

## DESKTOP- IN WEBANWENDUNGEN UMWANDELN

# Alter Wein in neuen Schläuchen?

Das Framework Wisej lässt Webanwendungen wie Desktop-Applikationen aussehen.

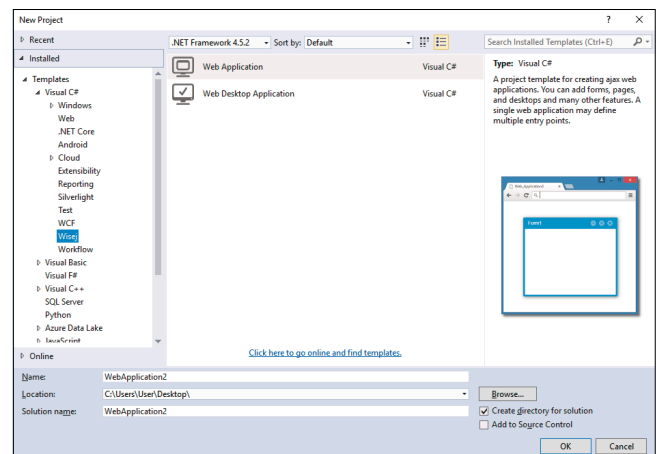
Geschäftsanwendungen werden heute oft noch als klassische Desktop-Programme konzipiert. Betriebssystemspezifische Programme unter Windows werden dann meist mithilfe von .NET und der Windows Presentation Foundation (WPF) gebaut, die Windows Forms abgelöst hat. Dennoch gibt es sehr viele Altprogramme, die auf diesem technologischen Ansatz beruhen. Die Gründe für eine Desktop-Anwendung liegen sowohl auf der Seite der Anwender als auch bei der Entwicklung. Die Nutzer kennen seit vielen Jahren die Arbeitsweise mit dem Desktop und sind im Umgang mit ihm vertraut. Häufig reagiert diese Art von Programmen auch effizienter und das Arbeiten geht – zumindest dem Gefühl nach – flüssiger von Hand. Dadurch, dass das Programm auf dem lokalen Rechner installiert ist, ist eine Netzwerkverbindung auch nur zum Datenaustausch nötig. Es ist nicht leugnen: Für klassische Büroaufgaben fühlt sich der Desktop sicherer und vertrauter an als eine Webanwendung, gewissermaßen als ob er auf einem soliden Fundament gebaut wäre.

Dennoch gibt es eine Reihe von Gründen, die eine Webanwendung vorteilhafter machen, wie zum Beispiel ein einfacheres Deployment und eine direkte Mehrbenutzerfähigkeit. Sichere und schnelle Netzwerkverbindungen und deutlich verbesserte Möglichkeiten beim Gestalten von Oberflächen haben die genannten Nachteile objektiv gesehen beseitigt.

Für den Entwickler gibt es eine große Anzahl an Programmieransätzen, Technologien und dazugehörigen Frameworks zum Erstellen von dynamischen Webapplikationen. Bekannte Technologien sind etwa ASP.NET, JSP und PHP. Für die Programmierung stehen einfache Editoren oder die bekannten Entwicklungsumgebungen zur Verfügung. Letztere werden dabei in der Regel durch Erweiterungen (Plugins) an die Anforderungen des gewählten Entwicklungsansatzes angepasst.

Jede Technologie stellt andere Ansprüche an die Serverumgebung und auch die Prozesse der Programmentwicklung unterscheiden sich. Bei der Auswahl sind die individuellen Vor- und Nachteile gegeneinander abzuwägen. Oft wird die Entscheidung für eine gewisse (Entwicklungs-)Technologie jedoch auch danach getroffen, welche Kenntnisse und Erfahrungen die beteiligten Entwickler mit dem einen oder anderen Produkt haben.

Trotz der Vielfalt bei den Webtechnologien scheint gelegentlich der richtige Ansatz nicht dabei zu sein; trotz modern gestalteter Oberflächen von Webanwendungen kritisieren



Wisej stellt zwei neue Projektvorlagen in Visual Studio zur Verfügung (Bild 1)

die Nutzer das bekannte Handling des vertrauten Desktops. Ebenso können sich Entwickler mit einem Wechsel vom Desktop ins Web durchaus schertun. Neben anderen Sprachen auf Client- und Serverseite (HTML, CSS, JavaScript, PHP) müssen sie auch eine vollständig andere Architektur verstehen. In diesen Fällen leidet die Produktivität.

Nicht vergessen darf man dabei, dass viele Branchenapplikationen Windows-Forms-Programme sind. Es ist mehr als wünschenswert, wenn bei einer Migration auf eine neue Technologie so viel wie möglich transformiert werden kann.

