



Rapid Development mit Wisej 2.0

# Umbauarbeiten

Veikko Krypczyk

Webanwendungen entwickeln ohne tiefe Kenntnisse von HTML5, CSS3 oder JavaScript – mit dem Framework Wisej soll das möglich sein. Auch die schnelle Migration von Legacy-Anwendungen aus dem .NET-Universum ins Web verspricht der Hersteller.

Webapplikationen laufen plattformunabhängig und ohne Installation in jedem Browser und auf jedem Gerät. Unzählige Techniken, Frameworks und Bibliotheken buhlen um die Gunst des Entwicklers. Trotzdem bleibt es oft kompliziert. Das Framework Wisej der amerikanischen Ice Tea Group verspricht eine effektive Entwicklung von Business-Webanwendungen ohne den mühsamen Einsatz von HTML5, CSS3 und JavaScript. Stattdessen gibt es einen komfortablen Designer, und den Code schreibt man in C# oder Visual Basic .NET. Ganz frisch ist die Release 2.0.

Bei einer neuen Businessapplikation stellt sich zunächst die Frage nach dem Anwendungstyp. Konkurrierend stehen sich die klassische Desktop- und die Webanwendung gegenüber. Beide haben Vor-

und Nachteile. Das Ende des Desktops wurde schon oft vorhergesagt. Dennoch werden immer noch neue Projekte auf dieser technischen Basis gestartet. Die Gründe dafür sind vielfältig, zum Beispiel:

- vollständiger Systemzugriff mit einem uneingeschränkten Zugriff auf die Hardware des Rechners;
- hohe Performance, da die ausführbaren Programme auf dem lokalen Rechner liegen;
- systemkompatible Gestaltung der Benutzerschnittstelle;
- etablierter und effektiver Entwicklungsprozess mit ausgereiften Tools und Frameworks;
- ein riesiger Fundus an Altapplikationen und damit wiederverwendbaren Quellcodes.

Dem gegenüber stehen die Webapplikationen. Ihre Vorzüge:

- keine Softwareinstallation auf dem Rechner, ein aktueller Browser genügt in der Regel;
- plattformübergreifende und geräteunabhängige Ausführbarkeit;
- Mehrbenutzerfähigkeit.

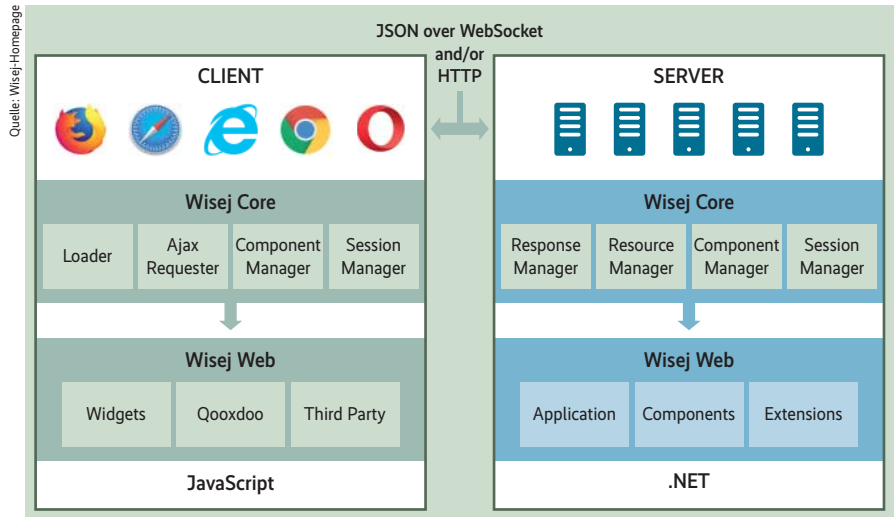
Der Trend geht seit einiger Zeit in Richtung der deutlich flexibleren Webapplikationen. Bei der Entwicklung einer solchen Anwendung stehen Unternehmen oft vor folgender Situation: Viele Entwicklungsprojekte starten nicht auf der sprichwörtlichen grünen Wiese. Meist gilt es, Altanwendungen auf den aktuellsten Stand zu bringen. Diese Applikationen laufen seit Jahren, dennoch wird die Unausweichlichkeit einer umfassenden Sanierung immer deutlicher.

