DIE ZEITSCHRIFT FÜR DIE MICROSOFT DEVELOPER COMMUNITY

VisualStudio1.de

01

1. Quartal 2016

DE 8,50 EUR
A/CH 9,50 EUR
EU 9,50 EUR

Microsoft R im Einsatz

Custom WPF Controls

Machine Learning

Neues in SharePoint 2016

Visual Studio Erweiterungen

Universal Windows Platform

(01)4196066108500 6

SharePoint konferenz Wien

20.-22. Juni 2016 Wien

Auf der Konferenz erwartet Sie wieder ein umfangreiches Programm, aus dem Sie frei nach Ihren Interessen und Bedürfnissen wählen können.

Sie sind SharePoint Developer, Administrator, Microsoft Partner, oder einfach nur an SharePoint und im Besonderen an SharePoint 2016 interessiert?

Dann ist die SharePoint konferenz genau das Richtige für Sie!

2 Tage

3 parallele Tracks

Über 30 Sessions

3 Workshops am 20.Juni

20 Experten

MVP's

Kontakt: Mail: events@ppedv.de oder Telefon: 0049 86779889-69

ppedv

EINS ZWEI ODER DREI

Ruft mich diese Woche ein Schulungskunde an und frägt mich um Rat. Macht man doch gerne, auch wenn die Frage unbeantwortet blieb und das Gespräch eine Stunde dauerte. Lag es an komplexen technischen Problemen? War die Aufgabenstellung etwa unklar? Wollte der Mann gar zum Mars fliegen?

Das war es alles nicht. Ein auf Winforms basiertes CRM sollte seinen Weg in die Zukunft finden. Klassische Client Server Architektur mit Microsoft von SQL Server bis zur UI.

„Man hört und liest ja überall, JavaScript und Angular und da gibt's ja zwei Versionen und soll man den Server mit Node.js machen und dann aber auch eine NoSQL Datenbank?"

Zu tiefste Verunsicherung macht sich breit, schließlich hängt das Wohl und Gedeih eines Unternehmens von dieser Technologie Entscheidung ab. Auch veritable Anhänger des Microsoft Technologie Stacks verkünden nun „The real cross platform" könne nur per JavaScript codiert werden.

Der Entwicklerstab ist aber auf C# WPF und sogar schon auf UWP eingeschworen. Wird diese Plattform es schaffen die nötige Reichweite zu erzielen, oder werden die Redmonder ähnlich wie bei Silverlight einfach über Nacht den Stecker ziehen? Oder wird gar Windows beerdigt, da ja nun SQL Server samt LINUX von Microsoft gibt und sowieso alle neuen und wichtigen Apps selbst von Microsoft auf Android und IOS entwickelt werden.

„und wie ist das eigentlich mit XAMARIN, kann man da mit seinen .NET Skills, das echte real native Cross Platform...?"

In der Tat eine gute Frage, die sich dieser Mensch mit hunderten wenn nicht tausenden anderen teilt. Speziell aus dem JavaScript und Open Source Lager wird der eigene Sieg bereits lauthals verkündet. Auf Twitter aufgeschnappt – „Ein JavaScript Framework wird schneller schlecht wie eine Packung Milch".

So richtet sich der Blick nach Redmond. Außer vielen rosa Herzen und wir haben euch alle lieb, sind da nur die „dunklen" Wolken der Cloud Strategie deutlich zu erkennen. .NET Core 1 hätte das Zeug node.js auf allen Plattformen entgegen zu treten, steht aber erst am Anfang.

Die wirklich treibende Kraft ist google, auf einer für sie gratis arbeitenden JavaScript Community Welle reitend. Aber auch das kann sich wieder ändern. So wie die Bedeutung von Java.

Kann ich nur eine alte Bauernregel das Wetter betreffend leicht geändert zitieren. „Alles ändert sich oder es bleibt wie es ist, Hauptsache Visual Studio und Intellisense".

