

DESIGN MODERNER APPS, TEIL 1

Einstieg: Das App-Konzept

Das Design moderner Apps basiert auf Recherche, Konzept und planvoller Umsetzung. Am Beginn geht es um Umdenken, um Konzepte, Ziele und Ansatzpunkte.

Moderne Applikationen müssen in allen Belangen überzeugen. Neben der Funktionalität spielen Design und User Experience (UX) eine entscheidende Rolle. Für Windows steht mit WinUI 3 ein modernes User Interface Framework zur Verfügung, das technisch in diesen Fragen keine Wünsche offenlässt. Bevor man jedoch zur tatsächlichen Umsetzung kommt, braucht es umfangreiche Vorarbeiten, was Design und User Experience betrifft.

In dieser Artikelserie beschäftigen wir uns mit dem Design von Apps von der Recherche bis zur Umsetzung. Als Grafikframework nutzen wir das brandneue WinUI 3 für moderne Windows-Apps. Viele Ansatzpunkte und Ideen lassen sich jedoch auch auf andere Systeme übertragen, beispielsweise auf .NET MAUI, wenn man plattformübergreifende Apps erstellen möchte.

Motivation

Beschäftigt man sich mit der Entwicklung von Apps für die unterschiedlichen Plattformen, dann wird man schnell feststellen, dass die grafischen Möglichkeiten inzwischen sehr vielfältig geworden sind. Mithilfe von Layout-Containern können die Elemente der Oberfläche passend ausgerichtet werden. Wir haben eine sehr umfangreiche Palette an grafischen Controls zur Verfügung, die mit den „langweiligen“ Eingabefeldern in „Grau-Grau“ nichts mehr gemeinsam haben. Bereits die Basis-Controls eines Grafikframeworks, wie die von WinUI 3 [1], siehe [Bild 1](#), bieten eine sehr umfangreiche Auswahl an Steuerelementen zur Gestaltung des User Interface. Einen noch besseren Überblick bekommen wir, wenn wir die dafür explizit bereitgestellte App WinUI 3 Control Gallery aus dem Microsoft Store [2] installieren ([Bild 2](#)).

Diese reiche Auswahl ist jedoch längst nicht das Ende der Fahnenstange. Drittanbieter wie Syncfusion, Telerik, DevExpress und andere bieten ebenfalls umfangreiche Sammlungen von Controls für die aktuellen Grafikframeworks an, so auch für WinUI 3. Weitere Elemente eines gelungenen UI-Designs sind Farben, Animationen, Styles und dergleichen.

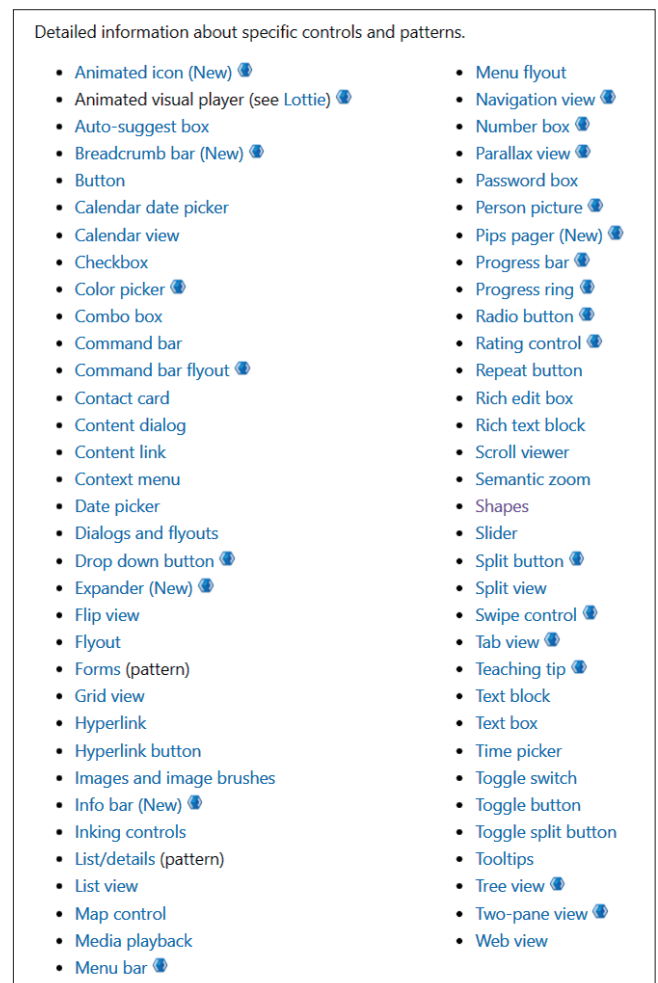
Die Liste ließe sich noch lange fortsetzen, ohne dass wir daraus im Moment einen Nutzen ableiten könnten. Wir können es aber auch kompakter zusammenfassen: Die Möglichkeiten der Gestaltung der Oberflächen, zum Beispiel mit WinUI 3 für Windows, sind heute grafisch sehr reichhaltig und daher mit den älteren Technologien wie Windows Forms oder WPF kaum mehr zu vergleichen.

Aus der Vielfalt der Möglichkeiten ist jedoch noch nicht abzuleiten, wie man zu einem passenden Design für eine mo-