Das Webentwicklungs-Framework Wisej der amerikanischen Ice Tea Group konzentriert sich auf diese Umstände. Die Website des Projekts verspricht, dass sich damit dynamische Webanwendungen auf gleiche Art und Weise wie Windows-Forms-Anwendungen erstellen lassen [1]. Die Rede ist dabei von „Real Time Web Applications“. Im Fokus steht klar der Geschäftskontext.

Wisej ist nicht dazu gedacht, einfache Webseiten zu erstellen, sondern interaktive Anwendungen, die sich für den Nutzer nahezu wie eine Desktop-Applikation anfühlen. Ebenso soll die Programmierung weitgehend auf den gleichen Ansätzen beruhen wie bei Windows Forms. Die Aussicht, eine in die Jahre gekommene Windows-Forms-Applikation mit vertretbarem Aufwand in eine moderne Webanwendung umzuwandeln, motiviert. Aber natürlich ist Wisej auch für neue Projekte geeignet.

Die Autoren haben Wisej getestet. Die folgenden Textabschnitte fassen ihre Erfahrungen zusammen. Dabei wird es nicht nur darum gehen zu zeigen, wie schnell das Setup von Wisej erledigt ist und wie sich – anhand einer kleinen Beispielanwendung – das Entwickeln damit anfühlt. Ein Blick hinter die Kulissen soll dem Leser auch verraten, welche Webtechnologie sich hinter Wisej verbirgt, die das Framework nahezu vollständig vor dem Entwickler verbirgt. Das Ziel ist, einen Eindruck von der Leistungsfähigkeit und Praxistauglichkeit von Wisej zu bekommen.

## Systemvoraussetzungen und Installation

Webapplikationen auf der Basis von Wisej laufen auf einem IIS-Server ab Version 8. Im Moment wird nur dieser Servertyp unterstützt. Nach der ersten Version will die Ice Tea Group künftig auch weitere Servertypen ansprechen. Das ist mit Sicherheit sehr sinnvoll. Ob es jedoch zügig gelingen wird, Microsofts .NET-Technologien auf einen Apache-Server unter Linux zu übertragen, muss sich erst noch zeigen. Auf jeden Fall würde dieser Schritt das Einsatzgebiet des Frameworks erheblich erweitern. In der Praxis läuft auf einer Vielzahl von Servern nämlich ein Linux-Betriebssystem.

Auf der Seite des Clients ist lediglich ein Webbrowser nötig. Dabei werden die aktuellen Versionen der üblichen Browser unterstützt, also Edge, Internet Explorer, Firefox, Chrome und Safari.

Die Entwicklung einer Wisej-Applikation findet in Visual Studio statt. Das dazu nötige Plug-in kann ab Visual Studio 2012 installiert werden. Sehr positiv ist, dass dafür die kostenlose Community-Version von Visual Studio genügt. Auch die Verwendung der IDE SharpDevelop ist möglich, das

ebenfalls kostenlos angeboten wird. Damit sind die technischen Voraussetzungen bereits erfüllt, alles Weitere bringt das Produkt mit.

