 1/2024



Eine Themenbeilage der Heise Medien GmbH & Co. KG



> UPDATE AUF .NET 8.0

Tipps und Tricks im Experteninterview

> DER WEG IN DIE IT

Wie IT-Neulinge und Projekte zusammenfinden

> PROGRAMMIERUNG

Entwurfsmuster in der Softwareentwicklung



Powered by 


IN AGILEN WORKSTREAMS

DIE CLOUD-LÖSUNGEN

VON MORGEN ENTWICKELN.

DARUM SIND WIR BEI DATEV.

Gemeinsam sichere Cloud-Lösungen und innovative Apps realisieren: Als Cloud-Entwicklerin oder -Entwickler erwarten dich bei DATEV vielfältige Aufgaben in einer agilen Innovations-Kultur. Informiere dich über freie Stellen und spannende Projekte bei einem der führenden IT-Dienstleister in Europa.

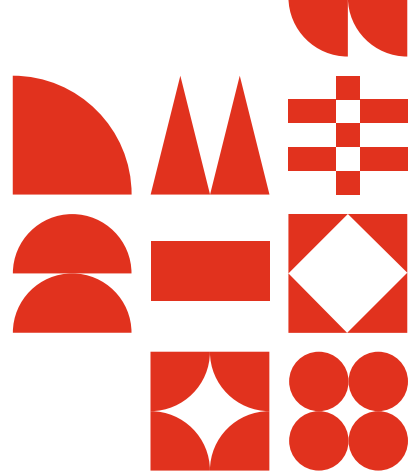


Valeria und Dominik,
Cloud-Entwicklerin und
-Entwickler bei DATEV

[DATEV.DE/KARRIERE](https://datev.de/karriere)



Zukunft gestalten.
Gemeinsam.



Von .NET über IT-Quereinstieg bis IKEA

Für die Frühjahrsausgabe der „We Are Developers!“-Beilage haben wir fünf .NET-Experten zum Interview gebeten. Wir haben Thomas Claudius Huber, André Krämer, Dr. Holger Schwichtenberg, Rainer Stropek und Christian Wenz gefragt, was die kürzlich erschienene Version mit Long-term-Support an neuen Funktionen bringt, wie diese sich optimal nutzen lassen und wie das Upgrade gelingt. Zu den Neuerungen in .NET 8 zählen Blazor United, ein zusätzliches Einsatzgebiet für den Ahead-of-Time-Compiler Native AOT sowie Sprachfeatures in C# 12.0, darunter Primary Constructors und Collection Expressions.

Auch in dieser Ausgabe soll es wieder um den Quereinstieg in die IT gehen. Nicht, weil uns die Themen ausgehen, sondern, weil es so ein wichtiges Thema ist! Der Fachkräftemangel ist in Deutschland allgegenwärtig – nicht nur in der IT. Doch konzentrieren wir uns auf die Softwareentwicklung: Allein im Jahr 2022 blieben insgesamt 137 000 offene IT-Stellen in Deutschland unbesetzt. Unser Autor Andreas Monschau, Senior IT-Consultant, berichtet aus zehn Jahren Berufserfahrung und zeigt, wie das Unternehmen, für das er tätig ist, IT-Neulinge mit Projekten und Kunden zusammenbringt. Der Artikel beschreibt das Trainee-Programm und erläutert die Auswahlkriterien, den Auswahlprozess sowie die Ziele des Programms.

Der dritte Beitrag der Frühjahrsbeilage widmet sich den Grundlagen der Programmierung. Es geht um Entwurfsmuster oder Design Patterns, die Autor Nils Kasseckert mit einem modularen IKEA-Regalsystem vergleicht. Der Artikel betont ihre Bedeutung für die Verbesserung der Code-Qualität, Testbarkeit und Wartbarkeit in größeren Softwareprojekten. Von .NET 8.0 über den Quereinstieg in die IT bis zu den Grundlagen der Softwareentwicklung – auch diese Ausgabe bietet wieder eine bunte Palette an Beiträgen.

Viel Freude beim Lesen!

M. Domogalla



INHALT

4 Interview:
Update auf .NET 8.0

10 IT-Quereinsteiger
treffen auf
Kundenprojekte

15 Entwurfsmuster
in der Softwareent-
wicklung

Young Professionals schreiben für Young Professionals

Unsere Beilage zu c't und iX basiert überwiegend auf einer Online-Artikelserie, die Young Professionals eine Bühne bietet für erste Fachartikel. Die Autoren und Autorinnen erhalten von der Heise-Redaktion Unterstützung beim Konzipieren und Schreiben.

Dein erster Fachartikel bei Heise

Die Serie ermutigt dazu, eigene Erfahrungen mitzuteilen, ein Projekt vorzustellen – oder wolltest du schon immer mal selbst einen Fachartikel schreiben und hast eine Idee?

Schreib uns: developer@heise.de

> Update auf .NET 8.0



Was ist am aktuellen Release .NET 8.0 bemerkenswert, wie gelingt die Migration und was könnte die Zukunft bringen? Das beantworten fünf .NET-Experten im Interview.

Microsoft hat .NET 8.0 mit drei Jahren Long-Term Support (LTS) im November 2023 veröffentlicht. Wir sprachen im Vorfeld des Release mit den .NET-Experten Thomas Claudius Huber, André Krämer, Dr. Holger Schwichtenberg, Rainer Stropek und Christian Wenz.

ix: *.NET 8.0 ist das erste LTS-Release seit .NET 6.0 im Jahr 2021. Was ist darin für euch die wichtigste Neuerung?*

Thomas Claudius Huber: Ich entwickle schon seit Jahren Full-stack-Anwendungen. Das heißt, ich verwende als Client üblicherweise eine Desktop- oder Web-App, eine Datenbank und unter Umständen eine API-Schicht zwischen Client und Datenbank. Bei manchen Datenbankzugriffen nutzte ich in der Vergangenheit Dapper, eine Library für objektrelationales Mapping (ORM), da sie reine SQL-Abfragen direkt auf Entities mappt, was in verschiedenen Szenarien sinnvoll war. Entity Framework Core 8 enthält mit Raw SQL Queries ein sehr ähnliches und flexibles Feature, mit dem ich denke, dass ich die meisten, wenn nicht sogar alle meiner Dapper-Datenbank-Zugriffe ersetzen kann. Daher freue ich mich besonders auf dieses Feature. Doch daneben gibt es viele weitere spannende Dinge, sowohl in den Core Libraries als auch in den verschiedenen Frameworks wie Blazor oder Windows Presentation Foundation (WPF). In WPF hat Microsoft beispielsweise mit .NET 8.0 eine neue `OpenFileDialog`-Klasse eingeführt, die die Auswahl eines Ordners erlaubt. Bisher musste man für einen

solchen Dialog die entsprechende Windows Forms Assembly referenzieren, was in Zukunft nicht mehr notwendig ist.

André Krämer: Da ich mit meinem Team schwerpunktmäßig Apps mit .NET Multi-platform App UI (.NET MAUI) entwickle, sind die vielen Qualitätsverbesserungen, die dort in der Version 8.0 kommen, für mich die wichtigste Neuerung. Ein Blick auf GitHub zeigt, dass das .NET-MAUI-Team mehrere Hundert Issues schließen und Pull Requests mergen konnte. Diese Verbesserungen betreffen hauptsächlich Fehlerbehebungen, Performancesteigerungen und ein besseres Speichermanagement. Bisher haben wir .NET MAUI überwiegend für Neuentwicklungen eingesetzt und mussten einige Zeit investieren, um Fehler in den Versionen 6.0 und 7.0 zu umgehen. Mit der Qualität in Version 8.0 werden wir nun auch die Migration bestehender Xamarin.Forms-Apps unserer Kunden vorantreiben, da dies nun sowohl technisch als auch wirtschaftlich sinnvoll ist.

Dr. Holger Schwichtenberg: Die meiner Meinung nach wichtigste Neuerung in .NET 8.0 ist definitiv Blazor United! Nun hat man bei .NET-basierten Browseranwendungen nicht mehr die Qual der Wahl zwischen Pest oder Cholera – Blazor Server und Blazor WebAssembly –, sondern kann das Beste aus beiden Welten in einer Anwendung vereinen und sogar einen nahtlosen und für Benutzerinnen und Benutzer unsichtbaren Wechsel zur Laufzeit realisieren.

Rainer Stropek: .NET 8.0 bringt zwei Dinge, die meine Arbeit kurz- und langfristig beeinflussen werden. Die erste Sache betrifft C#. In C# 12.0 kommen zwar keine bahnbrechenden Neuerungen, aber dafür solche, die „Everyday C#“ betreffen. Damit meine ich jene Sprachfeatures, die man nicht nur in Spezialsituationen, sondern jeden Tag verwenden kann. Die zwei wichtigsten Neuerungen in dieser Kategorie sind Primary Constructors und Collection Expressions. Nach einigen Wochen praktischer Arbeit mit .NET 8.0 merke ich schon jetzt, dass sie mir fehlen, wenn ich an älteren Codebasen arbeiten muss. Das Zweite ist eine eher langfristige Sache. Native AOT wurde im Gegensatz zu .NET 7.0 auf ASP.NET-Core-Web-

In a Nutshell

- Microsoft hat .NET 8.0 als aktuelle Version mit Long-Term Support (LTS) veröffentlicht.
- Zu den Neuerungen zählen Blazor United, ein zusätzliches Einsatzgebiet für Native AOT sowie Sprachfeatures in C# 12.0.
- Im Interview teilen fünf Experten ihre Meinungen zu den neuen Funktionen und geben Upgrade-Tipps.