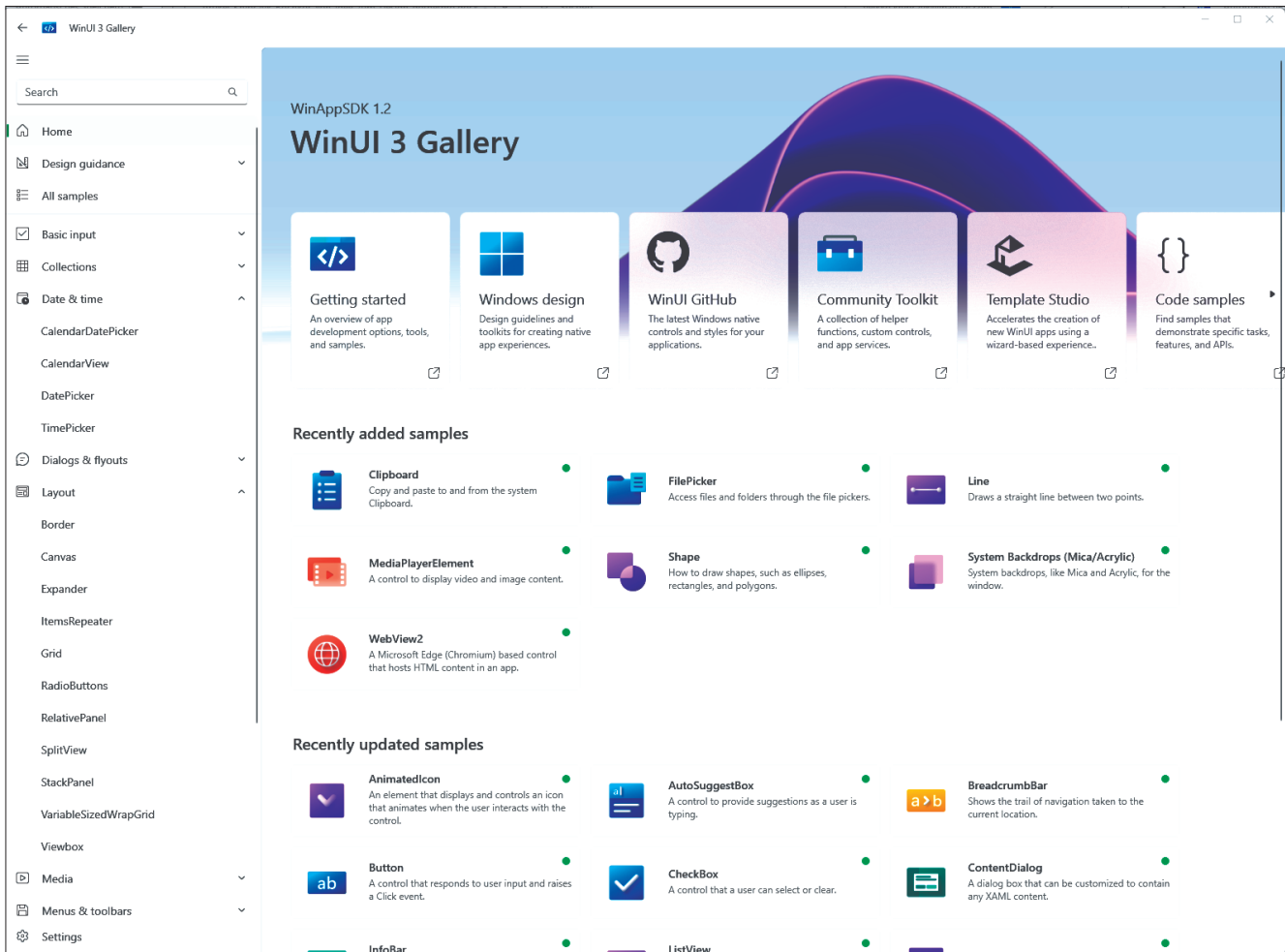
derne App kommt. Zu viele Informationen wie Designrichtlinien, Designsprache, Materialien, Konzepte et cetera strömen auf uns ein, sodass es möglich ist, dass man den Überblick verliert.

Sie könnten jetzt argumentieren, dass Ihnen das alles egal ist, und dafür zwei mögliche Gründe anführen:

- Sie sind für die App-Entwicklung und auch für das Design verantwortlich und machen es, so gut es geht. Sie versuchen, eine halbwegs „gut aussehende“ App zu gestalten, und hoffen, dass diese am Ende auch in Fragen der User Experience und des User-Interface-Designs den Ansprüchen genügt. Das Motto lautet: „Ein bisschen machen wir es wie früher und reichern es mit einigen aktuellen Elementen an.“



Übersicht über die Controls von WinUI 3 ([Bild 1](#))



Die App WinUI 3 Control Gallery (Bild 2)

- Sie bekommen einen vollständigen Entwurf (High-Fidelity-Prototyp) von einem Designer und müssen dessen Vorgaben „nur noch“ in die App-Oberfläche übersetzen. Ist der Designer vertraut mit WinUI 3, ist alles gut. Hat er Ihnen einen universellen Entwurf erstellt, dann werden an der einen oder anderen Stelle Nacharbeit, Rückfragen oder eben auch Improvisation gefragt sein.

Wie Sie es auch drehen und wenden, es wäre mehr als hilfreich, wenn Sie sich mit dem kompletten Prozess des User-Interface-Designs auseinandersetzen. Dieser startet viel früher als mit der ersten Zeile XAML für WinUI 3.

Und wenn Sie als UX-Designer unterwegs sind, kann Ihnen dann die Technik nicht egal sein oder wenigstens in den Hintergrund rücken? Lassen Sie es uns mit einer Analogie aus dem Bau eines Hauses beantworten. Sie erwerben ein Grundstück, gehen zum Architekten, äußern Ihre Vorstellungen zum neuen Gebäude, wie Zimmerverteilung, Anordnung, Dachform und so weiter und erhalten einen Entwurf. Der Architekt plant das Haus (Prototyp in der Softwareentwicklung) und zeigt Ihnen tolle 3D-Grafiken. Dann kommt die Bauphase, und auf einmal wird es schwierig. Statt eines ebenen Geländes steigt dieses in beide Richtungen an, oder man hat ein beachtliches Gefälle. Wir können jetzt ohne En-

de Erde abtragen (Entsorgungskosten) oder aufschütten (Stabilität). Mit anderen Worten: Das auf dem sprichwörtlichen digitalen Reißbrett entworfene Hauskonzept sieht zwar immer noch super aus, es passt aber nur bedingt in die Realität. Besser wäre ein sich in die Landschaft eingliedernder Entwurf gewesen, etwa mit Halbkeller und dergleichen.

Bei der modernen App-Entwicklung ist es ähnlich. Ein Entwurf der Oberfläche muss nicht nur konzeptionell passen und in Fragen des Designs überzeugen, sondern er muss auch mit der Technik harmonieren. Entwickelt man Apps für WinUI 3, dann sind eben die Designrichtlinien für Windows 11 maßgebend und als Optionen stehen die technischen Möglichkeiten von WinUI 3 zur Verfügung. Die Designer sollten das nicht nur am Rande einmal gehört haben, sondern es sollte ihnen vertraut sein.

Wir arbeiten daher im Team. App-Entwicklung inkludiert Design, und dieses geht nicht ohne ein rudimentäres Verständnis für die Technologie. In unseren Workshops legen wir deshalb auch auf beide Aspekte Wert und verbinden diese miteinander. Wir wählen einen ganzheitlichen Ansatz.