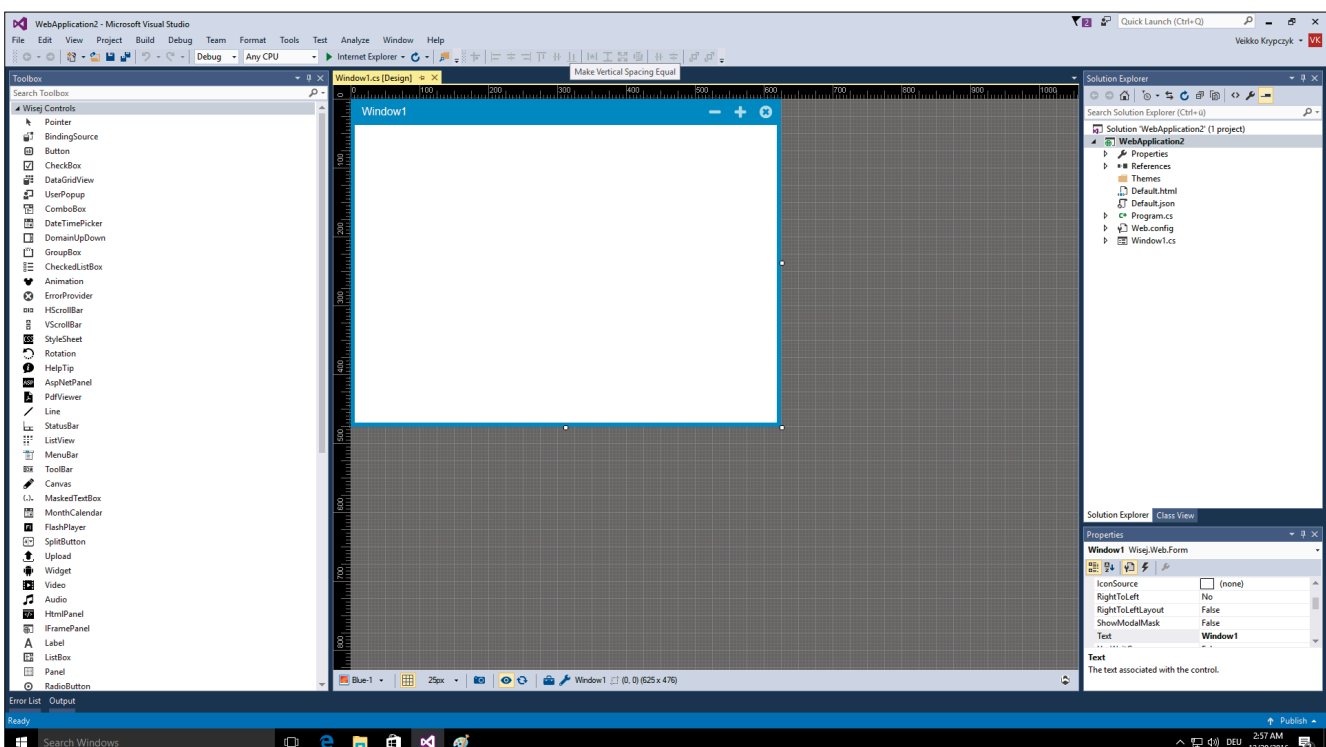
Wisej selbst ist kostenpflichtig. Die Entwicklerlizenz kostet 690 Euro. Eine unabhängige Serverlizenz – zum Hosten von Applikationen auf der Basis von Wisej – ist für 390 Euro zu haben; die Entwicklerlizenz enthält auch eine Serverlizenz. Zum Testen steht eine kostenlose Evaluierungsversion für 30 Tage zur Verfügung. Hierzu muss sich ein Interessierter zuvor beim Hersteller mit Namen und E-Mail-Adresse registrieren. Die Seriennummer für den Testzeitraum wird nach der Registrierung/dem Download angezeigt und zusätzlich per E-Mail übermittelt.

Nach der Registrierung auf der Wisej-Website gelangt man in den Download-Bereich. Die Installationsdatei umfasst erfreulicherweise nur 13,6 Megabyte und ist damit schnell geladen. Installieren Sie Wisej bei geschlossenem Visual Studio. Der Installationsassistent erkennt die vorhandenen Visual-Studio-Installationen und bietet eine abgestimmte Auswahl an. Auf dem Testrechner der Autoren lief Visual Studio 2015, Update 3; die Installation funktionierte ohne Probleme und Rückfragen.

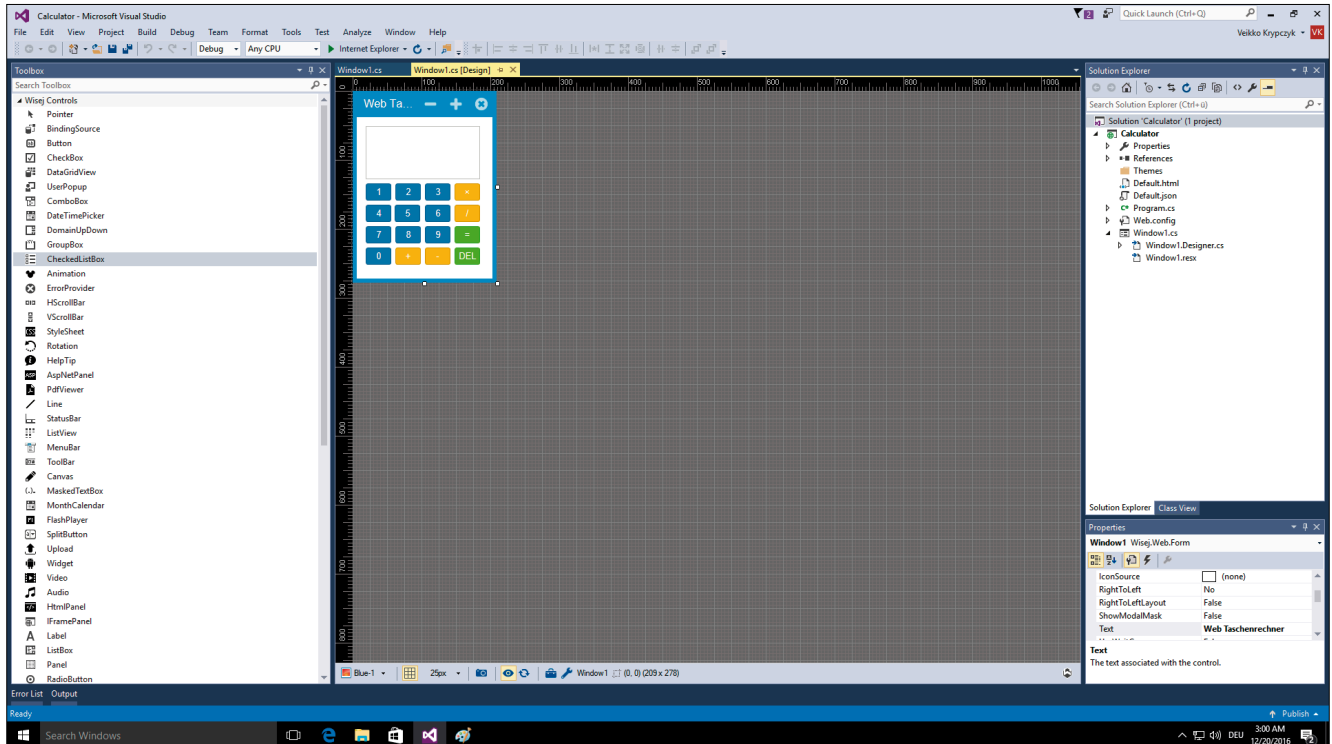
Ice Tea Group arbeitet fortlaufend sehr aktiv am regelmäßigen Update von neuen Versionen. Ende Dezember 2016 war die Version 1.3.16 aktuell. Bis zum Druck dieses Heftes dürften jedoch noch einige Updates von Wisej folgen, sodass sich unter Umständen das eine oder andere geändert hat.

