



NEUE WEGE BEIM REKRUTIEREN VON NACHWUCHSENTWICKLERN

Übungsraum für Junioren

Bereits vor dem ersten Bewerbungsgespräch trennt das „Digital Dojo“ für ein Praktikum oder eine Lehrstelle die Spreu vom Weizen.

Qualifizierte und motivierte Mitarbeiter zu finden ist eine der wichtigsten Aufgaben, die ein Unternehmen bewältigen muss. Dabei wird es bestrebt sein, eine zu große Fluktuation der Mitarbeiter zu vermeiden und sicherzustellen, dass neue Kollegen den Ansprüchen der Firma entsprechen. Meist lernt man sich in einem ersten Vorstellungsgespräch ein wenig kennen; manchmal folgt noch eine Probearbeit für einen oder mehrere Tage. Doch speziell im Bereich der Softwareentwicklung können dadurch einige der essenziellen Qualifikationen oft nicht überprüft werden.

Das IT-Unternehmen Count It hat sich ein digitales Dojo ausgedacht. Ein Dojo ist ursprünglich eine Übungshalle für japanische Kampfkünste. Dementsprechend ist die Grundidee eines digitalen Dojo, jungen Nachwuchsentwicklern im Internet eine Möglichkeit zu bieten, zuerst einfache und danach immer komplexere Aufgaben zu lösen. Die Überprüfung der Aufgaben erfolgt automatisiert im Hintergrund und lässt sich daher jederzeit und ohne Verzögerung ausführen. Einen Anreiz zur Teilnahme bilden einerseits Sachpreise und andererseits Karriereaussichten. Ein Logo, wie es in Bild 1 zu sehen ist, kann die geforderten Technologien grafisch sym-

bolisieren und dem Nachwuchsprogrammierer bereits sagen, dass diese im Unternehmen eingesetzt werden. Längerfristiges Ziel ist es, die Vergabe von Praktika an Schüler und Studenten über diese Plattform abzuwickeln. Durch die gestellten Aufgaben können somit noch vor Bewerbungsgesprächen erste Eindrücke über Können und Lösungsansätze der Teilnehmer gewonnen werden.

Der Prozess

Das digitale Dojo [1] ist im ersten Halbjahr 2015 online gegangen. Momentan steigt die Zahl der Mitglieder. Je nach Anmeldestatus ist der erste Programmierwettbewerb für Herbst 2015 oder Sommer 2016 geplant. Dabei sind Sachpreise zu gewinnen und es soll ein Tischfußball-Turnier geben. Nach diesem ersten persönlichen Kennenlernen sollen weitere Übungen mit modernen Technologien, Webcasts und E-Books den Kontakt aufrechterhalten. All diese Schritte sollen direkt über das Digital Dojo realisiert werden.

Der persönliche Kontakt mit den Teilnehmern soll auf keinen Fall zu kurz kommen, daher werden die „Dojonauten“ zu monatlichen internen IT-Schulungen eingeladen, bei de-

nen ein Firmenmitarbeiter seine Kollegen unterrichtet. Die Themen sind sehr unterschiedlich und reichen von neuen Technologien über Tools bis hin zu den „lessons learned“.

Daneben werden noch regelmäßig Webcasts aus der Virtuellen Akademie von Microsoft [2] gemeinsam angesehen. Sie sind als Abendveranstaltungen konzipiert und sollen für den Austausch mit Gleichgesinnten sorgen – bei Getränken und Knabberereien und beim Kräfteressen im Tischfußball.

Technische Umsetzung

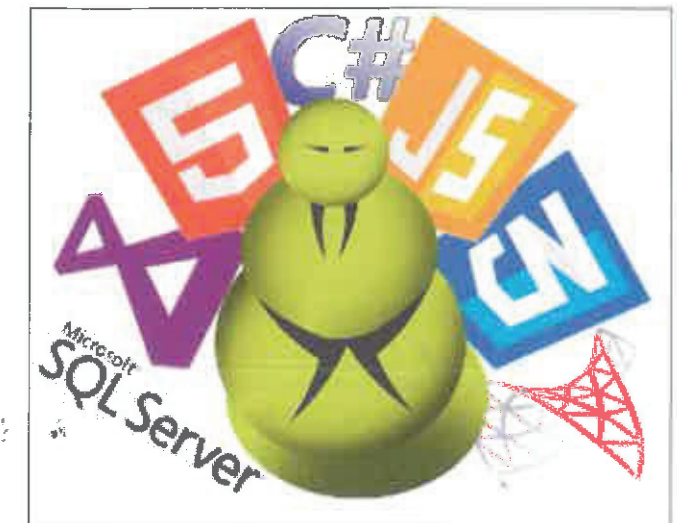
Eine Übung wird als Webseite angeboten, die im Open-Source-CMS Orchard für .NET [3] verwaltet wird. Neben den Anmelde-Feldern bietet sie auch erste Übungen. Diese drei Beispiele müssen für die Registrierung gelöst werden. Dabei handelt es sich um je ein Beispiel in C#, SQL und JavaScript. Die Auswertung der Lösungen erfolgt im Backend, wo neben einer Analyse der übermittelten Daten die Beispiele auch tatsächlich ausgeführt werden. Wer ins Dojo eintreten will, muss alle drei Aufgaben korrekt lösen.