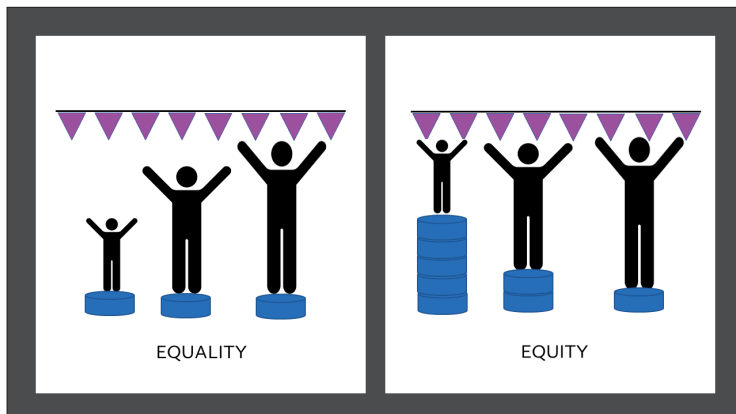
Diese Artikelserie könnte daher für Sie nutzbringend sein, wenn Sie App-Entwickler sind und die Oberfläche in Eigenregie gestalten müssen oder wenn Sie im Team arbeiten und gemeinsam (Entwickler und Designer) eine konzeptionell ►

passende, designtechnisch ansprechende und technisch saubere Lösung erarbeiten möchten.

Daher gilt: Unser Einstieg in die moderne App-Entwicklung ist nicht der Quellcode – XAML darf also noch warten –, sondern die Arbeit am Konzept.

Ausgangssituation

In der Praxis ist es oft so, dass ganz am Anfang des Entwicklungsprozesses lediglich eine grobe Vorstellung von der Grundfunktionalität der Anwendung existiert und man eine noch ziemlich ungenaue Idee davon hat, wie diese konkret aussehen soll. Es ergibt daher Sinn, systematisch vorzugehen. In einem ersten Schritt sollte man versuchen, die zugrunde liegende Problemstellung in einzelne Komponenten beziehungsweise Funktionalitäten zu zerlegen, die Zielgruppe der Nutzer zu identifizieren und zu verstehen, wer die Nutzer sind und was von der Anwendung erwartet wird.



Equity versus Equality (Bild 3)

In diesem Sinne hat UX-Design nicht nur einen großen Einfluss auf das Aussehen der Anwendung, sondern auch auf die Effizienz bei der Nutzung. Je nachdem, in welcher Situation die Anwendung eingesetzt wird, variieren die Anforderungen an die Gestaltung der Benutzeroberfläche. Das User Interface ist die Schnittstelle, an welcher der Nutzer mit der Anwendung interagiert.

Um die Potenziale und den wahren Mehrwert des UI-Designs zu erklären, stellen wir uns die Gestaltung eines Touchscreen-Displays einer Navigation vor, die im Auto eingebaut ist. Neben dem guten Aussehen soll das User Interface der Anwendung so gestaltet sein, dass der Fahrer während der Autofahrt in der Lage ist, die Anwendung ganz intuitiv zu bedienen, und dabei nicht von der Fahrt abgelenkt wird, denn bereits eine Ablenkungszeit von wenigen Sekundenbruchteilen steigert das Unfallrisiko signifikant. Hier geht es nicht nur um die Navigation und die Platzierung der Elemente. Die Wahl der Schriftart, die Schriftgröße, der Kontrast, die Entfernung des Geräts von den Augen des Fahrers, der Blickwinkel und vieles mehr spielen eine wichtige Rolle. Hinter dem UX-Design versteckt sich daher viel mehr als nur eine schöne Gestaltung der Oberfläche.

Überblick

Das User Interface ist das Erste, womit die Nutzer in Kontakt kommen. Welche Emotionen dabei aufkommen, beeinflusst daher erheblich das weitere Nutzerverhalten. Eine richtige Idee und ein gut durchdachtes Konzept sind wichtige Voraussetzungen für eine erfolgreiche designtechnische Umsetzung. Wichtig bei der Gestaltung digitaler Produkte ist daher, bei der Konzepterstellung und dem Design stets den Nutzer in den Fokus zu stellen und die Benutzererfahrung für jeden einzelnen User möglichst optimal zu gestalten.

Die Begriffe Accessibility, Universal Design und Inclusive Design sind schon lange bei der Gestaltung von User Interfaces präsent. Accessibility steht für eine bestmögliche Barrierefreiheit von Anwendungen und bezieht sich ursprünglich zum einem auf die Nutzung des World Wide Web durch Menschen mit Handicaps oder durch ältere Personen und zum anderen auf die Erreichbarkeit beziehungsweise Lesbarkeit einer Webseite durch Crawler.


Dieses spezielle Verständnis von Barrierefreiheit kann man auf die Gestaltung von Software im Allgemeinen übertragen. In Bezug auf Apps heißt es, dass man jegliche Barrieren vermeiden sollte, welche Nutzern die Verwendung des Produkts unmöglich machen könnte. Gemeint sind sowohl visuelle als auch auditive und physiologische Barrieren, welche einen Teil der User grundsätzlich oder in besonderen Fällen vom Einsatz der Software oder des Systems ausschließen.

Ein Beispiel ist die Verwendung des Navigationssystems während der Fahrt. Aus der Perspektive der Sicherheit betrachtet verbietet sich eine Bedienung über die Touch-Oberfläche. Eine Sprachsteuerung könnte ein Lösungsansatz sein, wenn diese in der Lage ist, ausreichend fehlertolerant zu reagieren und auch zielgerichtete Rückfragen zu stellen. Ein weiteres, vielleicht nicht ganz ernst zu nehmendes Beispiel ist eine Zugangskontrolle durch Spracherkennung. Diese ist äußerst praktisch, wirkt cool und man braucht keinen Schlüssel – bis man von einer größeren Zahnbehandlung zurückkehrend vor der eigenen Wohnungstür steht und gewissermaßen um Erlaubnis für den Eintritt nuschtelt (bettelt). Ein Beispiel für eine temporäre Barriere, die es ebenso wie die dauerhaften Barrieren zu vermeiden gilt. Das gelingt nur, wenn man die Situation der User gründlich erforscht.