## Probieren geht über Studieren

Wie funktioniert das Ganze nun? Nach dem Start von Visual Studio wird wie gewohnt der Assistent für ein neues Pro- ▶



Der Assistent erzeugt automatisch einen vollständigen Applikationsrumpf (Bild 2)



Per Drag-and-drop wird die Oberfläche zusammengestellt (Bild 3)

jekt aufgerufen. Wisej stellt zwei neue Projektvorlagen bereit: *Web Application* und *Web Desktop Application* (Bild 1). Eine *Web Application* zeigt das typische Verhalten einer Webanwendung im Browser; eine *Web Desktop Application* enthält zusätzlich einen eigenen Desktop, sodass sie dem Anwender noch mehr das Gefühl vermittelt, mit einem Desktop-Programm zu arbeiten. Die Projektvorlagen befinden sich in den Kategorien *Visual C#* und *Visual Basic*, das heißt, mit beiden Sprachen ist es möglich, eine Wisej-Anwendung zu entwickeln.

Für einen ersten Test wird die Projektvorlage *Web Application* gewählt. Der Assistent von Visual Studio erzeugt in gewohnter Weise den Projektrahmen, also die Struktur des Projekts inklusive eines leeren Fensters. Der erfahrene Windows (-Forms)-Programmierer wird sich sofort heimisch fühlen. Auf der linken Seite befindet sich die bekannte Toolbox und diese enthält wichtige und vertraute Steuerelemente für das Gestalten der Benutzeroberfläche (Bild 2).

Hier wird jedoch keine Oberfläche für eine Desktop-Anwendung gebaut, sondern die Oberfläche für eine Webapplikation. Das Prinzip muss hier nicht umfangreich erläutert werden; wer bereits mit Windows Forms programmiert hat, wird sofort damit zurechtkommen. Die Steuerelemente werden auf dem Formular platziert und die Eigenschaften im Editor wunschgemäß angepasst. Um das sogleich auszuprobieren, haben die Autoren eine *TextBox* und meh-

re *Button*-Objekte im Stil eines einfachen Taschenrechners platziert (Bild 3). Anpassungen im Code sind für das Gestalten der Benutzeroberfläche nicht vorgesehen, das heißt, der Prozess des Oberflächenentwurfs mithilfe des UI-Builders ist wie bei Windows Forms abschließend.

Nun wird es spannend: Die neue Applikation soll erstmals starten. Logik gibt es zwar noch keine, aber das Fenster innerhalb der Webseite müsste angezeigt werden.