Thomas Claudius Huber ist Microsoft MVP im Bereich Windows Development und Senior Principal Consultant bei der Trivadis AG. Als Trainer, Berater und Entwickler ist er in den Bereichen C#, XAML, TypeScript und Azure unterwegs.



André Krämer ist Gründer und CEO der Quality Bytes GmbH. Mit seinem Team entwickelt er Systeme zur Dokumentenautomatisierung mit TX Text Control und Mobile Apps mit .NET MAUI. Microsoft hat ihn elfmal in Folge mit dem MVP-Award ausgezeichnet.

APIs erweitert. Das ist eine kleine Revolution. Wenn ich bisher in der Cloud flotte, ressourcenschonende APIs brauchte und auf die Systemnähe von Rust verzichten konnte, griff ich zu Go. Die Tatsache, dass der Code direkt in Maschinensprache übersetzt wird, macht Cold-Start-Zeiten möglich, an die .NET bisher nur schwer herangekommen ist. Durch Native AOT ändert sich das. Diese Entwicklung zeichnete sich schon seit Langem ab, mit .NET 8.0 können wir als Entwicklerinnen und Entwickler aber endlich loslegen. Für mich weitet sich der Einsatzbereich von .NET dadurch speziell im Kontext von Serverless Cloud Computing weiter aus.

Christian Wenz: Die neue Version erfordert, dass auf .NET 6.0 oder .NET 7.0 basierende Anwendungen dorthin aktualisiert werden; der Support der LTS-Version .NET 6.0 läuft etwa ein Jahr nach Erscheinen von .NET 8.0 aus, der von .NET 7.0 bereits sechs Monate zuvor. Prinzipiell freue ich mich für den Themenbereich Webanwendungen über die vielen großen und kleinen Performance-Verbesserungen. Microsoft hat für die teils arg optimierten Einreichungen beim TechEmpower-Benchmark einiges an Kritik kassiert und kann jetzt zeigen, dass ASP.NET Core ohnehin sehr schnell ist. Wer an Details interessiert ist: Der ausführliche Blog-Post [alle Links zu diesem Interview unter ix.de/zjtm] beantwortet alle Fragen. Für ASP.

NET Core 8 gibt es auch noch einige Erweiterungen in den Bereichen Authentifizierung und Identity, die überfällig waren.

ix: Mit .NET 8.0 erscheint auch die neue Programmiersprachenversion C# 12.0. In welchen Anwendungsfällen dürften die neuen Sprachfeatures hauptsächlich zum Einsatz kommen?

Thomas: Ich denke, die neuen Sprachfeatures werden an verschiedensten Stellen älteren Code ersetzen, insbesondere die neuen Collection Expressions und Primary Constructors. Die Collection Expressions machen das Initialisieren eines Arrays deutlich einfacher.

Bisher konnten wir in C# ein Array mit drei Integer-Werten wie folgt initialisieren:

```
int[] myArray = new[] { 1, 2, 3 };
```

Mit den in C# 12.0 eingeführten Collection Expressions lässt sich dies kürzer und auch lesbarer schreiben. Dazu werden die Werte einfach in eckigen Klammern platziert:

```
int[] myArray = [ 1, 2, 3 ];
```

Primary Constructors, die bereits aus C# Records bekannt sind, gibt es jetzt für Klassen und Structs. Auch hier könnte es

Fraunhofer IIS

Wir tauchen für die reale Welt tief in virtuelle Welten ein.

Am Fraunhofer IIS forschen Sie an der Entwicklung innovativer Softwaresysteme, die unser echtes Leben bereichern.

Kommen Sie in das Team Fraunhofer IIS!
www.iis.fraunhofer.de/jobs

Veränderung startet mit uns!



Dr. Holger Schwichtenberg ist Autor von mehr als 90 Fachbüchern. Er arbeitet als CTE bei der MAXIMAGO-Softwareentwicklung und bietet mit seinen Kollegen Beratung und Schulungen unter www.IT-Visions.de an.



Rainer Stropek ist IT-Unternehmer, Softwareentwickler, Trainer, Autor und Vortragender im Microsoft-Umfeld. Er ist seit 2010 MVP für Microsoft Azure und entwickelt mit seinem Team die Software Time Cockpit.



Christian Wenz ist Berater, Trainer und Autor für Webtechnologien. Als Teilhaber der Arabiata Solutions GmbH sorgt er für schnellere und sicherere Webanwendungen. Er ist ASP.NET MVP und ASPInsider sowie Hauptautor der Zend PHP Certification.

verschiedene Stellen geben, die diese neue Funktionsweise kompakter und somit lesbarer macht. Das folgende Beispiel zeigt eine einfache Person-Klasse, die einen Konstruktor enthält, um darin die zwei Properties FirstName und LastName zu initialisieren:

```
public class Person
{
    public Person(string firstName, string lastName)
    {
        FirstName = firstName;
        LastName = lastName;
    }
    public string FirstName { get; set; }
    public string LastName { get; set; }
}
```

Mit C# 12.0 lässt sich mit einem Primary Constructor die gleiche Klasse wie folgt schreiben:

```
public class Person(string firstName, string lastName)
{
    public string FirstName { get; set; } = firstName;
    public string LastName { get; set; } = lastName;
}
```

Das ist deutlich kompakter. Sobald sich Entwicklerinnen und Entwickler an die Primary Constructors gewöhnt haben, wird der Code lesbarer.

André: Die Einführung von Primary Constructors in C# 12.0 ist besonders spannend. Sie ermöglichen den Zugriff auf ihre Konstruktorparameter innerhalb der gesamten Klassendefinition, was den Code kürzer, übersichtlicher und somit wartbarer macht. Das setzt den Trend der letzten C#-Versionen zu einer kompakteren C#-Sprachsyntax fort.

Holger: Die Primärkonstruktoren für Klassen, die es schon einmal 2014 als Prototyp gab, werden nun endlich in die Sprachsyntax von C# 12.0 eingebaut. Bisher gab es das nur für Record-Typen. Primärkonstruktoren werden meine Fingerkuppen schonen. Leider gibt es auch einen Wermutstropfen: Anders als bei den in C# 9.0 eingeführten Record-Typen erstellt der Primärkonstruktor nämlich keine öffentlichen Properties in der Klasse, sondern nur private Fields :(

Auch die Aliase für beliebige Typen sind eine tolle Verschlanung des zu erfassenden Codes, insbesondere für Tupel:

```
global using Author = (int ID, string Name, string Website);
...
    Author hs = (42, "Dr. Holger Schwichtenberg",
                "www.IT-Visions.de");
```

Rainer: Ich habe Primary Constructors und Collection Expressions schon als meine zwei Highlights von C# 12.0 hervorgehoben. Beide sorgen dafür, dass sich unnötiger Boilerplate-Code vermeiden lässt. So viele Klassen, die ich erstelle, brauchen Konstruktorparameter. Den Konstruktor manuell erstellen zu müssen, war bisher lästig – speziell, wenn man aus Sprachen wie TypeScript weiß, dass es auch ohne ginge. Man hätte zwar schon bisher auf Record Classes zurückgreifen können. Dann muss man aber Immutability akzeptieren. Das passt für manche Szenarien, aber bei Weitem nicht für alle. Mutable Classes sind in vielen Situationen ein Muss und Primary Constructors daher sehr hilfreich. Collection Expressions machen den C#-Code einheitlicher. Es gab bisher unzählige verschiedene Möglichkeiten, Listen zu initialisieren. Jede funktionierte etwas anders. Dann kamen List Patterns im Bereich Pattern Matching dazu, die wieder eine andere Syntax hatten. Collection Expressions sind ein wichtiger Schritt zur Vereinfachung. Für mich als C#-„Opa“, der seit 20 Jahren C# programmiert, sind die verschiedenen Syntaxvarianten kein Problem. Ich habe mich daran gewöhnt. Anfängerinnen und Anfänger kämpfen aber damit, ich merke das in meiner Arbeit als Trainer und Teilzeitlehrer an einer Informatik-Schule. Alles, was C# leichter zu lernen macht, finde ich gut, denn es sorgt dafür, dass uns der Nachwuchs im Bereich C# nicht ausgeht. Das ist für mich als IT-Unternehmer wichtig.

Christian: Ich halte bei gereiften Sprachen wie C# neue Sprachfeatures in der Regel nicht für entscheidend; häufig

handelt es sich zudem um Syntaxzucker, aber nichts Weltbewegendes. Von den neuen Möglichkeiten in C# 12.0 finde ich eine recht interessant, die allerdings – zu Recht – als „experimentell“ markiert worden ist: Interceptors. Vereinfacht gesagt, kann ein Interceptor Aufrufe zu manchen Methoden zur Compile-Zeit auf sich selbst umleiten. Das ist insbesondere für Codegeneratoren interessant. Spezifiziert ist dieses Feature schon [Spezifikation unter: [ix.de/zjtm](https://github.com/dotnet/runtime/blob/main/src/compilers/Toolbox/IX/ix.de/zjtm)], doch die API kann sich jederzeit ändern, bis hin zur Entfernung.

ix: Welche Tipps könnt ihr unseren Leserinnen und Lesern mitgeben, damit das .NET-Upgrade möglichst reibungslos gelingt?