## Aus alt mach neu

Notwendige Programmanpassungen und Fehlerkorrekturen lassen sich kaum noch umsetzen. Die verwendeten Programmiersprachen und Frameworks sind oft veraltet. Nicht selten findet man darunter noch Anwendungen, die mit Visual Basic 6 (VB6), also noch vor der .NET-Ära erstellt wurden. Die Benutzeroberfläche sieht nicht mehr zeitgemäß aus. Technisch besser, aber oft mit einem ebenso großen Modernisierungsbedarf verbunden sind Windows-Forms-Anwendungen, klassische Desktop-Applikationen auf der Basis von .NET.

Häufig möchte ein Unternehmen diese Anwendungen fit fürs Web machen. Gegen eine vollständige Neuimplementierung können folgende Gründe sprechen: Vorhandener Quellcode lässt sich nicht oder nur mit viel Aufwand wiederverwenden. Nicht nur die Benutzerschnittstelle, sondern auch die komplette Fach- und Datenhaltungsschicht sind meist komplett neu zu implementieren. Zudem ist der Lernaufwand bei einem Wechsel der eingesetzten Werkzeuge vom Desktop zum Web hoch. Programmierer müssen komplett andere Techniken beherrschen. Andererseits ist ihre Kenntnis der Fachdomäne unverzichtbar und müsste von einem anderen Entwicklungsteam erst mühsam erworben werden.

Einen Ausweg aus diesem Dilemma will die Ice Tea Group mit ihrem Framework Wisej bieten. Bisher lag dessen Fokus eindeutig auf der Modernisierung von Legacy-Anwendungen, mit der Version 2.0 empfiehlt der Hersteller nun auch den Einsatz in vollständig neuen Projekten. Als Vorteil hebt er die besondere Effektivität des Entwicklungszyklus hervor.



**Auf der Basis von JavaScript bilden Wisej-Widgets die Serverkomponenten auf dem Client ab (Abb. 1).**

Wisej wählt im Vergleich zu Konkurrenzprodukten einen vollständig anderen Ansatz für das Erstellen einer Webapplikation. Auf seiner Website verspricht der Hersteller, dass sich mit dem Framework dynamische Webapplikationen auf die gleiche Weise wie Windows-Forms-Anwendungen erstellen lassen. Der Fokus liegt klar auf dem Businesskontext.

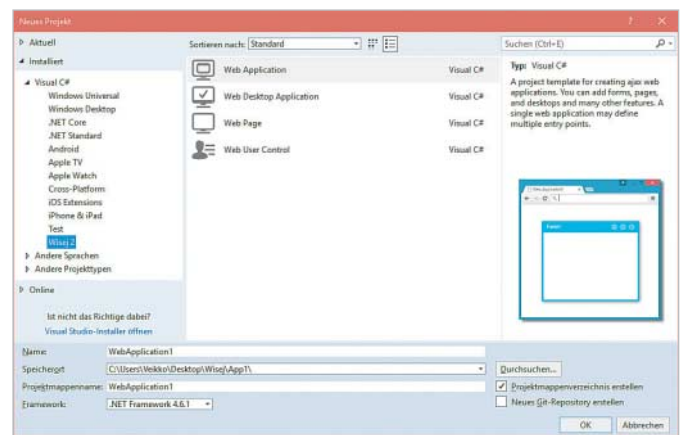
Bei der Arbeit mit Wisej kann ein Entwickler schnell vergessen, dass er eine Webapplikation baut. Er muss sich keine Gedanken über das Zusammenspiel von HTML5, CSS3 und JavaScript machen. Das alles passiert unter der Haube und er kann sich voll auf die Umsetzung seines Geschäftsmodells konzentrieren. Den Datentransfer zwischen Client und Server übernimmt Wisej ebenfalls.