Der Start erfolgt direkt aus der Visual-Studio-Entwicklungsumgebung heraus. Über den bekannten grünen Pfeil wird der Start der Webapplikation in den installierten Webbrowsern angeboten. Da es sich um eine dynamische Webanwendung handelt, muss im Hintergrund ein Server gestartet werden. Visual Studio übernimmt den Start und sämtliche

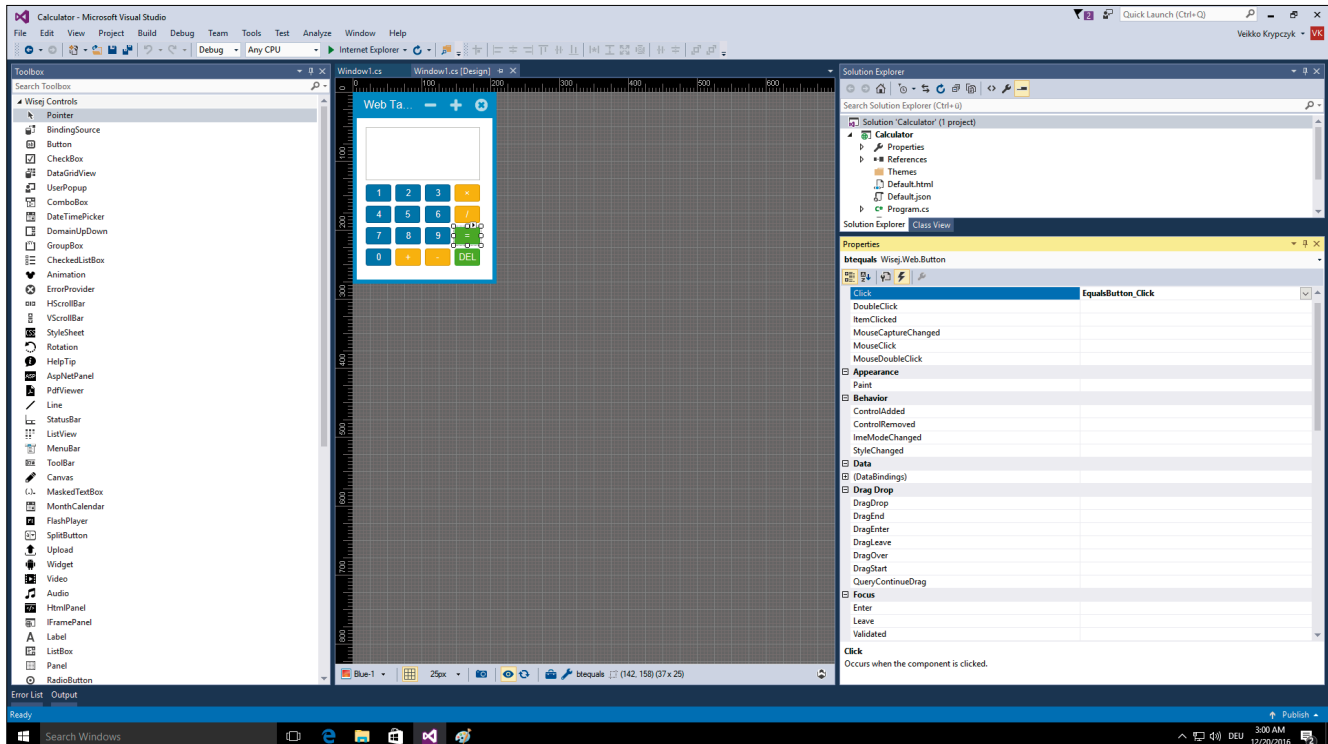
Konfigurationsarbeiten von IIS-Express. Dieses lauscht unter der üblichen lokalen Adresse localhost und im vorliegenden Fall auf Port 56049.

Wie gesagt: Die dafür notwendigen Einstellungen erfolgen ohne jeglichen manuellen Eingriff.

Das Ergebnis ist in Bild 4 zu sehen. Das Dialogfeld des Taschenrechners wird als eigenes Fenster innerhalb des Browsers angezeigt. Entstanden ist also eine Webapplikation, die wie eine Desktop-Anwendung anmutet. Natürlich passiert noch nichts, wenn man auf eine Schaltfläche tippt. Dies kann auch gar nicht sein, denn eine Programmlogik ist noch nicht implementiert. Das soll sich aber gleich ändern.



Die Webanwendung im Browser, hier noch ohne Funktionalität (Bild 4)



Die Programmlogik wird bei Wisej über Events an die Steuerelemente gebunden (Bild 5)

Also: Beenden Sie die Applikation – die aktuell im Debug-Modus gestartet ist – und zurück zu Visual Studio.