Thomas: Ich persönlich gehe wie folgt vor: Ich stelle die eigenen Projekte in der .csproj-Datei von .NET 6.0/7.0 auf .NET 8.0 um und probiere zunächst, ob alles kompiliert. Üblicherweise erfolgt dies recht reibungslos, da mit neuen .NET-Versionen sehr wenige Änderungen eingeführt werden, die den alten Code ungültig machen würden. Nach dem Kompilieren werfe ich einen Blick auf die eventuell neu auftretenden Warnungen und versuche, diese zu lösen. Schließlich lese ich die Breaking Changes, die auch eventuellen Einfluss auf das Laufzeitverhalten der Anwendung haben. Ist dies geschafft, dann ist das Projekt migriert, und man kann ab sofort von den neuen Features in .NET 8.0 profitieren.

André: Die Migration von .NET 6.0 auf .NET 8.0 bringt einige Breaking Changes mit sich, aber die Probleme sollten sich in Grenzen halten. Für den reibungslosen Ablauf von Version 6.0 auf 8.0 oder 7.0 auf 8.0 habe ich folgende Tipps:

Auch wenn die meisten Entwicklerinnen und Entwickler diesen Hinweis gerne ignorieren: Ich empfehle, vor der Migration die Hinweise zu Breaking Changes in der offiziellen Dokumentation zu lesen und zu prüfen, ob man betroffen ist. Nach der Migration sollte man umfassende Tests durchführen, um sicherzustellen, dass die eigene Anwendung wie erwartet funktioniert. Außerdem sollte man seinen Code auf die Verwendung von Klassen oder Methoden prüfen, die unter .NET 8.0 nun als obsolet gekennzeichnet sind. Neben den einfachen Updates von .NET 6.0 und 7.0 auf 8.0 dürften mit dem geplanten Support-Ende von Xamarin.Forms im Mai 2024 auch einige komplexere Upgrade-Vorhaben von Xamarin.Forms zu .NET MAUI 8.0 anstehen. Hier ist der erste Schritt, die Kompatibilität der eingesetzten NuGet-Pakete zu prüfen. Einige unter Xamarin.Forms beliebte Pakete wurden nicht nach .NET MAUI portiert. In solchen Fällen muss man entweder Alternativen finden, das Paket selbst portieren oder auf den Einsatz verzichten. Je nachdem, wie die eigene Lösung ausfällt, kann hier ein erheblicher Aufwand entstehen. Darüber hinaus empfehle ich vor der Migration zu prüfen, ob sämtlicher plattformspezifischer Code in Form von Custom Renderern und Effects auch unter .NET MAUI noch benötigt wird. Für einiges gibt es mittlerweile darin integrierte Funktionen.

Zu guter Letzt kann ich den Einsatz der aktuellen Version des .NET Upgrade Assistant ans Herz legen. Die Ergebnisse der aktuellen Version sind beeindruckend und um ein Vielfaches besser als die Ergebnisse früherer Versionen. Mittlerweile sind bei einfachen Apps kaum noch Anpassungen nach dem Einsatz des Upgrade Assistant notwendig. Was ich übrigens immer wieder beobachte, ist, dass Entwicklungsteams versuchen, die Migration von Xamarin.Forms zu .NET MAUI



CodeMeter – Ein endloser Erfolgskreislauf für Ihr Unternehmenswachstum

SCHÜTZEN SIE IHRE SOFTWARE

mit besten Verschlüsselungs- und Obfuskations-Technologien

ERFÜLLEN SIE WÜNSCHE

Ihrer Kunden mit vielseitiger und skalierbarer Lizenzierung

PROFITIEREN SIE

von Ihrer Arbeit auf globaler Ebene wieder und wieder



Starten Sie jetzt und fordern Sie Ihr CodeMeter SDK an [wibu.com/de/sdk](https://www.wibu.com/de/sdk)



+49 721 931720
sales@wibu.com
www.wibu.com



SECURITY
LICENSING
PERFECTION IN PROTECTION

mit einem kompletten Redesign der Softwarearchitektur zu verbinden. Leider verheben sich einige Teams in dieser Situation, was zum Scheitern der Migration führt. Daher empfehle ich im ersten Schritt, so wenige Architekturänderungen wie möglich während der Migration durchzuführen und diese für einen weiteren Schritt aufzusparen.

Holger: Ein Upgrade vom klassischen .NET Framework auf .NET 8.0 muss wohlüberlegt sein, denn – je nach Anwendungsarten und verwendeten Bibliotheken – kann solch eine Umstellung sehr viel Arbeit machen. Aber auch eine Umstellung von modernen .NET-Versionen wie .NET 6.0 oder 7.0 ist nicht damit getan, dass man `<TargetFramework>net8.0</TargetFramework>` in der Projektdatei setzt. Es gibt zahlreiche Breaking Changes und Verhaltensänderungen, die man beachten muss. Am besten eignen sich automatisierte Tests – Unit Tests und die oft vernachlässigten Integrationstests –, die eine Anwendung nach der Migration komplett testen.

Rainer: Wenn irgendwie möglich, sollte man den Code aktuell halten. Ich weiß aus eigener Erfahrung, dass es nicht leicht ist, alle Projekte immer auf die neuesten .NET-Versionen zu bringen. Oft fehlen die Personalressourcen oder das Budget. Wir sollten uns als Softwareingenieurinnen und -ingenieure aber bemühen, mit unseren Projekten halbwegs auf dem Stand der Technik zu bleiben. Wer schon auf .NET 6.0 oder 7.0 ist, wird mit dem Umstieg auf .NET 8.0 keine großen Probleme haben. Wer allerdings immer noch mit .NET Framework arbeitet, für den ist der Schritt auf neuestes .NET wieder etwas weiter und damit herausfordernder geworden.

Christian: Mit Grausen erinnere ich mich noch an frühere Migrationen, insbesondere von .NET Framework auf frühe Versionen von .NET Core. Inzwischen ist das deutlich einfacher geworden, auch weil das Framework mittlerweile stabil ist und sich nicht mehr so viel Entscheidendes ändert. Erste Migrationen von Codebasen mit .NET 6.0 oder .NET 7.0 auf eine Preview-Version von .NET 8.0 liefen problemlos. Gibt es noch Code, der ältere – und damit nicht mehr unterstützte – Versionen von .NET (Core) voraussetzt, ist natürlich höchste Eile geboten. In der Regel ist es am besten, sukzessive von Version zu Version zu aktualisieren. Beim Sprung von ASP.NET Core 5.0 auf ASP.NET Core 6.0 ist es eine gute Idee, die neue Standardstruktur – Verzicht auf Startup.cs und andere Änderungen – gleich mit zu übernehmen. Mit dem .NET Upgrade Assistant habe ich in der Praxis eher gemischte Erfahrungen gemacht, aber einen Versuch ist es freilich wert.

ix: Und zuletzt – gibt es ein Feature, das ihr in .NET 8.0 vermisst beziehungsweise euch für die Zukunft wünschen würdet?

Thomas: Es gibt nicht das eine zentrale Feature, auf das ich in einer der zukünftigen .NET-Versionen warte. Persönlich verfolge ich natürlich nach wie vor die Desktop-Entwicklung mit WPF, WinForms, WinUI, und neuerdings auch .NET MAUI und Blazor. Bei WinUI habe ich bereits 2019 ein Issue zur fehlenden Eingabevalidierung eröffnet [GitHub Issue unter [ix.de/zjtm](https://github.com/dotnet/winui/issues/ix.de/zjtm)], das leider nach all den Jahren immer noch nicht implementiert ist. Ob und wann das kommen wird, weiß man nicht. Persönlich finde ich den Ansatz mit Blazor Hybrid sehr spannend, da sich damit Komponenten plattformübergreifend für Web-, Desktop-, und Mobile-Apps verwenden lassen.

André: Ich hätte mir gewünscht, dass die Entwicklung responsiver Apps in .NET MAUI weiter erleichtert wird. Obwohl es bereits Möglichkeiten wie Visual States und Adaptive Triggers gibt, ist der Aufwand für wirklich responsive Designs noch zu hoch.

Holger: Mir fehlt weiterhin ein Ahead-of-Time-Compiler für viele .NET-Anwendungsarten wie Windows Forms und Windows Presentation Foundation (WPF). In .NET 7.0 gab es AOT nur für Konsolenanwendungen, in .NET 8.0 kommt AOT nur für Minimal-Web-APIs und gRPC hinzu.