Das Motto von Universal Design lautet: „One size fits all.“ Es wird versucht, eine Lösung zu entwerfen, die den Bedürfnissen möglichst vieler Nutzer entspricht. In der Praxis hat sich diese Vorgehensweise als schwierig herausgestellt, da die Nutzeranforderungen unterschiedlicher User-Gruppen sehr verschieden sind.

Das Motto von Universal Design lautet: „One size fits all.“ Es wird versucht, eine Lösung zu entwerfen, die den Bedürfnissen möglichst vieler Nutzer entspricht. In der Praxis hat sich diese Vorgehensweise als schwierig herausgestellt, da die Nutzeranforderungen unterschiedlicher User-Gruppen sehr verschieden sind.

Eine andere Herangehensweise bietet Inclusive Design an. Dabei versucht man eine Lösung zu finden, die unterschiedlichen Anforderungen gerecht wird. Bei den Designentscheidungen werden dabei persönliche Merkmale der User ebenso einbezogen wie die traditionell ausgegrenzten Nutzergruppen. Das Motto lautet: „Solve for one, extend for many.“ Gemeint ist, dass mit dem Ansatz des inklusiven Designs ver-



Josephine Held
Alter: 35
Beruf: Krankenschwester
Wohnort: Fulda
Familie: verheiratet, 2 Kinder

"Ich lese bei der Frühstückszubereitung gerne nebenbei Nachrichten."


Ziele

- Nachrichten Modul von dem Home Information System soll in der Lage sein die aktuellen Nachrichten aus unterschiedlichen Kategorien anzuzeigen.
- Ein integriertes Modul mit Kinderspielen

Frustrationen

- „Es wäre doch ganz schön, wenn ein Home Information System auch ein integriertes Modul mit Kinderspielen hätte, so dass die sich 5 Minuten beschäftigen könnten“

Josephine Held zusammen mit Ihrem Mann Jose und ihren 2 Kindern wohnt in Fulda. Der Mann Jose arbeitet im Vertrieb und ist oft unterwegs. Josephine arbeitet als Krankenschwester, ist zur Zeit mit ihrer Tochter Mella zu Hause. Josephine kümmert sich um die Kinder. Die Kinder Max und Mella sind 6 und 1,5 Jahre alt.



Moritz Müller
Alter: 45
Beruf: Projektleiter DB
Wohnort: Berlin
Familie: verheiratet, 3 Kinder

"Mir ist es wichtig, dass ich die Geoposition der Kinder jederzeit abrufen könnte."

Ziele

- Anzeige der Bus/Bahn Fahrpläne
- Anzeige der Geoposition
- Möglichkeit die Nachrichten und Bilder von den mobilen Geräten zu empfangen.

Frustrationen

- „Ich bin sehr nervös, wenn ich die Kinder per Handy nicht erreichen kann“
- „Ich würde gerne mit einem Klick sehen, wann die nächste Bahn kommt“

Moritz Müller wohnt mit seiner Frau und ihren 3 Kindern in Berlin Charlottenburg. Er arbeitet als Projektleiter bei der Deutschen Bahn. Seine Frau Nelle ist Beamtin bei dem Finanzamt. Die Kinder Paul (14), Agnes (12) und Bea (8) benutzen oft die öffentlichen Verkehrsmitteln, um in die Schule und zu den Freizeitaktivitäten zu gelangen. Moritz Müller ist es wichtig die Geoposition seiner Kinder jederzeit abzurufen.

Beispiele von Personas für das Home Information System (Bild 4)

sucht wird, ein Problem für einen Benutzertyp zu lösen. Die Vorteile dieser Lösung können sich auf viele andere Benutzertypen erstrecken. Es soll keiner von der Nutzung eines Produkts aus dem Grund ausgeschlossen werden, dass seine Bedürfnisse während des Designprozesses nicht ausreichend berücksichtigt wurden.

Der Ansatz des Equity-Focused-Designs geht noch einen Schritt weiter und fordert Designer auf, sich bei der Entwicklung auf die Bedürfnisse von Gruppen zu konzentrieren, die in der Vergangenheit unterrepräsentiert waren. Als gesetztes Ziel gilt, bisher ausgegrenzte Gruppen zu fördern und dadurch ein besseres Nutzererlebnis für alle Usergruppen zu schaffen. Für das Verständnis dieses Ansatzes spielen zwei Begriffe eine Schlüsselrolle: Equality und Equity.

Equality steht für Gleichberechtigung. Dabei geht es darum, gleiche Chancen und Unterstützung für alle Mitglieder der Gesellschaft zu bieten. Equity steht für Gerechtigkeit. Es wird versucht, jedem Menschen ein unterschiedliches Maß an Unterstützung zu bieten, sodass faire Ergebnisse erzielt werden können. Bild 3 verdeutlicht den Unterschied. Equity-Focused-Design setzt sich somit zum Ziel, die Anwendungen so zu gestalten, dass jeder Nutzer genau die Unterstützung bekommt, die er braucht. Dabei wird eine Gerechtigkeit der Chancen geboten. Es liegt in den Händen des UI-Designers, die unterschiedlichen Bedürfnisse der Nutzer zu erkennen und die Richtung für die Entwicklung vorzugeben.

Erst recherchieren

Die App-Entwicklung von der Idee bis zur Umsetzung ist keine triviale Sache. Hinter jeder gelungenen App steckt ein gut durchdachtes Konzept. Ein gutes Ergebnis setzt voraus, dass man die Bedürfnisse der Nutzer so gut kennt, dass man in der Lage ist, für jeden Nutzer das bestmögliche Nutzererlebnis zu bieten.

Es geht dabei um User Research beziehungsweise Benutzerforschung. Man versucht dabei, ein Verständnis der potenziellen Nutzergruppen zu entwickeln, Benutzerverhalten zu beobachten und Bedürfnisse und Motivationen aufzudecken. Der Prozess der User Research baut eine Brücke zwischen den Vorstellungen der Stakeholder (vor allem Geldgeber und Entwickler) über das Endprodukt und den Vorstellungen der Nutzer, die das Produkt tatsächlich einsetzen werden. User Research ist ein fester und fortlaufender Bestandteil des Entwicklungszyklus.