In Wisej lässt sich die Programmlogik wie bei Windows Forms über das Verknüpfen der zugehörigen Ereignisse einbinden. Dazu wählen Sie wie gewohnt einen *Button* aus (zum Beispiel den für das Gleichheitszeichen) und klicken im Eigenschaften-Editor auf das Symbol für die Ereignisse. Es zeigen sich daraufhin alle möglichen Events zu einem *Button*

(Bild 5). Hier lässt sich das *Click*-Ereignis mit Code verknüpfen; ein Doppelklick auf das gewünschte Ereignis erstellt den Rumpf der dazugehörigen Methode in der Code-behind-Datei. Die Programmlogik können Sie im gewohnten Stil in C# oder Visual Basic erstellen. Listing 1 zeigt den Quellcode für die Schaltfläche mit dem Gleichheitszeichen. Für den minimalen Taschenrechner ist nicht viel mehr nötig. Den vollständigen Quelltext des Beispiels können Sie in dem zum Ar- ▶

### ● Listing 1: Der Code für den Taschenrechner (Auszug)

```
private void EqualsButton_Click(
    Object sender, EventArgs e)
{
    if(mode == 1)
    {
        mode = 2;

        //Einzelne Zeilen auslesen
        String line0 = textBox1.Lines[0];
        String line1 = textBox1.Lines[1];

        val1 = Convert.ToDouble(line0);
        operation = line1.Substring(0, 1);
        line1 = line1.Substring(1);
        val2 = Convert.ToDouble(line1);
        double result = 0;
        switch (operation)
        {
            case "+":
                result = val1 + val2;
                break;
            case "-":
                result = val1 - val2;
                break;
            case "x":
                result = val1 * val2;
                break;
            case "/":
                result = val1 / val2;
                break;
        }
        textBox1.Text += "\n= " + result;
    }
}
```

tikel gehörenden Projekt auf der Heft-Disk studieren.

Der erneute Start der Beispielapplikation zeigt: Jetzt funktioniert der Taschenrechner auf erwartete Weise (Bild 6).

Machen wir uns kurz klar: Oberfläche und Programmlogik der Webanwendung wurden komplett in C# erstellt, und dabei haben Sie keine einzige Zeile in HTML, CSS oder JavaScript programmiert.

### Unter der Motorhaube

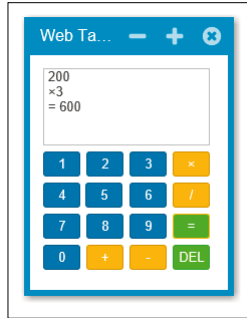
Beim Entwickeln mit Wisej vergisst man schnell, dass man gerade eine Webapplikation baut. Der Entwickler muss sich keinerlei Gedanken über das Zusammenspiel von HTML/CSS für die Oberfläche und JavaScript für die Logik machen. Das alles passiert unter der Oberfläche und der Entwickler kann sich vollständig darauf konzentrieren, sein Geschäftsmodell umzusetzen. Auch den Datentransfer zwischen Client und Server übernimmt Wisej.

Bild 7 zeigt den Aufbau des Frameworks und das Zusammenspiel von Client und Server. Auf der Seite des Clients kommt intern die JavaScript-Bibliothek Qooxdoo zum Einsatz. Diese bildet das Aussehen der Windows-Forms-Controls im Browser nach. Wisej erweitert diese Elemente um eigene Elemente und die Möglichkeit, Controls von Dritten einzubinden. Auf der Seite des Servers wird letztendlich mit dem .NET Framework programmiert. Damit kann eine Wisej-Anwendung die Möglichkeiten und Datenbankmechanismen von .NET nutzen – eine vertraute Welt.

### Merkmale und Optionen

Wisej bietet einige interessante Merkmale. Die wichtigen Punkte hierzu sollen im Folgenden kurz erläutert werden:

- **Datenbanken:** Branchenapplikationen für das Web basieren oft auf Datenbanken. Zur Datenbankbindung stehen in Wisej die bekannten Steuerelemente aus Windows Forms zur Verfügung, zum Beispiel das Datengitter. Mittels Datenbankbindung wird die Verknüpfung zur Datenbank hergestellt. Interessant wird es, wenn Anwendungen gebaut werden, die auf mehrere 10.000 Datensätze und mehr zugreifen. Für das Leistungsverhalten ist unter anderem das Ausmaß des Datentransfers zwischen Client und Server entscheidend. Umfassende Praxistests stehen noch aus. Erste Erfahrungsberichte sprechen jedoch davon, dass die Leistung sehr gut sein soll und es zu keinen Einschränkungen bei der Arbeit mit großen Datenmengen kommt.
- **Designer:** Das Entwerfen der Webseite ist sehr einfach. Der Hersteller bezeichnet den Editor als „Pixel Perfect Designer“. Das Ergebnis sieht zur Laufzeit so aus, wie es zur Entwurfszeit zusammengestellt wird.
- **Responsives Webdesign:** Die Applikationen passen sich den unterschiedlichen Geräteklassen automatisch an. Das ist wichtig, weil die Geräte, auf denen die Anwendung laufen könnte, immer vielfältiger werden.
- **Data Update:** Über JSON-Datenpakete erfolgt der Datenaustausch zwischen Client und Server.