Rainer: Oh ja, Roles and Extensions. Ich verfolge das Design dieses Sprachfeatures schon seit Längerem. Leider hat es den Weg in C# 12.0 nicht gefunden. Ich hoffe, es kommt in C# 13.0. Interfaces und Extension Methods sind in C# und speziell ASP.NET Core allgegenwärtig und das, was Roles and Extensions bringen wird, hebt das Konzept auf ein komplett neues Level. Wer so wie ich das Feature im Auge behalten möchte, kann das auf GitHub machen. Dort findet man erste Beispiele, die demonstrieren, welcher C#-Code damit möglich werden wird.

Christian: .NET MAUI stabilisieren und mehr „It just works!“-Erlebnisse hinsichtlich Authentifizierung und Identity. In den beiden letztgenannten Punkten hat Microsoft in .NET 8.0 einiges getan, aber eine solide Implementierung des für viele SPA-Anwendungen gesetzten BFF-Musters – Backends for Frontends –, um Token im Local Storage zu vermeiden, sehe ich noch nicht – und greife zum kommerziellen Duende IdentityServer. (mai)

Das Interview führte Maika Möbus, Redakteurin bei ix.

Quellen

Alle von den Interviewten genannten Links finden sich unter ix.de/zjtm

Es gibt **10** Arten von Menschen. iX-Leser und die anderen.



Jetzt Mini-Abo testen:
3 Hefte + Bluetooth-Tastatur
nur 19,35 €

www.iX.de/testen



3 x als Heft



> Wie IT-Neulinge und erste Projekte zusammenfinden



Andreas Monschau

In Zeiten des Fachkräftemangels setzen viele Unternehmen vermehrt auf Quereinsteiger. Um diese fürs erste Projekt fit zu machen, gibt es einiges zu beachten.

Im politischen Kontext kommt das Schlagwort vom Fachkräftemangel häufig zum Einsatz – verneint von den einen, bekräftigt von den anderen. Statistiken belegen, dass sich grundsätzlich nicht bestreiten lässt, dass allein im Jahr 2022 insgesamt 137.000 offene IT-Stellen in Deutschland unbesetzt geblieben sind (alle Quellen zum Artikel unter ix.de/z84c).

Abgesehen davon, dass sich Recruiter schwertun, passende Bewerberinnen und Bewerber zu finden, könnten unattraktive Gehälter ausgeschriebener Stellen ein Grund für nicht besetzbare Stellen sein. Neben den offenen IT-Stellen gibt es ein weiteres großes Problem: Unternehmen und Behörden im IT-Umfeld wollen ihre Projekte in der Regel mit fachlich kompetenten Menschen besetzen, an denen es allerdings an allen Ecken mangelt – das betrifft nicht nur die Softwareentwicklung, sondern auch DevOps, Softwaretests (hier insbesondere die Testautomatisierung), Security, Datenschutz oder agile Praktiker.

Im Idealfall besteht ein stetiger Zugriff auf fähige Menschen im Mid- und Seniorlevel, doch leider gibt der (Personal-)Markt die passenden Personalien oftmals nicht her. Eine vielversprechende Alternative könnten daher Quereinsteigerinnen und -einsteiger sein – Menschen, die nur darauf warten, sich in der IT-Branche beweisen zu können. Häufig wird ihnen jedoch die Chance verwehrt, da die Meinung vorherrscht, dass sie „eigentlich nichts von dem können, was im Projekt wirklich gebraucht wird“.

Wie können sie dennoch eine Chance erhalten? Was muss sich ändern, um Menschen zu finden, in denen das Potenzial für mehr schlummert? Welche Unterstützung müssen sie erhalten, um zu Projektmitarbeiterinnen und -mitarbeitern zu werden, die echten Mehrwert schaffen?

Der vorliegende Erfahrungsbericht zeigt auf, welche Maßnahmen wir als IT-Consulting-Unternehmen ergriffen haben,

um ein erfolgreiches Traineeprogramm (nicht nur) für Quereinsteiger über einen Zeitraum von fast zehn Jahren aufzubauen. Der Artikel stellt einige Lösungsstrategien vor und beleuchtet den Umgang mit Rückschlägen.

Woher kommen Quereinsteiger?

Die Gründe für den Quereinstieg in den IT-Bereich können mannigfaltig sein. Manche Kandidatinnen und Kandidaten interessieren sich in ihrer Freizeit für Computer, andere wiederum lockt ein besseres Gehalt. Es war faszinierend zu sehen, wie individuell und vielfältig die Beweggründe der Bewerberinnen und Bewerber waren, am Trainingsprogramm teilzunehmen. Das Spektrum war groß: angefangen bei Studienabsolventen unterschiedlichster Fachrichtungen bis hin zu Studienabbrechern oder ausgebildeten Fachinformatikern.

Die Suche nach geeigneten Kandidaten

Geeignete Quereinsteiger benötigen spezifische Eigenschaften: Neugier und Biss, sich in neue Themen einzuarbeiten, sind wichtig. Eine schnelle Auffassungsgabe und hohe Lernbereitschaft sind vorausgesetzt. Zwingend erforderlich ist auch eine gute Reflexionsfähigkeit und Belehrbarkeit – das sind Punkte, auf die wir großen Wert legen und an denen im Laufe der Jahre eine geringe Anzahl an Trainees gescheitert sind.

Gewisse technische Grundkenntnisse und Erfahrungen sind durchaus von Vorteil, viel wichtiger ist allerdings, wie eine Kandidatin oder ein Kandidat mit (neuen) Herausforderungen zurechtkommt. Nach einem ersten Kennenlernen, in dem die grundsätzlichen Eckdaten besprochen werden, bekommen die Kandidaten eine kleine zu bearbeitende Aufgabe mit einem Zeitaufwand von 20 bis 60 Stunden, je nach bereits vorhandenem Wissensstand. Wenn beispielsweise

das Interesse am Java-Traineeprogramm besteht, geht es um eine praxisnahe Implementierungsaufgabe mit vorgegebenen Randbedingungen. Je nach Vorkenntnissen ist diese Aufgabe dann für den Kandidaten mehr oder weniger anspruchsvoll. Sie umfasst immer den Einsatz von Technologien, die sie nicht ausreichend beherrschen, denn das selbstständige Einarbeiten ist Teil der Aufgabe.

Wer sich mit dieser Art des Lernens und Arbeitens schwer tut, würde auch nachher als Trainee nicht glücklich werden. Damit ergibt sich eine erste Selektion. Es kommt gelegentlich vor, dass sich Kandidaten nach der Aufgabenstellung zurückziehen und nichts mehr von sich hören lassen – ein solches Verhalten lässt sich meist als Absage werten. Andere wiederum stellen sich der Herausforderung, wachsen daran und lösen die Aufgabe – das sind in der Regel genau die Leute, die wir suchen. Wir ermutigen Kandidaten dazu, einen Zeitplan zu erstellen und uns vorab zuzusenden – dadurch ist es möglich, einen Einblick in die Gedankengänge zu erhalten.

Ist die Aufgabe aus Sicht der Kandidaten erfolgreich abgeschlossen, führt ein Trainer aus unserem Traineeprogramm ein umfangreiches Code-Review durch. Da es sich bei der Aufgabenstellung um einen übersichtlichen Geschäftsfall

handelt, ähneln sich die Lösungen immer wieder (ähnliche Google-Suchen führen zu vergleichbaren Tutorials oder Stack-Overflow-Beiträgen), es ist aber ebenso bemerkenswert zu sehen, wie kreativ Menschen bei der Bewältigung von Problemen sein können – insbesondere, wenn sie so gut wie keinen IT-Background haben und sich ohne Vorkenntnisse der Aufgabe nähern.

In der Regel führt der Review zu einem weiteren, vorrangig technischen Interview. In diesem Fall ist es wichtig, nicht einfach nur Fachwissen abzufragen (das in der Regel meist ohnehin nicht existiert), sondern zu prüfen, ob die Kandidaten verstehen, was sie abgeliefert haben. Dabei sollten sie die folgenden Fragen beantworten können:

- Haben die Kandidaten den Code selbstständig implementiert? (Natürlich ist es legitim, sich Hilfe zu holen, um zumindest einen Anfang zu finden. Wenn die komplette Lösung aber von Dritten oder ChatGPT zugeliefert wird, so fällt das spätestens bei der nächsten Frage auf.)
- Können sie den eigenen Code verstehen und in eigenen Worten erklären?
- Sind sie in der Lage, spontan Änderungen am Code durchzuführen?

cid.com

It's a spaceship.



Unsere Kunden haben eine Vision für die Zukunft ihres Unternehmens. Die Herausforderung besteht darin, diese zu erreichen.

Bei CID entwickeln wir maßgeschneiderte Software, die echte Wettbewerbsvorteile schafft. Wir nennen das Advantage Engineering.

Werde Teil unseres Teams!

Über 300 Engineers warten bereits auf dich.



Digitale Prozesse, Modern Data Stacks und KI, Cloud.

Unsere SMART FACTORY! Principles machen den Unterschied. Seit 27 Jahren.

Wir sind Experten darin, Experten im Business unserer Kunden zu werden.

Wir arbeiten eng mit unseren Kunden zusammen, auch in hybriden Squads, damit sie langfristig die Kontrolle behalten.

Wir arbeiten agil und entwickeln Produkte, keine Projekte, für Qualität und Langlebigkeit.