Die Benutzerforschung findet sowohl vor als auch während und nach der Produktentwicklung statt:

- **Foundation Research:** Diese wird vor der Produktentwicklung durchgeführt und beantwortet die Fragen: Was soll gebaut werden? Wie soll das Endprodukt aussehen?
- **Design Research:** Findet während der Entwicklung statt und bezieht sich beispielsweise auf Fragestellungen der folgenden Art: Wie soll gebaut werden? Wie sollen die geplanten Features umgesetzt werden?
- **Post Launch Research:** Diese Form der Forschung findet nach dem Release des Produkts statt und stellt folgende Fragen: War die Umsetzung erfolgreich? Was könnte besser gemacht werden?

Um eine passende Lösung zu entwerfen, muss man die Gefühle und Gedanken der Nutzer verstehen und in der Lage sein, sich in diese einzufühlen. Dabei versucht man, sich in die Rolle des Nutzers zu versetzen und das Produkt so zu erleben, wie der Benutzer es später verwendet. Je besser man die Bedürfnisse der Nutzer versteht und deren Wünsche antizipieren kann, desto größer ist die Chance, ein Produkt zu entwerfen, mit dem der Nutzer langfristig zufrieden ist.

Im Rahmen des Requirements Engineering versucht man, die Anforderungen an das System zu ermitteln. Die Interviews während der User Research ermöglichen es, die Nutzer besser kennenzulernen. Auch aufmerksame Beobachtungen der Nutzer während einer Interaktion mit dem Produkt sind sinnvoll. Wichtig ist dabei, möglichst frei von Vorurteilen zu bleiben. Ansonsten besteht die Gefahr, eine Anwendung zu entwickeln, die nicht den Nutzervorstellungen entspricht, sondern zu stark vom User Researcher, vom Requirements Engineer oder dem Entwickler beeinflusst wird. ▶

www.dotnetpro.de 9.2023

41

Um eine Vorstellung von den künftigen Nutzern zu gewinnen, eignet sich das Konzept der Personas. Eine Persona beschreibt eine individuelle Gruppierung von echten Personen aus der Zielgruppe des Auftraggebers. Die potenziellen Anwender werden mithilfe dieser Technik den zuvor erstellten Mustercharakteren zugeordnet. Sie geben dem oft unbekannten Nutzer im Entwicklungsprozess von Anfang an ein Gesicht und damit dem Entwicklungsteam klare Vorstellungen darüber, was diese auszeichnet. Die Erstellung von Personas verläuft sehr konkret, das heißt, dem Nutzer werden ein Name, ein Alter, ein Aussehen und meist auch Wünsche, Fähigkeiten, Meinungen und Hobbys zugeordnet. Ein Bild visualisiert die Persona und verleiht ihr ein Aussehen. Als Abbild der potenziellen Nutzer der Beispielanwendung Home Information System haben wir zwei Personas erstellt (Bild 4).

Sobald wir eine Vorstellung von den Nutzererwartungen und Anforderungen an das System haben, bekommen wir ebenfalls eine erste Vorstellung von den Inhalten, die zu der Anwendung gehören.

Struktur schaffen

Jede Anwendung beinhaltet Inhalte wie etwa Texte, Bilder, Videos, Audios oder Grafiken. Mit der Art und Weise, wie die Inhalte organisiert sind, beschäftigt sich die Information Architecture (IA). Auch Navigation und Layout spielen entscheidende Rollen. Je nachdem, wie gut es dem Entwicklungsteam gelingt, die Inhalte zu organisieren, kann der Benutzer begeistert sein und es genießen, durch die App zu navigieren. Klappt es nicht, dann ist er überfordert, verwirrt und abgeneigt von der Nutzung der App.

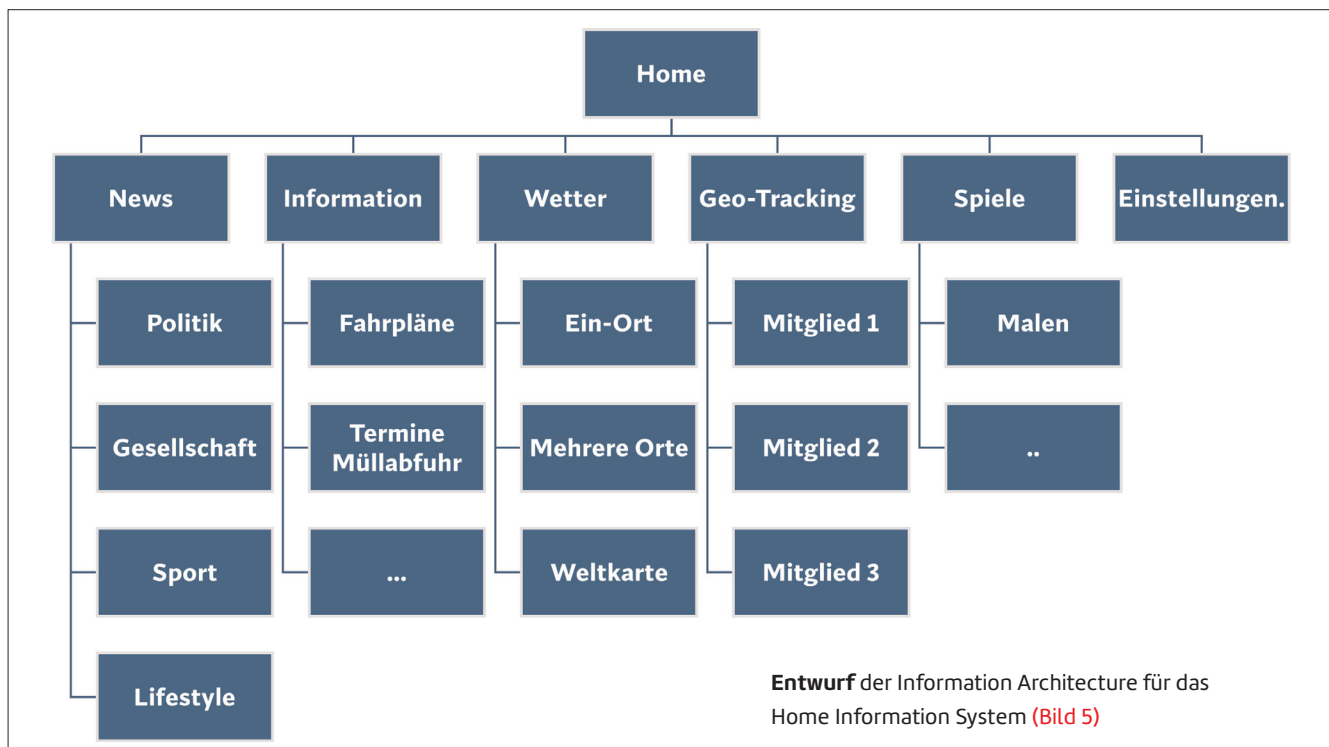
Um den langfristigen Erfolg der Anwendung sicherzustellen, ist es wichtig, sich mit der Gestaltung des Nutzererleb-