Der Taschenrechner als fertige Webanwendung (Bild 6)

- **Erweiterbarkeit:** Der Funktionsumfang lässt sich über Erweiterungen vergrößern. Beispiele sind Extensions für Hintergrundaufgaben oder zum Lokalisieren der Applikation.

Auf der Wisej-Website steht zu diesen Programmfunktionen und zu anderen Merkmalen eine Dokumentation zur Verfügung. Ebenso stehen zahlreiche Beispielanwendungen bereit. Die Online-Dokumentation war zum Testzeitpunkt zwar noch nicht vollständig, die wichtigsten Punkte waren jedoch bereits ausführlich beschrieben.

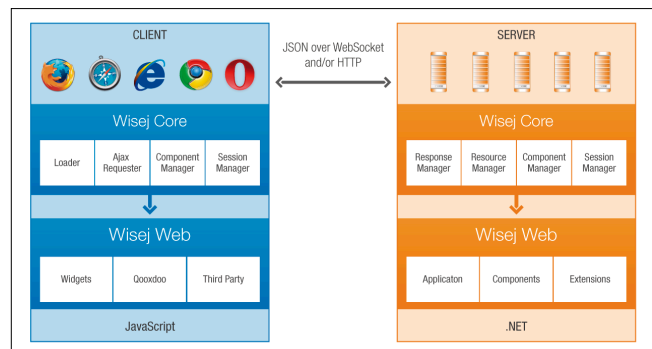
### Das Design

Eine weitere interessante Funktion ist unbedingt zu erwähnen: Das Design von Wisej-Anwendungen lässt sich sehr leicht anpassen. Das komplette Aussehen kann man mit nur einem Mausklick wechseln. Um das zu erreichen, wird einfach das zugrunde liegende UI-Thema ausgetauscht. Bereits während des Entwerfens werden die Änderungen sichtbar. In Bild 8 wurde das Thema des Taschenrechners von *Blue-1* auf *Classic-2* geändert. Zur Auswahl stehen weitere vordefinierte Themen.

Ebenso ist es möglich, die Farbauswahl eines Themas für die einzelnen Elemente selbst anzupassen oder gar ein eigenes Thema zu erstellen. Auch dies kann im Dialog mit der IDE erfolgen. Wisej stellt dazu ein eigenes kleines Werkzeug – den Wisej Theme Builder – zur Verfügung. Mit ihm ist es möglich, beispielsweise das Aussehen von Typen von Steuerelementen, zum Beispiel alle *Button*-Elemente, zentral zu definieren. Der Wisej Theme Builder generiert den entsprechenden Code automatisch. Auch hier ist kein manueller Eingriff nötig.

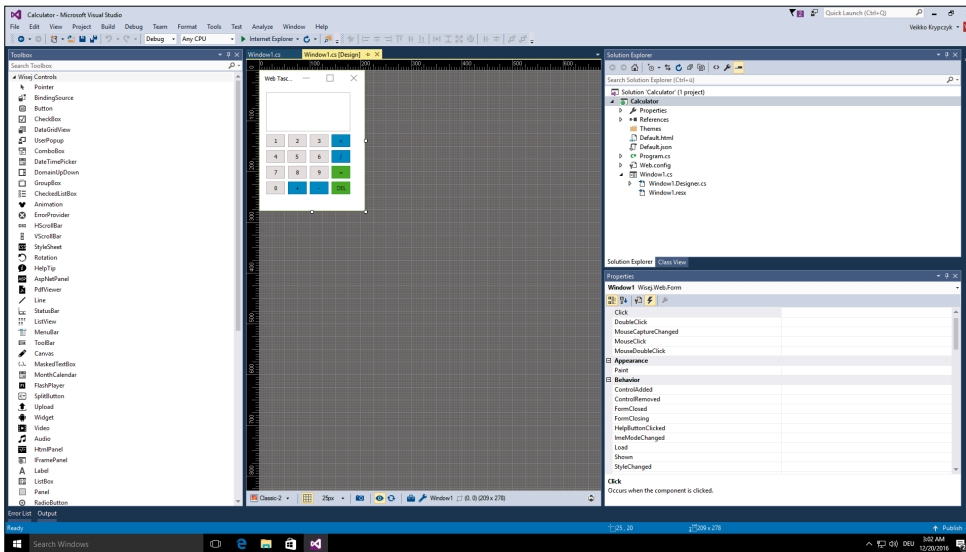
### Wisej als Migrationswerkzeug

Ein besonderer Reiz von Wisej könnte darin liegen, das Tool als Migrationswerkzeug einzusetzen. In der Praxis gibt es ja unzählige Windows-Forms-Anwendungen, die bereits seit vielen Jahren im produktiven Betrieb arbeiten. Der Migrationsdruck auf eine Windows-Forms-Applikation rührt nicht nur daher, dass diese UI-Technologie in der Window Presen-