Wir bieten tiefe Expertise und breite Erfahrung in Cutting-Edge Software, Künstlicher Intelligenz, und der Cloud.

Aus den Antworten auf diese Fragen ergeben sich bereits viele Erkenntnisse:

- Sind die Kandidaten lernfähig?
- Welche Herangehensweise haben sie gewählt?
- Können sie die Sachverhalte, die für sie unter Umständen völlig neu sind, erklären?
- Kommen sie mit Stresssituationen zurecht?
- Wie belehrbar sind sie und wie gehen sie mit direktem Feedback um?

Für uns als Unternehmen ist die Fähigkeit wichtig, mit direktem Feedback umzugehen – wir reden offen und ehrlich miteinander und wollen daraus lernen. Wenn die Fragen zufriedenstellend beantwortet wurden und sich alle Beteiligten sicher sind, gemeinsam den nächsten Schritt gehen zu wollen, steht einem Start in das Traineeprogramm nichts mehr im Weg. Der beschriebene Prozess orientiert sich am Umgang mit Quereinsteigerinnen und Quereinsteigern im Bereich Softwareentwicklung. Für andere Spezialisierungen, wie DevOps oder auch Testing, lassen sich ähnliche Aufgaben kreieren und nach dem genannten Muster bewerten.

Was ist das Ziel?

Über die Zeit hat es sich als richtig erwiesen, die neuen Trainees genau dort abzuholen, wo sie stehen, und ihnen eine Liste an Themen zu geben, in die sie sich anhand einer praktischen Aufgabe intensiv einarbeiten. Wichtig ist hierbei, dass es sich um echte Geschäftsprozesse handelt und nicht um abstrakte akademische Aufgaben. Das hat sich für die individuelle Motivation als vorteilhaft erwiesen und bildet den späteren Projektalltag genauer ab.

Fachlich wird eine solche Aufgabe einen Trainee während seines ganzen Traineelebens auf die eine oder andere Weise begleiten. Zum Thema Begleitung: Die Trainees sind nicht auf sich selbst gestellt – ein Trainerteam unterstützt, fördert und fordert sie. Dazu später mehr.

Das große Ziel des Programms ist es, Menschen so weit zu befähigen, dass sie guten Gewissens in einem Kundenprojekt zum Einsatz kommen und Mehrwert schaffen –

- für den Kunden (der darauf vertraut, dass der Quereinsteiger gut vorbereitet wurde)
- für den Arbeitgeber (der in Form des Traineeprogramms in sie investiert hat)
- und für sie selbst.

Am Ende können sich die Teilnehmenden gewiss sein, in kurzer Zeit anspruchsvolle Inhalte gelernt zu haben und diese anwenden zu können.

Was können Menschen in kurzer Zeit lernen?

Die Länge des Traineeprogramms ist zeitlich begrenzt – im Schnitt gehen wir von sechs bis sieben Monaten intensiven Trainings aus, bis jemand beispielsweise als Junior DevOps-Engineer erfolgsversprechend in ein Kundenprojekt eintreten kann. Der Fokus liegt in dieser Zeit auf den wesentlichen Werkzeugen, die für das spätere Projektleben notwendig sind. Gleichzeitig ist es wichtig, im Umgang mit bestimmten Tools tiefgreifende Kenntnisse zu erlangen, beispielsweise mit Terraform oder Ansible.

Das Traineeprogramm bringt einen fordernden Balanceakt mit sich, der sich mit entsprechender Vorbereitung und Organisation meistern lässt. Es bewährt sich, beispielsweise im Umfeld Softwareentwicklung, auf einen standardisierten, marktrelevanten Technologie-Stack zu setzen. Es gilt, die Technologien abzudecken, die die potenziellen Kunden in ihren Projektausschreibungen fordern. Innerhalb dieses Rahmens ist es dann möglich, an vielen Stellen den Blick über den Tellerrand zu werfen.

Den Markt im Blick behalten

Aus vertrieblicher Sicht sollten Trainees beziehungsweise Alumni (erfolgreiche Absolventen) möglichst breit aufgestellt sein, was die Kenntnisse bezüglich verwendeter Werkzeuge betrifft. Allerdings lohnt es sich, genauer hinzuschauen, da sich sonst der Umfang des Programms schnell aufbläht.

Um herauszufinden, welche Skills Kunden fordern, ist eine stetige Analyse des Markts notwendig. Der Vertrieb behält die Bedürfnisse der Kunden im Blick, daher ist ein stetiger Austausch zwischen vertrieblichen Kollegen und dem Trainerteam zwingend notwendig.

Journey to Junior

Man kann nicht direkt mit der Tür ins Haus fallen – Wissen will kontinuierlich aufgebaut werden. Bevor sich an die spannenderen und komplexeren Themen gewagt wird, steht zunächst ein theoretischer, grundlegender Exkurs an – das gilt für alle Traineespezialisierungen.

Die bereits erwähnte praxisbezogene Aufgabenstellung über mehrere Aufgabenblöcke bringt für jeden neuen Block neue Techniken und Tools mit sich. Das führt häufig zu Aha-Erlebnissen, nachdem viele Dinge zunächst per Hand gemacht worden sind. Diese Erlebnisse sind wichtig, und zwar im ganzen Traineeprogramm, denn sie sagen im Grunde Folgendes aus: „Ich habe erkannt, was ich hier mühsam selber gemacht habe – und hier gibt es etwas, das meine Arbeit abnimmt, und ich bin in der Lage zu verstehen, wie es funktioniert!“ Das fördert die Motivation und sorgt dafür, dass sich die Trainees mit Eifer auf die Aufgaben konzentrieren.

Betreuung durch Trainer

Jeder Trainee wird für eine bestimmte Zeit durch einen Trainer betreut, der ihm Feedback zu den gelieferten Lösungen gibt und ihn bei Problemen sowie Fragen unterstützt. Bei den Trainern handelt es sich um Mitarbeiter mit langjähriger Erfahrung. Zum Teil waren sie bereits in der betrieblichen Ausbildung von Fachinformatikern tätig, andere sind ehemalige Trainees, die das Programm mit hervorragenden Ergebnissen durchlaufen haben. Alle sind schwerpunktmäßig in Kundenprojekten tätig und haben für ihre Trainees ein Zeitbudget von maximal vier Wochenstunden.

Meist wechseln die Trainer zwischen Aufgabenblöcken, damit gewährleistet ist, dass die Trainees regelmäßig unterschiedliche Sichtweisen auf verschiedene Themen erhalten – denn auch Trainer sind Individuen, die sich zwar an die Inhalte des Programms halten, aber dennoch unterschiedliche Einstellung haben – wie im echten Projektleben.

Jeder Trainer betreut seine Trainees nach eigenen Vorstellungen: Manche führen feste Termine ein, andere arbeiten auf Zuruf – diese Freiheit gewährt das Unternehmen gerne, da die Ergebnisse stets stimmen.

Wissen weitergeben und sich vernetzen

Erwerben von Wissen ist das eine – es weiterzugeben, ist jedoch etwas anderes. Der Austausch unter den Trainees wird aktiv gefordert und funktioniert in der Regel auch reibungslos. Darüber hinaus halten wir es für sinnvoll, wenn die Trainees in regelmäßigen Abständen kurze Vorträge über technisch relevante Themen halten. Im Regelfall besteht die Zuhörerschaft aus den anderen Trainees aller Fachrichtungen, aus den Trainern sowie aus interessierten Kollegen. Da-

rüber hinaus ermutigen wir die Trainees, beispielsweise an Java-User-Group-Treffen teilzunehmen – zum einen, um von den Fachvorträgen zu lernen, aber auch, um sich mit anderen Entwicklerinnen und Entwicklern zu vernetzen.

Wir halten es für wichtig, dass die jeweils aktuelle Trainee-Generation mit den Alumni ins Gespräch kommt. Kolleginnen und Kollegen auf Seniorlevel können viel über ihre Projekte erzählen, aber die Erfahrungen der Alumni sind für die Trainees besonders wertvoll und nehmen ihnen in vielen Fällen etwas von dem Druck oder der Angst vor dem ersten Zusammentreffen mit Kunden.

Lernzielkontrolle

Obgleich die Trainees unter Betreuung mindestens eines Trainers stehen, der sich auch die Zwischenergebnisse ihrer Aufgaben anschaut, wird früher oder später eine Standortbestimmung notwendig. Die folgenden Umstände sind der Auslöser:

- Ein Trainee hat einen Aufgabenblock seiner Journey to Junior beendet.
- Ein Trainee lernt fleißig, aber es herrscht Uneinigkeit unter den Trainern, wie gut er oder sie den vermittelten Stoff beherrscht.
- Das letzte Viertel der Trainingszeit wurde nicht erreicht und es wird geprüft, wie der Markt auf das Profil reagiert.