Bei der ersten Aufgabe gilt es ein Interface in C# zu implementieren:

```
public interface ICalculator
{
    int Calculate(int valueA, int valueB,
        Calculations operationMethod);
}

public enum Calculations
{
    Add, Subtract, Multiply, Divide
}
```

Wenn der Teilnehmer das Beispiel implementiert hat, übermittelt er die Lösung an den Server, der sie analysiert. Aus Sicherheitsgründen sind etwa bestimmte Namensräume nicht erlaubt. Außer über solche Fehler aufgrund von Sicherheits-



Digital-Dojo-Logo (Bild 1)

vorgaben wird der Teilnehmer auch über Compilerfehler, logische Fehler oder eine korrekte Lösung informiert.

Die implementierte Klasse wird zur Laufzeit mit der .NET-Klasse *CSharpCodeCompiler* übersetzt. Die Assembly, die dabei entsteht, wird danach über Reflection aufgerufen und ausgeführt. Auf diese Weise lassen sich alle möglichen Codestücke zur Laufzeit kompilieren und ausführen, solange sie ein bestimmtes Interface implementieren.

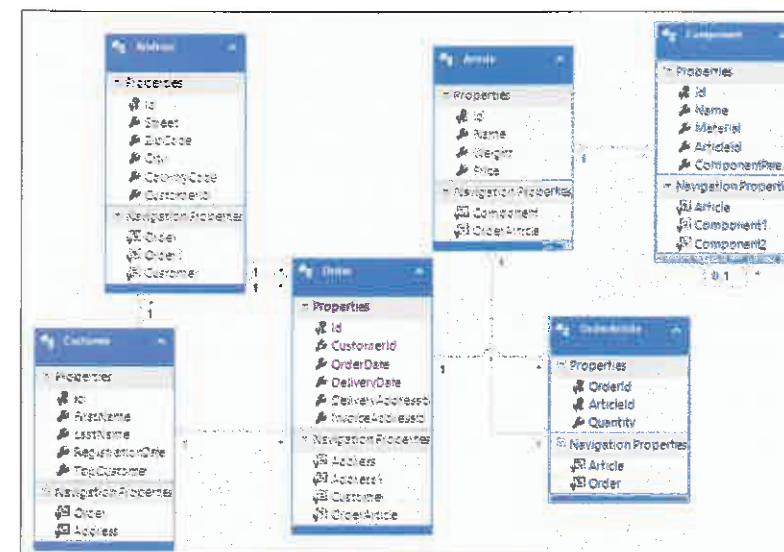
Die zweite Aufgabe prüft die Datenbank-/SQL-Kenntnisse des Teilnehmers. Ihr liegt die Datenbank zugrunde, die in Bild 2 zu sehen ist. Dahinter steht die Anwendung eines Unternehmens, das die besten fünf Aufträge aus dem ersten Quartal 2014 ermitteln will. Diese fünf Aufträge müssen nach Gesamtwert sortiert werden. Die Herausforderung besteht darin, dass die Lösung mehrere Tabellen benötigt. Auch hier wird der Teilnehmer am Schluss über die Lösung aufgeklärt.

Die dritte, eher einfache Aufgabe aus dem Bereich Webentwicklung rundet die Aufnahmeprüfung ab. Dabei sind die vier Grundrechenarten zu folgenden Aufrufen zu implementieren:

```
add(a, b)
subtract(a, b)
divide(a, b)
multiply(a, b)
```

Die Eingabe ist in drei Felder geteilt: für HTML, CSS und JavaScript. Für den Eintritt in das Dojo müssen die vier Methoden korrekt implementiert werden. Bei dieser Aufgabe erhält der Teilnehmer auf zwei Wegen Rückmeldung. Zum einen wie bei den ersten beiden Beispielen und zum anderen wird seine Lösung auf der Seite dargestellt, sodass er seinen Taschenrechner sofort testen kann.

Die Editoren für JavaScript, HTML und CSS werden mit der JavaScript-Bibliothek CloudEdit umgesetzt [4]. Sie ermöglicht es, auf einfache ▶



Das ER-Diagramm für die SQL-Aufgabe (Bild 2)

Weise Editoren für die drei Technologien im Webbrowser darzustellen. Sie unterstützt HTML5, CSS3 und JavaScript-Entwicklungen mit jQuery 1.x. Natürlich ist jQuery dabei kein Muss; auch reines JavaScript ohne zusätzliche Bibliotheken ist möglich. Wie in Bild 3 zu sehen, erinnert die Darstellung der Editoren an jsFiddle, eine sehr populäre Online-Plattform zum Schreiben, Testen und Veröffentlichen von kleinen JavaScript-Programmen.