Die Architektur von Wisej (Bild 7)





Einfaches Ändern des Oberflächendesigns mithilfe vordefinierter UI-Themen (Bild 8)

tation Foundation schon seit vielen Jahren einen Nachfolger gefunden hat, sondern auch aus der Tatsache, dass sich die heutigen Anforderungen mit einer webbasierten Applikation oft besser umsetzen lassen. Wisej könnte Hilfe bei der Migration von Windows-Forms-Anwendungen bieten. Die Ähnlichkeit des Programmieransatzes macht das Tool dafür interessant.

Die UI-Anpassung könnte mithilfe einiger Korrekturen erfolgen. Häufig wird es wohl genügen, die *using*-Klauseln anzupassen. Die Eigenschaften und Ereignisse von Windows-Forms-Controls und Wisej-Steuerelementen ähneln sich sehr. Auch die Fachlogik dürfte sich weitgehend übernehmen lassen, da beide Programmtypen C# oder Visual Basic verwenden.

Die folgenden Punkte darf man bei einer anstehenden Migration von einer Desktop- auf eine Webapplikation jedoch nicht vergessen. Für diese Probleme muss man individuelle Lösungen erarbeiten:

- Mehrbenutzerbetrieb: Windows-Forms-Anwendungen laufen stets auf einem Rechner. Wisej-Webanwendungen starten auf dem Server. Das heißt, diese Anwendungen müssen gegebenenfalls mit konkurrierenden Datenzugriffen mehrerer Nutzer umgehen können.
- Nutzerverwaltung: Webapplikationen benötigen in der Regel eine Nutzerverwaltung. Diese ist in der Regel neu einzurichten.
- Hardware-Zugriff: Zugriffe aus den Windows-Forms-Anwendungen auf die lokale Hardware, zum Beispiel auf einen Drucker, laufen bei einer Server-Applikation ins Leere. Auch hier ist ein anderer Ansatz nötig.

Wie immer ist bei einem Migrationsprojekt abzuwägen, welchen Aufwand dieses gegenüber einer Neuentwicklung verursacht und welche Transformationsprozesse sich durch intelligente Werkzeuge automatisieren lassen. Die Autoren sehen hier durchaus gute Chancen, Wisej erfolgreich einzusetzen.

## Fazit

Um nun am Schluss die im Titel aufgeworfene Frage zu beantworten: Wisej ist kein alter Wein in einem neuen Schlauch. Mittels Wisej ist es möglich, Webapplikationen zu erstellen, die sich wie Desktop-Anwendungen anfühlen. Das Produkt ist auf Geschäftsanwendungen ausgerichtet.

Das Erstellen macht einen durchdachten und leicht zu erfassenden Eindruck. Gefallen hat auch, dass sich das Design schnell anpassen lässt. Mit einem Klick kann der Entwickler das Thema wechseln und die Anwendung wartet mit einem anderen Look auf. Für

noch individuellere Anpassungen steht ein Themen-Editor zur Verfügung. Ein erfahrener Desktop-Programmierer wird es schätzen, so gut wie nicht mit den dahinterstehenden Webtechnologien in Berührung zu kommen.

Eine weitere Stärke von Wisej könnte in der Migration von Altanwendungen liegen. Zwar gehört dazu mehr, als nur Code aus der Windows-Forms-Anwendung umzuwandeln. Mit Wisej erscheint der Aufwand jedoch geringer zu sein als etwa beim manuellen Umwandeln in Webtechnologie. Ist der Entwicklungsprozess zudem effizient, dürften sich die Lizenzgebühren schnell amortisieren. So ziehen die Autoren das Fazit: Die nächste Webapplikation wird in Wisej erstellt.

Der Quellcode zur Beispielanwendung des Artikels ist auf der Heft-Disk ebenso wie auf der Website der dotnetpro und bei LARInet [2] zu finden. ■

[1] Wisej, <http://wisej.com>

[2] LARInet, <http://larinet.com>



### Dr. Veikko Krypczyk

hat Betriebswirtschaftslehre mit dem Schwerpunkt Wirtschaftsinformatik studiert und darin promoviert. Er ist Entwickler und Fachautor.

<http://it-fachartikel.de>



### Olena Bochkor

hat Betriebswirtschaftslehre unter anderem mit dem Schwerpunkt Wirtschaftsinformatik studiert. Weitere Informationen zu diesen und anderen Themen der IT finden Sie unter

<http://it-fachartikel.de>