## Migration von .NET zu JavaScript

Abbildung 1 zeigt die Architektur einer Wisej-Applikation, das heißt das Zusammenspiel von Client und Server. Auf der Serverseite werden Steuerelemente wie Buttons oder Textboxen verwendet. Diese Komponenten basieren auf .NET und werden zur Designzeit erstellt. Die große Auswahl an Standardkomponenten kann der Entwickler durch Extensions erweitern. Diese Vorgehensweise erlaubt es, den gleichen Entwicklungszyklus wie bei Windows-Forms-Anwendungen nachzubilden. Der Entwickler kann Komponenten für die Benutzerschnittstelle aus einer Palette auswählen, sie mithilfe eines grafischen Designers platzieren und die Programmlogik über Events verknüpfen. Da als Ergebnis jedoch keine Desktop-, sondern eine Webapplikation generiert wird und der Browser nur mit HTML5, CSS3 und JavaScript umgehen kann, muss am Ende eine Transformation erfolgen. Das erledigt das Framework im Hintergrund.

Wisej-Widgets bilden die serverseitig verwendeten Komponenten für die Benutzerschnittstelle eins zu eins ab. Es handelt sich um JavaScript-Klassen auf der Grundlage der Bibliothek Qooxdoo, die Wisej

**Auf Wunsch konfiguriert Wisej bei der Installation direkt ein bestehendes Visual Studio und bringt gleich neue Projektvorlagen mit (Abb. 2).**



erweitert. Ein JavaScript-Widget dient als Rendering-Engine für die grafische Oberfläche. An dieser Stelle hat der Programmierer die Möglichkeit, Komponenten anderer Anbieter auf der Basis von JavaScript einzubinden. Diese beiden Ebenen, das heißt die Client- und die Serverseite, bilden die statische Struktur, also den Aufbau der Applikation.

Die Interaktion zwischen Client und Server basiert auf den folgenden Prinzipien: Vereinfacht gesagt erhält die serverseitige Middleware Ajax-Anfragen vom Client im JSON-Format, sendet diese Requests als .NET-Events an die serverseitigen Komponenten, sammelt die Änderungen an der Benutzerschnittstelle und schickt die Unterschiede als JSON-Paket

zurück an den Client. Die Antwort verarbeitet die Middleware auf Clientseite weiter.

## Interaktion zwischen Client und Server

Auf dieselbe Weise sammeln Widgets Event-Anfragen und übermitteln sie an den Server. Aufseiten des Servers programmiert man mit dem .NET-Framework. Damit kann eine Wisej-Anwendung die Geschäftslogik und Datenbankwelt von .NET nutzen. Es entsteht zwar letztlich eine klassische Webapplikation, aber mit der technischen Struktur des Webs kommt der Entwickler nicht oder nur wenig in Berührung.

Wisej verfolgt das Single-Page-Konzept. Es lädt eine HTML-Seite und manipuliert dann die einzelnen Objekte, um den geänderten Anwendungsinhalt anzuzeigen. Intern arbeiten ein schneller .NET-JSON-Serializer und Parser auf Serverseite, die Clientseite nutzt den JavaScript-JSON-Serializer und den Parser, die standardmäßig eingesetzt werden.

Als integrierte Entwicklungsumgebung kommt Visual Studio zum Einsatz.



- Das Framework Wisej hilft bei der Entwicklung von Webanwendungen, die in C# oder VB.NET geschrieben werden.
- In der Client-Server-Architektur des Frameworks bilden Wisej-Widgets die Serverkomponenten auf dem Client durch JavaScript-Klassen ab.
- Wisej eignet sich sowohl für Neuentwicklungen unter Visual Studio als auch für die Migration von Legacy-Anwendungen vom Desktop ins Web.