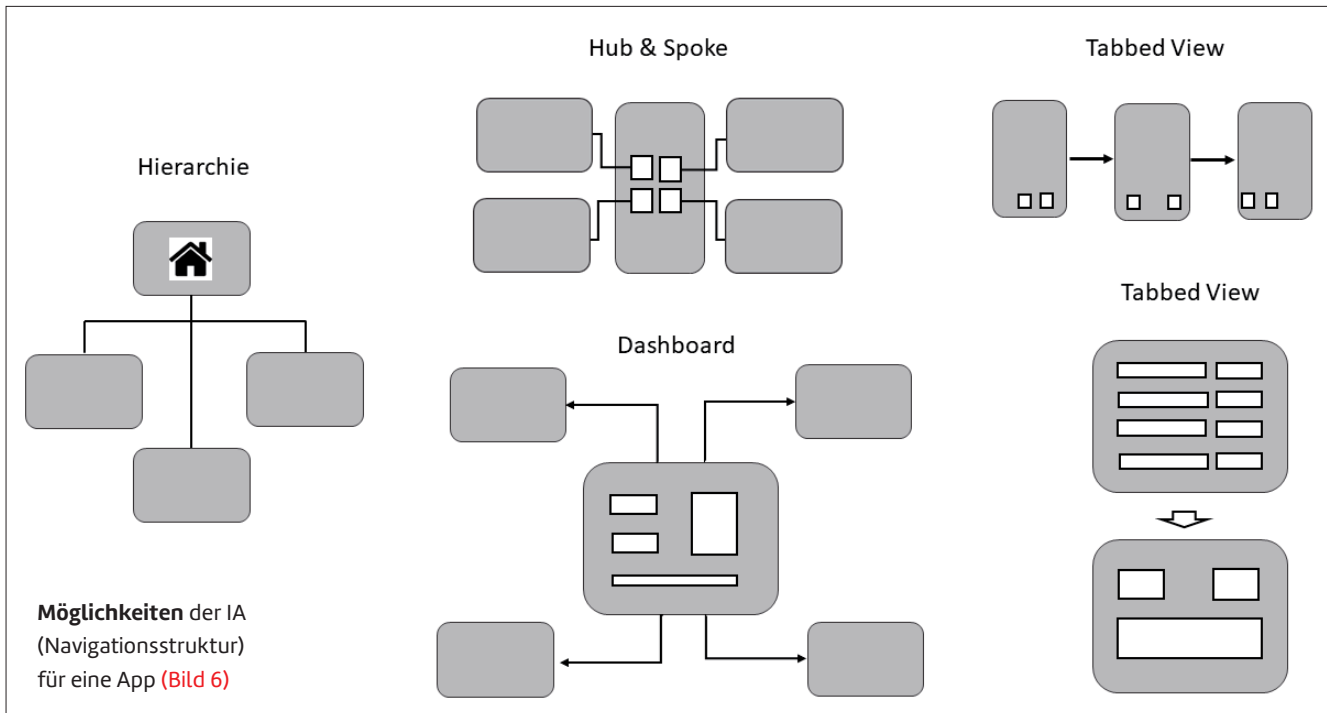
Beispielprojekt: Home Information System

In dieser Artikelserie wollen wir versuchen, den gesamten Vorgang von der Idee bis zu der Umsetzung des Designs aufzuzeigen. Dafür haben wir als Beispiel die Entwicklung eines Home Information System ausgesucht. Die Idee ist es, ein Informationssystem zu entwickeln, das in der häuslichen beziehungsweise in der privaten Umgebung auf einem Touchscreen-Display laufen soll. Es soll als Informations-, Auskunft- und Entertainmentquelle dienen, die über die aktuellen News, das kommende Wetter, den Fahrplan des örtlichen Nahverkehrs und Ähnliches informiert. Außerdem soll das System über eine Tracking-Funktion verfügen, die es möglich macht, den Standort aktiver Nutzer zu ermitteln. Diese Funktion kann vor allem für Eltern minderjähriger Kinder nützlich sein.

nisses zu beschäftigen. Man sollte sich Gedanken darüber machen, was man dazu beitragen kann, die Informationen für die User bequem und angenehm zu organisieren und somit ein bestmögliches Nutzererlebnis zu gewähren. Angestrebt wird dabei eine einfache und intuitive Gestaltung der IA. Die sollte leicht verständlich sein und die Navigation muss logisch und intuitiv sein.

Gleich vorweg: Eine Schachtelung der Navigationspunkte, wie bei den Anwendungsmenüs althergebrachter Desktop-Applikationen, kann heute nur noch professionellen IT-Experten zugemutet werden. Eine Menüstruktur wie in Visual Studio dürfte heute für die wenigsten Anwendungen, auch wenn sie auf einem großen Bildschirm laufen, passend sein.





Auch die Option zur Erweiterung der App sollte enthalten sein. Dan Brown, ein erfahrener Informationsarchitekt, hat acht Prinzipien als Grundlage für die Herangehensweise an IA formuliert [3]. Diese beziehen sich in erster Linie auf die Gestaltung von Webseiten. Sie sind jedoch auch auf andere Anwendungsarten übertragbar:

- **Prinzip der Objekte:** Die Inhalte sind lebendig, unterliegen einem eigenen Lebenszyklus und sind mit Verhaltensweisen und Attributen ausgestattet.
- **Prinzip der Wahlmöglichkeiten:** Die Nutzer sollen eine sinnvolle Auswahloption an relevanten Inhalten bekommen. Zu viele Wahlmöglichkeiten sorgen für Unentschlossenheit.
- **Prinzip der Offenlegung:** Den Nutzern soll nur das angezeigt werden, was sie brauchen, um sich weiter zu informieren beziehungsweise mit der Anwendung zu arbeiten.
- **Prinzip der Exemplare:** Für Kategorien, die nicht selbsterklärend sind, sollen Beispiele verwendet werden.
- **Prinzip der Haustüren:** Nicht alle Nutzer landen zu Beginn auf der Startseite. Man soll den Nutzern, die auf Unterseiten landen, Navigationshilfen bieten und versuchen, die Webseite von überall aus zugänglich zu machen. Übersetzt in die Welt der Apps bedeutet das: Egal wo wir uns befinden, wir brauchen stets einen Hinweis, wo wir sind und wie wir zu anderen Funktionen, beispielsweise zur Startseite, wechseln können.
- **Prinzip der Mehrfachklassifizierung:** Es sollen verschiedene Möglichkeiten angeboten werden, um die Inhalte zu durchsuchen. Die Suche und die Menüs auf der obersten Ebene sind die zwei gängigsten Möglichkeiten.
- **Prinzip der zielgerichteten Navigation:** Die Navigation soll möglichst einfach gehalten werden. Alles, was zu einer unnötigen Erhöhung der Komplexität führt, sollte vermieden werden.