CloudEdit baut auf dem Editor-System Ace auf [5]. Dieses bietet Syntaxhervorhebung für mehr als 110 Programmiersprachen, die sich auch noch durch eigene Sprachdefinitionen erweitern lassen, und hat über 20 vordefinierte Designs. Wie in vielen IDEs passt das System die Einrückung von Zeilen und Blöcken automatisch an und weist bei JavaScript, CoffeeScript, CSS und XQuery direkt auf Syntaxfehler hin. Etliche weitere aus Editoren bekannte Merkmale sind ebenso enthalten: Drag-and-drop, korrekte Zeilenumbrüche und so weiter.

Validierung von JavaScript

Zum Übersetzen von JavaScript wird die Open-Source-Bibliothek Jint verwendet [6]. Sie enthält einen JavaScript-Interpreter für .NET und ist zu ECMA-Script 5.1 kompatibel. Das Skript läuft in einer eigenen abgekapselten Umgebung (Sandbox) und hat somit keinen Zugriff auf das außerhalb liegende System.

Zusätzlich bietet Jint noch weitere Funktionalitäten. Zum Beispiel ist es möglich, aus dem interpretierten JavaScript-Code CLR-Objekte zu erzeugen. Dabei werden Variablen, die in JavaScript definiert sind, in entsprechende CLR-Objekte übernommen. Dies geht sogar so weit, dass die Funktionen in Delegationen übernommen werden und somit von .NET aus aufgerufen werden können. Auf diese Weise lassen sich aus der .NET-Laufzeitumgebung die für die JavaScript-Aufgabe implementierten Methoden ausführen und prüfen.

Das System im Hintergrund

Das zugrunde liegende CMS Orchard wird von einer sehr aktiven Community entwickelt und verwendet. Das System ist sehr flexibel aufgebaut und kann an vielen Punkten erweitert werden; es ist unter der BSD-Lizenz veröffentlicht.

Es gibt eine große Bandbreite an verfügbaren Themen und Modulen für Orchard, zudem kann es durch eigene Module und Erweiterungen weiter angepasst werden. Die Themen lassen sich, solange die vorgegebenen Richtlinien eingehalten werden, jederzeit austauschen und anpassen. Da das System in C# entwickelt und sein Code frei zugänglich ist, gilt das auch für die Kernbereiche.

Orchard kann mit seinem ersten Release im Jahr 2011 als eher junges CMS gelten und seine Architektur als „State of the Art“. Das System basiert auf ASP.NET MVC und NHibernate. Die sehr gelungene Anwendung des Entwurfsmusters Dependency Injection ist die Basis für die Erweiterbarkeit und Testbarkeit des Systems. Die Datenbankschicht auf NHibernate-Basis bietet komfortable Zugriffe auf die Datenbank und ermöglicht mit dem Versionsmanagement des CMS automatische Datenbank-Updates auf einfachstem Weg.



Der HTML-Editor für die JavaScript-Aufgabe (Bild 3)

Zusammenfassung

Neue Nachwuchsprogrammierer zu finden, die Wissen und Begeisterung mitbringen, ist eine wichtige Aufgabe für IT-Unternehmen. Das digitale Dojo geht dabei neue Wege. Um herauszufinden, ob junge Kandidaten (Schüler, Studenten) für ein Praktikum geeignet sind, müssen sie beim Registrieren im Internet im Digital Dojo drei Aufgaben lösen. Dieser Test soll zeigen, wie es um ihre Kenntnisse bestellt ist. Sind sie dabei erfolgreich, werden sie als Praktikanten in die Arbeit des IT-Unternehmens eingebunden.

Auf der einen Seite soll es ermöglichen, die Talente für die Zukunft des Unternehmens bereits frühzeitig persönlich kennenzulernen; auf der anderen Seite sollen dadurch die modernen Vorgehensweisen des Unternehmens auch nach außen getragen werden. Im Rahmen des SEO-Monitorings wird auch die Verwendung des Dojo analysiert. Der Erfolg wird in einigen Jahren an den Besuchern und den eingetretenen Mitgliedern gemessen werden können. ■

[1] Digital Dojo, <http://dd.countit.at>

[2] Microsoft Virtual Academy, www.microsoftvirtualacademy.com

[3] Orchard, www.orchardproject.net

[4] CloudEdit, www.dotnetpro.de/SL1510DigitalDojo1

[5] Ace, <http://ace.c9.io>

[6] Jint, www.dotnetpro.de/SL1510DigitalDojo2



Markus Szöky

ist Abteilungsleiter, .NET-Softwarearchitekt und agiler Projektleiter bei der Count It Group in Hagenberg/Oberösterreich, die maßgeschneiderte ERP-, DMS-, Individualsoftware auf Basis von .NET bietet. Sie erreichen ihn über m.szöky@countit.at oder www.countit.at.

dnpCode

A1510DigitalDojo