Wisj 2.0 arbeitet sowohl mit der aktuellen Version 2017 als auch mit älteren Releases. Laut Hersteller soll Wisj mit der kommenden Version von Visual Studio ebenfalls kompatibel sein. Für die Arbeit mit der Testversion, die man von der Wisj-Homepage herunterladen kann [1], genügt die kostenlose Community Edition des Visual Studio. Die Installation des Frameworks ist schnell erledigt, das Setup ist nur wenige MByte groß. Nach der Installation stehen die neuen Entwicklungsoptionen direkt in Visual Studio zur Auswahl (siehe Abbildung 2).

Nach dem Start von Visual Studio wählt der Entwickler eine der installierten Vorlagen von Wisj, danach gelangt er direkt in den grafischen Designer. Über die Toolpalette kann er in gewohnter Weise die Benutzerschnittstelle gestalten und für jedes Objekt die passenden Eigenschaften einstellen. Wisj hat die typischen Eigenschaften der Controls um spezifische Attribute für die Webprogrammierung ergänzt, zum Beispiel *CssStyle* oder *CssClass*.

Die Toolpalette enthält alle wichtigen Controls, mit denen man eine grafische Benutzeroberfläche für eine Businessapplikation erstellen kann. Darunter finden sich Basis-Controls wie Texteingabefelder, Labels und Buttons sowie komplexere Controls wie ein DataGridView zum Darstellen von Daten in Tabellenform. Neben den visuellen Controls gibt es nichtvisuelle Steuerelemente, etwa einen OpenFileDialog, mit dem man häufige Anforderungen kapseln kann, statt sie eigenständig zu programmieren.

Für Migrationsprojekte aus Windows Forms enthält Wisj für fast alle Steuerelemente adäquate Controls. Zusätzliche

moderne Controls bieten spezielle Funktionen für Webapplikationen an; dazu zählen der TinyEditor für ein Texteingabefeld mit Formatierungsanweisungen oder ein Steuerelement zur Darstellung einer Google-Maps-Karte, zu der man direkt den API-Key hinterlegen kann.

## Die Konfiguration im Hintergrund

Nach dem Start der Webapplikation über Visual Studio erscheint sie direkt im Standardbrowser. Da es sich um eine dynamische Anwendung handelt, muss man im Hintergrund einen Server starten. Das übernimmt Visual Studio inklusive sämtlicher Konfigurationsarbeiten für den Internet Information Server (IIS-Express). Dieser lauscht unter der üblichen lokalen Adresse localhost und unter dem in den Projektoptionen vorgegebenen Port.

Als Test für diesen Artikel soll eine kleinere Windows-Forms-Anwendung in eine Webapplikation portiert werden. (Die Quellen dazu sind über den iX-Listing-Service erhältlich.) Inhaltlich geht es um die Berechnung von Kreditdaten. Nach dem Eingeben der Werte für den Kreditbetrag, den Zinssatz, den Tilgungssatz und die Laufzeit soll die Anwendung den Tilgungsplan ermitteln und anzeigen (Abbildung 3). Die Programmlogik liegt in Form mehrerer C#-Klassen vor. Für die Webapplikation wird ein neues Wisj-Projekt angelegt und als Projekttyp Web Page ausgewählt. Mit den Controls TextBox, Label, Button und DataGridView wird die Oberfläche im Designer erstellt.