- **Prinzip des Wachstums:** Eine Option des Wachstums sollte offengehalten werden.

Die Prinzipien der Information Architecture bilden die Grundlage für das Design und die effiziente Umsetzung der Anwendungen. Die IA ist ein Grundgerüst, an das die Navigation, einzelne Funktionalitäten, visuelle Elemente und alles Weitere anknüpfen.

Nach einem ausführlichen Brainstorming konnten wir den Entwurf einer Information Architecture für unser Beispielprojekt Home Information System erstellen, gezeigt in [Bild 5](#).

Es gibt grundsätzliche Unterschiede in der IA von mobilen und Desktop-Anwendungen. Der Hauptgrund dafür: Die Verwendung eines Smartphones unterscheidet sich qualitativ von der Verwendung eines PC. Die Benutzerinteraktion mit dem Smartphone basiert auf dem Wischen und Tippen statt auf Klicken und Verwendung von Tastaturkürzeln. Auch die Größe des Bildschirms spielt eine Rolle.

In diesem Artikel beschäftigen wir uns mit der IA-Gestaltung von Apps, die mit WinUI 3 entwickelt werden. Diese Art von Apps laufen unter Windows, also zum Beispiel auf einem Arbeitsplatzrechner (Desktop) oder wie unser Smart Home System auf einem Mini-PC mit einem kleineren Bildschirm. Dieser nutzt die Touch-Eingabe statt der Interaktion mit Maus und Tastatur. Wir stellen fest: Die Konzepte vermischen sich; WinUI 3 unterstützt auch diese Gerätetypen, indem es statt Ereignissen für Mausinteraktionen (spezialisiert) mehrere Arten von Ereignissen, also Maus, Tastatur, Touch, Stift und so weiter, unterstützt.

Kommen wir zurück zur Information Architecture. Es gibt zahlreiche IA-Patterns, die eine Hilfe bei der Gestaltung der App sein können. Nachfolgend geben wir einen kompakten Überblick (siehe auch [Bild 6](#)) auf der Basis von [4]: ▶

- **Hierarchie:** Dieses IA-Muster findet seine Anwendung hauptsächlich bei der Gestaltung von Webseiten. Es eignet sich für Apps, die aus einzelnen Seiten bestehen. Auch moderne WinUI-3-Apps können nach diesem Muster aufgebaut werden. Jede Seite stellt die Oberfläche für ein Thema beziehungsweise ein Problem dar. Diese Seiten ordnen sich dabei in eine Hierarchie ein. Problematisch kann es jedoch werden, wenn durch die Verwendung des Hierarchie-Patterns eine zu tiefe Navigationsstruktur entsteht. Es gilt: Eine App ist kein Schrank aus Fächern, Schubladen, Kästen und dergleichen.
- **Hub:** Es gibt eine Indexseite, den sogenannten Hub, mit Spokes, zu denen man navigieren kann. Um zu einem anderen Spoke zu wechseln, müssen Benutzer zunächst zum Hub zurückkehren. Dieser Ansatz eignet sich gut für multifunktionale Apps, bei denen jedes Tool oder Feature seinen eigenen Zweck und seine eigene interne Navigation hat.
- **Tabbed View:** Dieses Muster der Informationsarchitektur ist wohl jedem Benutzer vertraut. Der Inhalt ist in verschiedenen Tabs platziert und Benutzer können mithilfe der Tabs zwischen ihnen wechseln. In WinUI-3-Apps werden die Tabs am oberen Bildschirmrand angezeigt. Dieses IA (Navigationsmuster) ist auch auf mobilen Geräten weit verbreitet. Die Nutzer kommen direkt damit zurecht. Die Tabbed View wird üblicherweise auf oberster Ebene verwendet, sie kann jedoch auch mit anderen Navigationsmustern kombiniert werden. So könnten wir die Tabbed View auf Ebene zwei einsetzen, um ein Hauptthema weiter aufzusplitten. Ein Beispiel: Die Hauptnavigation ist ein Dashboard (zentrale Startseite), ein Menüpunkt darin führt zu den App-Ein-

stellungen, und diese werden dann in Sektionen mithilfe von Registerkarten systematisiert (Tabbed View).

- **Dashboard:** Dieses IA-Pattern besteht aus verschiedenen Elementen (Teileseiten, Details), die einen Teil der Informationen auf der Indexseite anzeigen. Der Vorteil besteht darin, dass es den Nutzern ermöglicht, die wichtigsten Informationen auf einen Blick zu konsumieren, zu analysieren und zu priorisieren. Es eignet sich für Apps, die inhaltsbasiert und multifunktional sind.
- **Filtered View:** Das Pattern ermöglicht den Benutzern, zwischen alternativen Ansichten zu wechseln, indem die angezeigten Inhalte gefiltert werden können. Eine Filtered View eignet sich für Apps, die darauf angelegt sind, große Mengen an Inhalten anzuzeigen und den Benutzern die Freiheit zu geben, individuell zu entscheiden, wie sie den Inhalt erkunden möchten.

Die oben dargestellten Patterns können beliebig kombiniert werden. Wichtig ist jedoch, ein übergeordnetes Muster auszusuchen und die daraus folgenden Entscheidungen in Fragen der IA daran auszurichten.