Es stehen nun mehrere Instrumente zur Verfügung: Zum einen kann der Trainer seinem Schützling eine Aufgabe stellen, die er oder sie innerhalb einer Zeitspanne lösen muss. Hier bietet es sich an, die Aufgabe so zu formulieren, dass sie sowohl bekannte als auch unbekannte Inhalte berück-





heise Jobs
IT TAG

DIE IT-JOBMESSE

Das erwartet dich:
Bewerbungsfotos
Lebenslaufcheck
Vorträge
Catering

TERMINE:

Stuttgart 09.04.2024	München I 18.04.2024	Leipzig 25.04.2024	Hannover 16.05.2024
Frankfurt 28.05.2024	Köln I 19.06.2024	Nürnberg 25.06.2024	

Die Veranstaltungen sind kostenlos.

powered by  heise Jobs

sichtigt, die der Trainee dann in kurzer Zeit erarbeiten muss. Nach dem Abschluss dieser Challenge führt der Trainer einen Review des Ergebnisses durch und bespricht ihn gemeinsam mit dem Trainee. Gegebenenfalls kommen beide zu dem Entschluss, weitere Schritte einzuleiten, die zuvor oberflächlich bearbeitete Inhalte vertiefen.

Alternativ ist es möglich, Interviewtrainings durchzuführen, die den Trainees technisch intensiv auf den Zahn fühlen. Dabei geht es nicht darum, jemanden bloßzustellen, sondern Lücken zu identifizieren, die sie im Anschluss selbst oder mit Hilfe der Trainer füllen.

Am Ende des Weges

Irgendwann ist jede Journey-Aufgabe bearbeitet, man kann den Technologie-Stack in- und auswendig herunterbeten, hat diverse Interviewtrainings hinter sich und das Profil ist öffentlich – was nun? Eigentlich könnte es immer so weitergehen, bis sich das passende Projekt gefunden hat, allerdings gilt es Folgendes zu beachten: Auch Trainer-Kapazitäten sind endlich, und stets nehmen neue aussichtsreiche Trainees am Programm teil. Daher gelten die Trainees ab dem Zeitpunkt, an dem ihre Journey abgeschlossen ist, als vorerst ausgelernt. Sie verdienen sich ihre ersten Sporen in einem internen Entwicklungsprojekt. Hier können sie gemeinsam mit anderen Trainees und weiteren Mitarbeitern weiterarbeiten und lernen. Parallel dazu beschäftigen sie sich mit eigenen Wunschthemen oder speziellen Tools, die der Vertrieb gerne im Skillset sieht. Früher oder später ist es so weit – der erste Kunde klopft an, das erste Projekt startet und die Trainees können das Erlernte zur Anwendung bringen.

Erfahrungswerte

In den letzten Jahren haben in dem Unternehmen, bei dem der Autor des Artikels tätig ist, über 30 Mitarbeiterinnen und Mitarbeiter das Traineeprogramm durchlaufen – mehrheitlich erfolgreich. Wenn es jedoch zu Problemen kam, die schließlich zur Auflösung des Traineeverhältnisses führten, lag es vorrangig daran, dass (ehrlicherweise gesagt auch gegenseitig) gewisse Erwartungen nicht erfüllt wurden, was einerseits die Verfügbarkeit der Trainer und andererseits die erwartete Leistungsbereitschaft betraf. Jedes Mal, wenn ein Trainee vorzeitig ausschied, haben wir die Gründe genau analysiert und sowohl das Training wie auch besonders das Auswahlverfahren nachgeschärft. Durch diese Lernkurve hat sich die Abbrecherquote von mehr als 50 Prozent in den ersten Jahren auf 9 Prozent im vergangenen Jahr deutlich reduziert. Unser Ziel ist es, nur solche Kandidaten an Bord zu nehmen, bei denen wir uns sehr sicher sind, dass sie erfolgreich sein werden.

Bei Betrachtung der Statistik der letzten Jahre ist es hochinteressant, aus welchen Branchen die Kandidatinnen und

Kandidaten teilweise kamen – von (absolvierten wie abgebrochenen) Studiengängen wie BWL, Chemie oder Lehramt bis hin zu Ausbildungen wie Steuerfachgehilfen oder Rettungssanitäter hat sich eine große Bandbreite an Menschen auf das Programm beworben – und konnten den Anforderungen gerecht werden.

Mit offenen Karten spielen

Das Traineeprogramm hat sich für unser Unternehmen als sehr erfolgreich erwiesen, viele der Kolleginnen und Kollegen haben es durchlaufen und rein rechnerisch liegt jetzt eine Erfolgsquote von 90 Prozent vor. Die Übergänge in die ersten Kundenprojekte verliefen in den meisten Fällen glatt, natürlich war es auch schon mal holprig: Die Erwartungshaltungen der Kunden wurden nicht vollständig getroffen. Das ist allerdings kein Problem, das ausschließlich den Einsatz von Junioren betrifft.

Gegenüber Projektkunden ist es wichtig, mit offenen Karten zu spielen – wir kommunizieren klar, dass es sich bei dem Mitarbeiter um einen Quereinsteiger handelt, der auf Juniorlevel gehoben wurde. Wir weisen allerdings im Zuge dessen auch darauf hin, was die Mitarbeiterin beziehungsweise der Mitarbeiter in den vergangenen Monaten geleistet hat und welche Entwicklung hinter ihr/ihm liegt.

Ein wie hier beschriebenes Traineeprogramm kann ein Mittel zur Lösung vieler Probleme sein – zum Beispiel versetzt es ein Unternehmen in die Lage, passende Mitarbeiterinnen und Mitarbeiter zu finden, zu fördern und zu formen. Gleichzeitig öffnet es vielen willigen Quereinsteigern die Tür.

Diese Chance haben sie sich redlich verdient, denn schließlich kann sich aus der anfänglichen Annahme „der kann ja nichts von dem, was wir brauchen“ die Überzeugung entwickeln: „Ja, der kann genau das, was wir auch machen.“ (mdo)

Quellen

Alle Links zum Artikel finden sich unter ix.de/z84c.

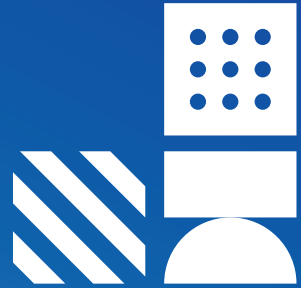


Andreas Monschau

ist seit über 10 Jahren als Senior IT-Consultant mit den Schwerpunkten Softwarearchitektur- und Entwicklung sowie Teamleitung bei Haeger Consulting in Bonn tätig und aktuell als Solution Designer im Kundenprojekt unterwegs. Neben seiner Projektstätigkeit leitet er das umfangreiche Traineeprogramm des Unternehmens.

> Entwurfsmuster in der Softwareentwicklung

Nils Kasseckert



Softwareentwicklung ohne Entwurfsmuster ist vergleichbar mit einer Suppe ohne Salz. Design Patterns sind fundamental wichtig für den Job, stellen Neulinge aber vor große Hürden.

Entwurfsmuster ähneln einem modularen IKEA-Regalsystem, das in verschiedenen Größen und Farben erhältlich ist. Sie lassen sich immer gleich kombinieren und erfüllen stets denselben Zweck. Die Farbe ist mit der Programmiersprache vergleichbar, die Regalgröße mit dem Entwurfsmuster. Je nach Anwendungsfall und Projektgröße bietet sich die Verwendung eines Regals in einer anderen Farbe oder Größe an.

Entwurfsmuster bieten Lösungsansätze für typische Probleme in der OOP-Welt und sind auch bei immer komplexer werdenden Frameworks aktuell. Sie haben kein festgelegtes Design und beinhalten keinen fertigen Code. Developer können sie unabhängig von der jeweiligen Programmiersprache einsetzen.

Aktuell gibt es mehr als 100 verschiedene Entwurfsmuster. Jedes Entwurfsmuster lässt sich einem Typus zuordnen. Die grundlegenden Typen sind: Erzeugungsmuster, Strukturmuster sowie Verhaltensmuster. Inzwischen gibt es aber auch weitere Typen. Sie sind aufgrund der immer größer werdenden Menge an Entwurfsmustern hinzugekommen, um das Feld weiter aufzuteilen. Der Einfachheit halber beschränkt sich dieser Artikel auf die grundlegenden Typen.

Der erste Typ sind Erzeugungsmuster. Sie dienen ausschließlich der Erzeugung neuer Instanzen und ermöglichen die Erstellung eines neuen Objektes unabhängig von seiner konkreten Implementierung. Dadurch ist die Objekterstellung ausgelagert und gekapselt. Die am häufigsten genutzten Erzeugungsmuster sind Singleton und Factory-Method.

Ein Unikat erstellen

Ein Singleton kann man sich wie das OOP-Äquivalent einer globalen Variable ohne Zugriffsschutz vorstellen. Es ist während der gesamten Laufzeit nur ein einziges Mal erzeugbar, weshalb in einem System maximal eine Instanz eines Single-

tons existiert. Es kapselt die Verwaltung der Instanzen und stellt einen einheitlichen Zugriffspunkt zur Verfügung.

Jedes Singleton besteht aus einer privaten statischen Variable und einer veröffentlichten statischen Methode. Die Variable enthält die aktuelle Instanz des Objektes über seinen gesamten Lebenszyklus. Die öffentliche Methode `getInstance()` gibt eine statische Variable zurück. Nur das Objekt selbst darf die Instanz erzeugen. Ein privater Konstruktor stellt das sicher.