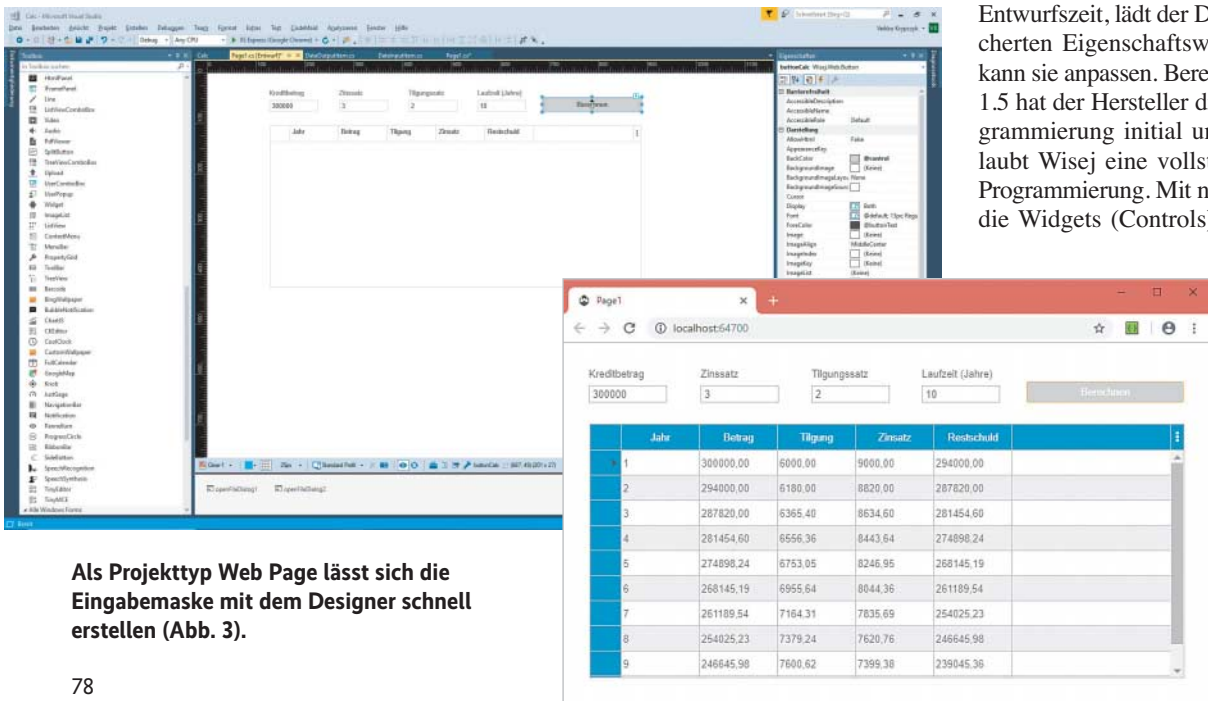
Die in den C#-Dateien vorliegende Programmlogik kann man ohne Anpassung der Klassen in das neue Projekt übernehmen. Auch die Verknüpfung des GUI mit der Programmlogik erfolgt wie in einem Windows-Forms-Projekt (Code-behind). Das heißt, man bindet die Ereignisse an die Steuerelemente, etwa das Click-Ereignis an einen Button. Weitere Anpassungen am Quellcode waren für eine Migration von einem Windows-Forms- (Desktop-Applikation) zu einem Wisj-Projekt (Webapplikation) nicht notwendig. Nach dem Starten des fertigen Projekts funktioniert die Webapplikation im Browser fehlerfrei (Abbildung 4).

Der Hersteller bezeichnet Wisj selbst als Migrationswerkzeug für Windows-Forms-Applikationen. Oft gelingt die Umstellung in wenigen Schritten, einige Hinweise findet man im Kasten: „Vom Desktop zur Webapplikation“.

## Neuerungen der Release 2.0

Im Februar 2019 hat Wisj mit der Release 2 ein umfassendes Update erfahren. Der Hersteller hat wichtige Funktionen überarbeitet, Fehler bereinigt und das Framework technisch auf den neuesten Stand gebracht. Gegenüber den Vorversionen 1.\* gibt es darüber hinaus noch einige Erweiterungen.

Im Designer erleichtern Hilfslinien Positionierung und Ausrichtung der Objekte. Man kann jetzt mehrere Profile für die Gestaltung der Benutzeroberfläche anlegen. Damit ist die Umsetzung eines responsiven Designs für unterschiedliche Gerätearten (Desktop, Smartphone, Tablet) möglich. Wechselt man das aktive Profil zur Entwurfszeit, lädt der Designer die gespeicherten Eigenschaftswerte neu und man kann sie anpassen. Bereits mit dem Update 1.5 hat der Hersteller die asynchrone Programmierung initial unterstützt, jetzt erlaubt Wisj eine vollständig asynchrone Programmierung. Mit neuen Properties für die Widgets (Controls) – unter anderem



Als Projekttyp Web Page lässt sich die Eingabemaske mit dem Designer schnell erstellen (Abb. 3).