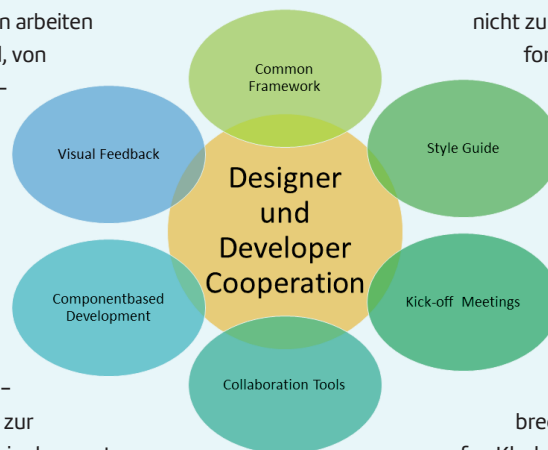
Konzept mit Wireframes und Prototypen umsetzen

Mit dem Erstellen von Wireframes kann man Konzepte visualisieren. Es handelt sich dabei um eine vereinfachte Darstellung, wie eine Anwendung aussehen wird, wenn diese fertig ist. Es ist ein erster schematischer Entwurf, der die Nutzeroberfläche unter Vernachlässigung der konkreten Texte und Bilder zeigt.

Developer und Designer – gemeinsam zur besseren Nutzererfahrung

Bei der Entwicklung von Anwendungen arbeiten Designer und Entwickler Hand in Hand, von der Ideenfindung über Mockups, Wireframing und Prototyping bis hin zu der funktionierenden Anwendung (Bild 7). Ausschlaggebend für eine gute Zusammenarbeit sind funktionsübergreifendes Wissen und ein dynamischer Ansatz. Ein gemeinsam angestrebtes Ziel ist es, eine qualitativ hochwertige und funktionierende Nutzererfahrung zu erstellen. Von einer gemeinsamen Strategie zur Entwicklung, einem gemeinsamen Designkonzept und einem gemeinsamen Umsetzungsplan können sowohl Designer als auch Entwickler profitieren.

In der Praxis sind leider nach wie vor einige Vorurteile anzutreffen, dergestalt, dass es beim Design um reine Ästhetik gehe, während bei der Entwicklung nur die Funktionalität, die Benutzerfreundlichkeit und die Skalierbarkeit realisiert würden. Tatsächlich greifen die beiden Bereiche ineinander und sind



Developer versus Designer (Bild 7)

nicht zu trennen. Es gibt zahlreiche Herausforderungen und Hindernisse bei der Gestaltung der Zusammenarbeit zwischen Designern und Entwicklern, welche durch gezielte Maßnahmen vermieden beziehungsweise gemildert werden können. Dazu gehören beispielsweise regelmäßige Kick-off-Meetings unter Einbeziehung von Designern und Entwicklern sowie auch des Kunden. Solche Veranstaltungen wirken wie Eisbrecher, stoßen den Dialog an und schaffen Klarheit. Kick-off-Meetings sind vor allem hilfreich, um Vorstellungen über das Endprodukt auszutauschen, Herausforderungen zu kommunizieren und gute Lösungen abzuleiten. Häufiges Testen und gegenseitiges Feedback tragen dazu bei, die Zusammenarbeit optimal zu gestalten. Beispielsweise sollten die entwickelten Skizzen und Prototypen regelmäßig auf ihre Umsetzbarkeit hin überprüft werden.

Die Wireframes konzentrieren sich auf die Struktur und die Darstellung der Benutzerführung. Anhand von Wireframes bekommt man einen Überblick über die Potenziale und die möglichen Einschränkungen des gewählten Konzepts. Für die Stakeholder ist es eine erste gute Gelegenheit, sich eine Vorstellung von der Umsetzung des Projekts zu verschaffen. Ohne Inhalte und Details werden das Layout, die Navigation und die Funktionalität aufgezeigt.

Man unterscheidet zwischen Low-Fidelity-Wireframes und High-Fidelity-Wireframes. Erstere sind einfache Entwürfe, die sich auf die Struktur und die Anordnung der Elemente konzentrieren. Diese sind schnell und einfach zu erstellen. Die High-Fidelity-Wireframes sind dagegen aufwendiger und werden eingesetzt, um eine technisch detailliertere und optisch ausgereifere Darstellung zu präsentieren. Im Gegensatz zu den statischen Wireframes sind Prototypen interaktiv.

Fazit und Ausblick

Wir sind am Ende des ersten Teils der Artikelserie angekommen. Unser Workout geht weiter beziehungsweise erst richtig los. Wir werden genau an dieser Stelle weitermachen und mit Wireframes und Prototypen starten und uns dann mit dem Layout der App beschäftigen. Hier haben wir mit WinUI 3 auch eine Menge an Optionen, um die allgemeinen Konzepte in die Praxis umzusetzen. Spätestens wenn das Layout in der App realisiert werden soll, brauchen wir etwas Quellcode – in diesem Fall in XAML. ■

[1] Microsoft Learn, Controls for Windows apps, www.dotnetpro.de/SL2309WinUI3_1

[2] Microsoft Store, WinUI 3 Control Gallery (App), www.dotnetpro.de/SL2309WinUI3_2

[3] Dan Brown, Eight Principles of Information Architecture, www.dotnetpro.de/SL2309WinUI3_3

[4] Applikey, Designing The Information Architecture (IA) of Mobile Apps, www.dotnetpro.de/SL2309WinUI3_4



Elena Bochkor

arbeitet am Entwurf und Design mobiler Apps und Webseiten. Von der systematischen Nutzerforschung über visuelle Prototypen bis hin zu einem barrierefreien Design gestaltet sie die User Experience moderner digitaler Produkte. <https://wp.larinet.com>



Dr. Veikko Krypczyk

ist Softwareentwickler, Trainer und Fachautor und u. a. auf die Themen WinUI 3 und .NET MAUI spezialisiert. Sein Wissen gibt er über Fachartikel, Seminare und Workshops weiter und er unterstützt auch individuelle Projekte. <https://wp.larinet.com>

dnpCode

A2309WinUI3

DEV ACADEMY

HANDS-ON-WORKSHOPS UND
WEITERBILDUNG FÜR SOFTWARE-
ENTWICKLER UND -ARCHITEKTEN



3 TAGE

ASP.NET Blazor - SPA-Anwendungen mit C# und .NET

TRAINING!

Was wird behandelt

- Blazor-Grundlagen
- Routing und Binding mit Blazor
- Formularanbindung mit Blazor
- Validierung von Daten mit Blazor
- Datenbereitstellung mit REST und Integration in Blazor

Christian Giesswein
Trainer