Das Konstrukt ist leicht anzuwenden und zu verstehen. Zudem vereinfacht es Architekturentscheidungen. Dem stehen aber auch einige Nachteile entgegen. Zum einen benötigt man bei jedem Zugriff auf ein Singleton den vollständigen Klassennamen. Dadurch entsteht eine starke Kopplung an den konkreten Typ, was bei automatisierten Softwaretests ein großes Problem darstellt. Dort ersetzen Dummy-Objekte normalerweise nicht benötigte Objekte. Das ist mit einem Singleton jedoch bei klassischen Programmiermethoden nicht möglich. Zum anderen verschlechtert der Einsatz von Singletons wegen der fehlenden Zugriffskontrolle die Systemarchitektur.

Entkopplung von Klassen

Das Erzeugungsmuster Factory Method ähnelt einer Fabrik, die für die Produktion von Objekten zuständig ist. Ein Objekt erzeugt sich bei dieser Methode nicht selbst und ist auch nicht von einer konkreten Klasse abhängig – so wie es beim Singleton der Fall ist. Stattdessen erstellt die statische Factory-Methode ein Objekt in einer Factory-Klasse.

Problematisch ist jedoch die Abhängigkeit des Objekts zu einem speziellen Typ. Aus diesem Grund arbeitet das Erzeugungsmuster Factory-Method mit Interfaces anstelle von konkreten Typen. Weiterhin kann die Factory Method in der -Klasse einen Parameter enthalten, der die Art des zu erzeug-

genden Objektes bestimmt. Somit lassen sich beispielsweise unter Einsatz von globalen Variablen in der Testumgebung andere Objekte erzeugen als außerhalb der Testumgebung.

Anstelle von Interfaces ist die Verwendung von Unterklassen möglich. Mit dem Erzeugungsmuster Factory Method lassen sich die mitunter schwierig lesbaren Konstruktoren selbst formulieren. Zudem verwaltet es Singletons und gibt je nach Parameter verschiedene Klassen zurück.

Die Notwendigkeit von zusätzlichen Factory-Klassen sowie die benötigte Abhängigkeit zu Interfaces beziehungsweise Oberklassen stellen die einzigen großen Nachteile des Musters dar. Auf der anderen Seite können Entwicklerinnen und Entwickler durch den Einsatz von Factory-Methoden erfahrungsgemäß ihren Code besser testen und warten. Ohne dieses Konstrukt wäre ein einfaches Austauschen der Klassen nicht ohne Weiteres möglich. Häufig verwendet man in der Praxis die Dependency Injection – wie auch im Java-Framework Spring. Sie führt im weitesten Sinne das Erzeugungsmuster automatisiert aus. Das Erzeugungsmuster Factory Method läuft dabei im Hintergrund; Developer müssen sich nicht um die Objekterzeugung kümmern.

Beziehungen definieren

Strukturmuster sind die zweite Kategorie grundlegender Entwurfsmuster. Sie spezifizieren die Beziehungen zwischen einzelnen Entitäten. Ziel ist es, den Entwurf der Software zu erleichtern. In der Praxis verwendet man dafür häufig die Strukturmuster Adapter und Proxy.

Adapterobjekte kommen zum Einsatz, wenn die Schnittstelle einer existierenden Klasse nicht zu der benötigten Schnittstelle passt. Häufig ist das bei Bibliotheken der Fall, da sie in der Regel unveränderbar sind. Es bietet sich auch die Nutzung von wiederverwendbaren Klassen an, die mit nicht vorhersehbaren anderen unabhängigen Klassen zusammenarbeiten sollen.

Ein Beispiel ist die Entwicklung einer App, die die aktuellen Lottozahlen im XML-Format vom Lottoserver herunterlädt und sie in Listenform darstellt. Nach einer Aktualisierung zeigt sie die letzten Lottozahlen grafisch aufbereitet an. Um diese Ausgabe zu ermöglichen, verwenden Entwicklerinnen und Entwickler eine bereits existierende Grafikbibliothek. Sie benötigt jedoch Daten im JSON- und nicht im XML-Format. Also ist ein Adapter für die Client-Klasse notwendig, die die Daten vom Lottoserver abrufen. Er parst die XML-Daten ins JSON-Format, damit die Grafik-Bibliothek arbeiten kann. Den Adapter kann man entweder als Objektadapter oder als Klassenadapter umsetzen.

Beim Objekt-Adapter-Ansatz stellt die existierende Client-Klasse – hier die Lotto-App – ein Interface zur Verfügung. Der

Adapter implementiert sie und kommuniziert mit einer Serviceklasse – hier GrafikLib. Er hält eine Referenz zur Serviceklasse, bringt die Daten in das korrekte Format und ruft anschließend die passende Service-Methode auf.

Die Client-Klasse ist aufgrund der Nutzung von Interfaces nicht an einen speziellen Adapter gebunden. Je nach Anforderung ist es möglich, den Adapter durch einen anderen auszutauschen, ohne den Client-Code anzupassen. Das Schema entspricht dem Open/Closed-Prinzip.

Voraussetzung für einen Klassen-Adapter ist die Möglichkeit der Mehrfachvererbung. Er erbt die Eigenschaften und Methoden sowohl von der Client- als auch von der Serviceklasse. Der Adapter überschreibt dann die jeweiligen Methoden, was die Zusammenarbeit mit verschiedenen Klassen ermöglicht. Anstelle der Client-Klasse verwendet der Entwickler nur noch die Adapterklasse. Das erhöht die Abhängigkeit zwischen den Klassen, weshalb ein flexibler Austausch ohne größere Codeanpassungen nicht möglich ist.

Adapter führen zu komplexerem Code, da Entwickler neue Interfaces und Klassen einführen müssen. Je nach Anwendungsfall und Möglichkeit ist es sinnvoll, die Serviceklasse so abzuändern, dass sie zum restlichen Code passt. Dennoch ermöglicht das Entwurfsmuster Adapter eine große Wiederverwendbarkeit und Flexibilität beim Programmieren.

Unauffällige Kommunikation

Das Strukturmuster Proxy bietet sich für die Steuerung der Zugriffskontrolle oder für das verzögerte Laden von Bildern, Videos und Audiodateien an. Die Steuerung von Zugriffskontrollen ist ein klassisches Problem in der Programmierung. Ein Beispiel veranschaulicht das: In einer Datenbank existieren Daten, die nur bestimmte Nutzer sehen dürfen. Eine Zugriffskontrolle ist nur dann möglich, wenn Klassen nicht direkt auf die Datenbank zugreifen können. Auch bei der Kommunikation zwischen zwei Hardwarekomponenten bietet sich die Verwendung eines Proxys an. Zum Beispiel ist typischerweise die Anzahl der maximal übertragbaren Daten pro Zeiteinheit begrenzt und man muss sie vorher einer Plausibilitätsprüfung unterziehen.

Zwischen dem Client und dem Service (einer Datenbank oder der Hardware) führt man einen Proxy mit derselben Schnittstelle ein. Der Client stellt die benötigte Schnittstelle als Interface bereit. Der Proxy implementiert sie anstelle des eigentlichen Objekts. Er kommuniziert mit dem Service und führt davor und/oder danach weitere Operationen aus, ohne dass der Client etwas von der Zwischenschicht merkt.

Die Vorteile dieser Methode sind klar ersichtlich: Zum einen kann man den Lebenszyklus des Serviceobjektes vom Proxy zentral steuern. Der jeweilige Client braucht sich da-

rum nicht mehr zu kümmern. Besonders bei aufwendigen Operationen wie dem Aufbau einer Datenbankverbindung kann er Laufzeit einsparen. Weiterhin funktioniert der Proxy, auch wenn der Service noch nicht zur Verfügung steht, und kann dementsprechend reagieren. Ein Proxy-Objekt ist austauschbar, ohne dass man dafür den Client oder Service ändern muss – dies entspricht ebenfalls dem Open/Closed-Prinzip.

Nachteilig ist, dass dieses Strukturmuster neue Klassen benötigt. Außerdem muss der Proxy die Daten mit einer gewissen Verzögerung zurückgeben können, asynchrone Programmierung ist also empfehlenswert.

Zuständigkeiten bestimmen

Der letzte Typ der aufgezählten Entwurfsmuster sind Verhaltensmuster. Sie beschäftigen sich vorrangig mit der Modellierung von komplexem Verhalten der Software. Hierzu gehört die Zuteilung von Zuständigkeiten an Objekte, aber auch Algorithmen, um die Flexibilität einer Software hinsichtlich ihres Verhaltens zu erhöhen. Sie zu verstehen, stellt für Anfänger oftmals eine besondere Hürde dar.

Möchten Objekte von der Zustandsänderung eines anderen Objekts erfahren, bietet sich das Verhaltensmuster Observer an. Um es besser zu verstehen, hier ein Beispiel: Es existieren die Objekte Kunde und Laden. Der Kunde möchte gerne als Erster ein neues Smartphone kaufen. Deshalb fährt er mehrfach am Tag zu einem bestimmten Laden und überprüft, ob das Gerät vorrätig ist. Da sich das neue Smartphone jedoch noch auf dem Lieferweg befindet, bleiben die Besuche ohne Erfolg. Würde der Laden eine E-Mail an alle Kunden schicken und sie darüber informieren, dass das Gerät verfügbar ist, müsste der Kunde nicht mehr so oft das Geschäft betreten. Andererseits könnten sich nicht interessierte Kunden über diese Benachrichtigung ärgern und sie als Spam einstufen.