Die angegebene Programmlogik in den C#-Klassen übernimmt die Berechnung des Tilgungsplans (Abb. 4).



# Vom Desktop zur Webapplikation

Die Ice Tea Group beschreibt auf ihrer Website die einzelnen Schritte bei der Migration einer Desktop-Anwendung zu einer Webapplikation mit Wisej nach Muster:

**1. Analyse der Legacy-Applikation:** Die große Herausforderung besteht in einer Transformation der Benutzerschnittstelle. Sofern man nur Standard-Controls verwendet, dürfte sich ein passendes Control in Wisej finden. Schwieriger wird es, wenn man auf Komponenten von Drittanbietern gesetzt hat. Hier muss man gegebenenfalls das Interface mit anderen Controls überarbeiten.

**2. Wisej-Projekt anlegen:** Es gibt zwei Optionen. Man kann das bisherige Projekt zu einer Wisej-Anwendung umbauen oder ein neues Projekt anlegen und die Inhalte integrieren. Letztendlich läuft es auf dasselbe hinaus. Dennoch scheint der Weg über eine neue Projektvorlage etwas strukturierter, da der Entwickler dabei intensiver in das Altprojekt „eintaucht“ und die zu migrierenden Inhalte begutachtet. Das Anlegen eines neuen Wisej-Projekts erfolgt über den Assistenten. Die Bezeichnungen der Namespaces im neuen Projekt sollten denen des Migrationsprojekts entsprechen.

**3. Dateien hinzufügen:** Die relevanten Dateien werden in das neue Projekt kopiert. Dabei sollte man die ursprüngliche Projektstruktur beibehalten. Es gilt, nicht zu viele Änderungen auf einmal vorzunehmen. Wer die Struktur des Projektes etwas anpassen möchte, muss das entweder noch im Legacy-Projekt tun oder nach erfolgreicher Migration. Sind die Ordner und Dateien in das neue Projekt kopiert, werden diese Elemente über den Projektmappen-Explorer hinzugefügt.

**4. Bereinigen der uses-Anweisungen:** In diesem Schritt ersetzt man im gesamten Projekt die Verweise *Windows.Forms* durch *Wisej.Web*.

**5. Clean und Rebuild:** Zum Schluss bereinigt man das gesamte Projekt und startet einen Rebuild. Einige manuelle Anpassungen – je nach Umfang und Komplexität des Projekts und der eingesetzten Controls – sind sicherlich notwendig. Man kann die „fehlerhaften“ Stellen im Code zunächst auskommentieren und sie später überarbeiten.

**6. Start:** Wenn der Compiler keine Fehler mehr feststellt, kann man die Webanwendung starten. Im Idealfall erscheint das Programm im Browser.

*CssClass* und *CssStyle* – lassen sich individuelle Designs umsetzen.

Wisej-Anwendungen lassen sich im Design leicht anpassen und man kann mit einem Mausklick zwischen Darstellungen mit unterschiedlichen Themes wechseln. Solche Änderungen sind bereits während der Entwurfszeit sichtbar. Zur Auswahl stehen vordefinierte Themes und der Entwickler hat außerdem die Möglichkeit, die Farbauswahl für die einzelnen Elemente jeweils individuell anzupassen und bei Bedarf ein eigenes Theme zu erstellen. Auch das lässt sich dialogorientiert erledigen. Wisej stellt dazu den Theme Builder zur Verfügung und generiert den zugehörigen Code (CSS) automatisch.

Über Extensions lässt sich das Funktionsspektrum von Wisej leicht erweitern. Mithilfe von Geolocation ist es beispielsweise relativ einfach, einen Service für die Ortsbestimmung zu nutzen. ChartJS bietet die Möglichkeit, Businessdiagramme wie Balken oder Linie leicht einzubauen. Mit Speech kann man Funktionen zur Sprachausgabe integrieren und Translation hilft bei der automatischen Textübersetzung auf der Basis von Yandex, Google oder Bing.

Derzeit unterstützt Wisej direkt nur den Internet Information Server unter Windows. Man kann aber eine mit dem Framework entwickelte Anwendung rela-

tiv flexibel über die Cloud bereitstellen – etwa über Amazon AWS oder Microsoft Azure. Auf Windows-Rechnern ohne IIS lassen sich Wisej-Webanwendungen ebenfalls ausführen und über die Wisej.Host-Service-Extension können sie auch in der Docker-Umgebung eines Linux-Rechners mit einem Apache-Server laufen.

Für Wisej-Anwendungen ohne Server stellt der Hersteller die Datei *Wisej.Application.exe* zum Download als Projekt inklusive Quellcode zur Verfügung. Kopiert man diese Dateien in den betreffenden Projektordner und ruft das Programm auf, erscheint ein Startfenster und die Wisej-Anwendung wird ausgeführt und lädt im Hintergrund die benötigten Dateien.

## Fazit

Weitere Anpassungen sind in den meisten Fällen nicht notwendig. Technisch gesehen wird dabei kein Webserver auf dem Rechner ausgeführt, sondern mithilfe von OWIN (Open Web Interface for .NET) und Katana die clientseitige Webanwendung direkt auf dem lokalen .NET-Framework gestartet. Diese Optionen sind gerade für Übergangsszenarien bei einer laufenden Migration im Enterprise-Bereich interessant.

Mit Wisej lassen sich Webapplikationen ohne tief gehende Kenntnisse der Webtechniken entwickeln. Für in die Jahre gekommene Altanwendungen, zum Beispiel auf der Basis von Windows Forms, Visual Basic oder sogar VB6, scheint dies ein gangbarer Weg zu sein, vorhandenen Quellcode zu recyceln. Die Neuimplementierung solcher Legacy-Anwendungen – etwa in JavaScript auf der Basis eines Frameworks – würde zunächst keine zusätzlichen Funktionen produzieren, aber dennoch viel Arbeit bereiten, was viel Zeit und letztlich Geld kostet. Jeder Algorithmus und jede Funktion, die man weitgehend mit Copy-and-Paste übernehmen kann, spart wertvolle Entwicklerstunden. Für ausgewählte Migrationsszenarien ist Wisej daher ein interessantes Werkzeug.

Wer eine komplett neue Applikation entwickeln will, sollte sich überlegen, welchen Stellenwert sie haben wird. Handelt es sich um ein einmaliges Projekt oder wird es zukünftig einen größeren Bedarf an modernen Webapplikationen geben? Welches Wissen bringen die Entwickler mit, sind sie fit in den nötigen Webtechniken oder müssten sie viel dazulernen? Wisej könnte auch in diesem Fall ein hilfreiches Werkzeug sein, das einen anderen Entwicklungsansatz verfolgt als andere Frameworks zur Webentwicklung.

Entscheidend ist in beiden Fällen, ob .NET weiterhin die technische Basis künftiger Anwendungsentwicklungen bilden soll. (ka@ix.de)

**Dr. Veikko Krypczyk**

ist Softwareentwickler und IT-Fachautor.

## Quellen

[1] Alle URLs: [ix.de/ix1904076](http://ix.de/ix1904076)



## IX-Wertung

- ⊕ Webanwendungen ohne HTML5, CSS3 und JavaScript
- ⊕ Migration von Legacy-Anwendungen
- ⊕ schnelles Prototyping
- ⊕ Erweiterbarkeit durch Extensions
- ⊖ nur für IIS
- ⊖ Abhängigkeit vom .NET-Framework

## Daten und Preise

### Wisej 2.0

**Anbieter:** Ice Tea Group

#### Preise:

- Developer Edition: 690 US-Dollar (mit Serverlizenz)
- Server Edition: 390 US-Dollar
- Testlizenz (15 Tage): kostenlos