Dieser Konflikt lässt sich durch das Verhaltensmuster Observer lösen. In diesem existieren im Wesentlichen die Akteure Publisher und Subscriber. Der Publisher (der Laden) stellt Informationen zur Verfügung. Der Subscriber (der Kunde) möchte hingegen die Informationen erhalten. Der Subscriber registriert sich beim Publisher, der ihn bei definierten Ereignissen benachrichtigt.

Der Publisher besitzt neben seiner eigentlichen Logik verschiedene Methoden, mit denen sich Subscriber an- und abmelden. Er kann sie auch benachrichtigen. Dabei meldet sich die Subscriber-Klasse nicht direkt, sondern über ein Interface an. Das Interface definiert die Nachrichtenschnittstelle. Meist besteht sie aus einer einzigen Update-Methode mit einem Kontextobjekt. Das Kontextobjekt beinhaltet zum Beispiel den Eventnamen oder weitere Eventdetails. Der konkrete Subscriber implementiert das Interface. Dadurch kann er sich benachrichtigen lassen. Für die Verknüpfung beider Elemente ist eine zusätzliche Client-Klasse notwendig. Diese erzeugt sowohl die Publisher- als auch Subscriber-Klasse und registriert den Subscriber beim Publisher.

Durch dieses Verhaltensmuster lassen sich Beziehungen zwischen Objekten zur Laufzeit abbilden. Es führt zu hoher Wiederverwendbarkeit, Modularität und Flexibilität. Klassische Einsatzgebiete sind beispielsweise die Benachrichtigung bei Netzwerkverlust oder asynchroner Kommunika-



Nichts geht ohne eine starke IT!

Ob bei Energie und Klimaschutz, dem internationalen Warenverkehr, der Wirtschaftsförderung oder der Stärkung von Menschenrechten.

Das BAFA, eine breit aufgestellte Bundesoberbehörde im Geschäftsbereich des Bundesministeriums für Wirtschaft und Klimaschutz (BMWK). Ein kompetenter Dienstleister für Bürgerinnen und Bürger, unterstützt die Wirtschaft vom Startup bis zum internationalen Konzern.

Dicht am politischen Zeitgeschehen stellen wir in kürzester Zeit anspruchsvolle IT-Lösungen zur Verfügung und hierfür brauchen wir Dich als

- Datenbankadministrator (w/m/d)
- Systemadministrator (w/m/d)
- Softwareentwickler (w/m/d) oder
- Mitarbeitenden im Benutzerservice (w/m/d)

Wir suchen Unterstützung für unsere Teams an den Standorten in Eschborn bei Frankfurt am Main sowie in den Außenstellen in Weißwasser/Sachsen, Borna/Sachsen und Berlin.



Bewirb Dich gleich auf unserer Karrierewebsite



Bundesamt
für Wirtschaft und
Ausfuhrkontrolle



tion zwischen verschiedenen Objekten. Die Benachrichtigung folgt keiner Reihenfolge, sondern läuft zufällig ab. Es gibt jedoch auch Nachteile: Meldet sich der Nutzer beim Publisher nicht ab oder meldet er sich mehrfach an, kann dies bei der Software zu nicht definierten Zuständen führen. Solche Fehler sind meist nur schwer zu finden. Besonders bei großen Projekten kann die Änderung eines Objektes zu einem Dominoeffekt führen, wenn ein Subscriber gleichzeitig wiederum ein Publisher ist. Auch ein sich immer wiederholender Aufruf ist dann nicht ausgeschlossen.

Doppelten Code vermeiden

Wenn man in verschiedene Klassen eine ähnliche Funktion implementieren möchte, die sich nur im Detail unterscheidet, eignet sich das Verhaltensmuster Template Method. Es kann eine Art Skelett eines Algorithmus in der Basisklasse definieren, das man in den Unterklassen in bestimmten Schritten überschreiben kann.

Vertiefen lohnt sich!

Das Thema der Entwurfsmuster ist viel größer als hier dargestellt. Es empfiehlt sich, sich mit dem Thema weiter aus-

einanderzusetzen. Aus meinen eigenen Erfahrungen heraus mag gerade zu Beginn die Fülle der Informationen und die Theorielastigkeit abschrecken. Es lässt sich jedoch beim richtigen Einsatz von Entwurfsmustern die Code-Qualität, Testbarkeit sowie Wartbarkeit erheblich verbessern. Besonders bei größeren Softwareprojekten lohnt es sich, mit Entwurfsmustern zu arbeiten. (mdo)

Quellen

Dieser Artikel ist in voller Länge, inklusive Abbildungen, Listings und Literaturhinweisen, bereits auf heise Developer erschienen, siehe ix.de/zqrb.



Nils Kasseckert

arbeitet als Lead Developer bei einem großen Netzbetreiber. Seit 2013 betreibt er zusätzlich sein Nebengewerbe „AppSupporter“ und promoviert im Bereich „Erneuerbare Energien“. Er interessiert sich für Softwarearchitektur, Embedded Systems sowie jegliche neue Technologien.

Impressum We Are Developers!

Redaktion: iX
Telefon: 0511 5232-387, **E-Mail:** post@ix.de

Herausgeber: Ansgar Heise

Chefredakteur (verantwortlich für den Textteil): Dr. Oliver Diedrich

Konzeption und redaktionelle Leitung: Madeleine Domogalla (mdo@ix.de) -590

Redaktion: Maika Möbus (mai@ix.de) -589

Autoren dieser Ausgabe:
 Thomas Claudius Huber, Nils Kasseckert, André Krämer, Andreas Monschau,
 Dr. Holger Schwichtenberg, Rainer Stropek, Christian Wenz

DTP-Produktion:
 Lisa Hemmerling, Heise Medienwerk, Rostock

Korrektur:
 Dörte Bluhm, Lara Bögner, Heise Medienwerk, Rostock

Titelbild:
 Lisa Hemmerling, Heise Medienwerk

Verlag:
 Heise Medien GmbH & Co. KG,
 Postfach 61 04 07, 30604 Hannover; Karl-Wiechert-Allee 10, 30625 Hannover;
 Telefon: 0511 5352-0, Telefax: 0511 5352-129

Geschäftsführer:
 Ansgar Heise, Beate Gerold

Mitglieder der Geschäftsleitung:
 Jörg Mühle, Falko Ossmann

Anzeigenleitung (verantwortlich für den Anzeigenteil):
 Michael Hanke (-167), E-Mail: michael.hanke@heise.de,
www.heise.de/mediadaten/ix

Leiter Vertrieb und Marketing:
 André Lux

Druck:
 Dierichs Druck + Media GmbH & Co. KG,
 Frankfurter Straße 168, 34121 Kassel

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des Verlages verbreitet werden; das schließt ausdrücklich auch die Veröffentlichung auf Websites ein.

Printed in Germany
 © Copyright by Heise Medien GmbH & Co. KG


Inserenten

Bundesamt für Wirtschaft und Ausfuhrkontrolle	Eschborn	17	DATEV eG	Nürnberg	2
BWI GmbH	Bonn	20	Fraunhofer-Institut für Integrierte Schaltungen IIS	Erlangen	5
CID GmbH	Freigericht	11	WIBU-SYSTEMS AG	Karlsruhe	7

Die hier abgedruckten Seitenzahlen sind nicht verbindlich. Redaktionelle Gründe können Änderungen erforderlich machen.

betterCode() building  ct <webdev>


» Continuous Lifecycle » [Container Conf]

 data2day

 enterJS

// heise devSec()

 inside agile

 Minds Mastering Machines

ONLINE & VOR ORT DEVELOPER April – Juli 2024 KONFERENZEN

Weitere Informationen unter:
heise-academy.de/formate/konferenzen

Veranstalter:



 dpunkt.verlag



- **9. – 11.4.2024**
JavaLand (am Nürburgring)
javaland.eu
- **16. und 18.4.2024**
Mastering GitOps (Online)
mastering-gitops.de
- **17. – 18.4.2024**
Digital Design & UX Next (München)
dd-ux.de
- **24. – 25.4.2024**
Minds Mastering Machines (Köln)
m3-konferenz.de
- **7. – 8.5.2024**
enterJS (Mainz)
enterjs.de
- **14. und 16.5.2024**
betterCode() API (Online)
api.bettercode.eu
- **11.6.2024**
Mastering Observability (Online)
mastering-obs.de
- **18. – 21.6.2024**
CloudLand (im Phantasialand/Brühl)
cloudland.org
- **20.6.2024**
betterCode() Spring (Online)
spring.bettercode.eu
- **18. – 19.7.2024**
WeAreDevelopers World Congress (Berlin)
mit heise Developer Area
wearedevelopers.com/world-congress

CYBER_ FORCE

Digitale Transformation der Bundeswehr vorantreiben.
Innovative Technologien entwickeln und bereitstellen.
Für stabile und leistungsfähige IT sorgen.

Was ist deine Mission?

BWIr **b dich jetzt!**

bwi.de/professionals