Ihr nicht minder ratloser
HANNES PREISHUBER

INHALT AUSGABE 01/2016

Editorial 3

Inhalt 4

Neues aus dem Netz 6

No Spectators, only Participants 8

Impressum 33

Comic 49

Termine 49

Bücher 88

Kolumne 90

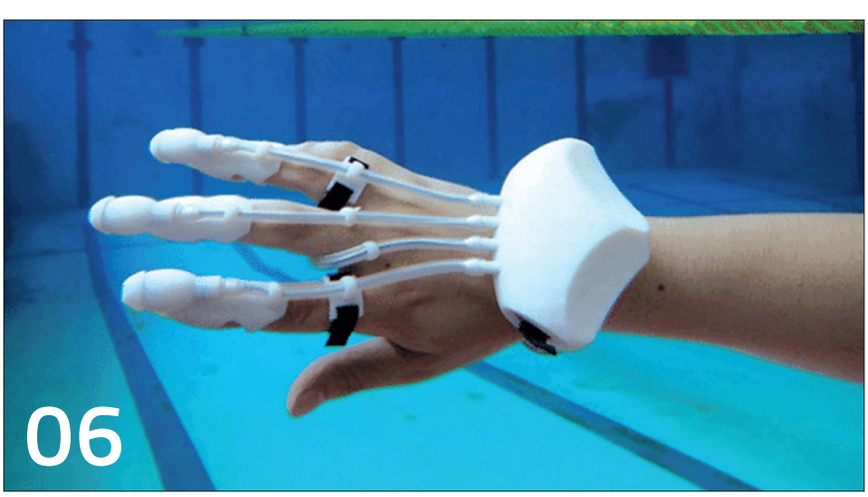
BUSINESS

Präferenzanalyse mit „Microsoft Social Engagement“ 51

Fest verschlossen – Anwendungen mit Lizenz-bibliotheken absichern 55

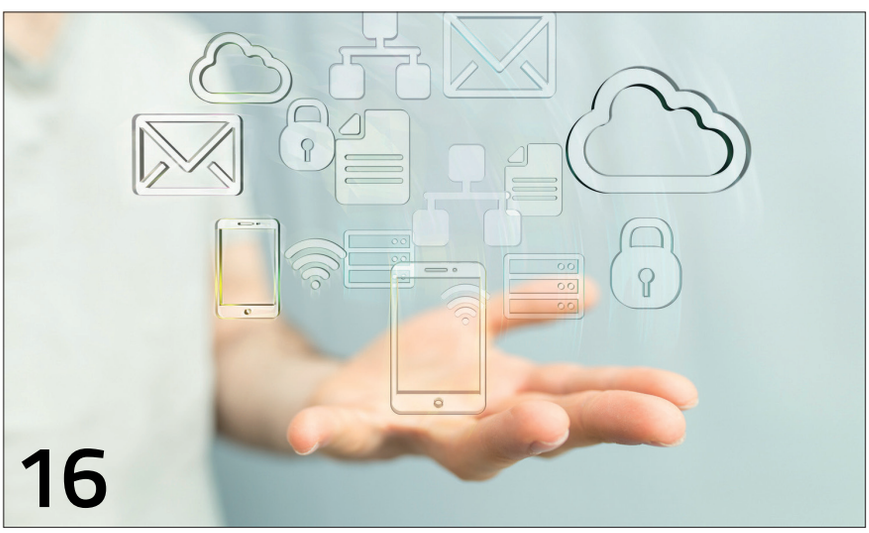
Neuerungen in SharePoint 2016 84

09



13

39



16

84

55

GRUNDLAGEN

Robotik und künstliche Intelligenz 9

Neuigkeiten von DreamSpark 13

Die Microsoft Cloud in Deutschland 16

Stillstand vermeiden 20

Auf den Kunden zugehen 24

E-Commerce mit Smartstore.Net 39

Einstieg in Azure Machine Learning 43

Universal Windows Platform API 62

CODING

Visual Studio Erweiterbarkeit ... 25

Effektives Unit Testing mit Fakes 58

Custom Controls in WPF 65

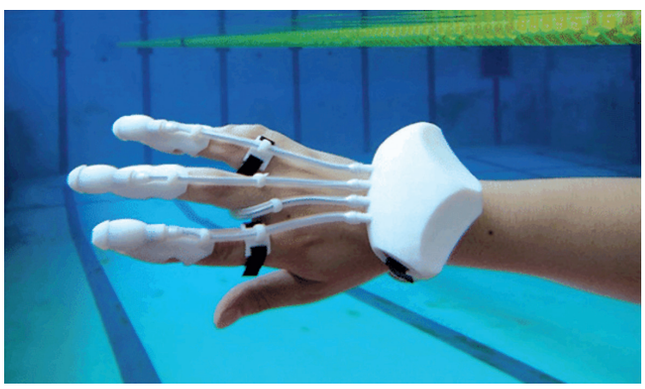
Microsoft und R 72

Workshop - Mobile Applikation mit Android Studio entwickeln 78

NEUES AUS DEM NETZ

Underwater Glove

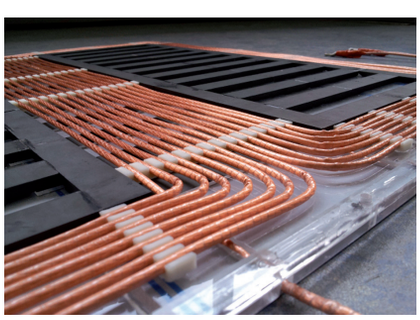
Wissenschaftler des Empowerment Informatics Programms der Tsukuba University haben einen innovativen Sonarhandschuh entwickelt, mit dem es möglich ist, tief unter Wasser Objekte zu erfühlen, auch wenn man noch hunderte Meter von diesen entfernt ist. So könnte man zum Beispiel die gesunkene Titanic erfühlen, ohne dabei kilometerweit auf den Grund des Meeres zu tauchen. Ziel des Sonarhandschuhs ist eben genau dies: Objekte unter Wasser fühlbar zu machen auch wenn man sich noch weit von diesen befindet oder die Sicht unter Wasser einfach zu schlecht ist.
Nähert sich der Träger des Sonarhandschuhs einem Objekt, so wird der Druck in dem Handschuh und speziell in den Fingerspitzen erhöht. Anstelle von Vibrationsmotoren, nutzt der Handschuh direkt das Wasser und pumpt dieses je nach Entfernung zu umliegenden Objekten, mittels kleiner Motoren die mit einem Arduino Pro Mini verbunden sind, in die Fingerspitzen. So soll die Unterwasserwelt, hochauflösend fühlbar werden. Vergleichbar mit den Sensoren von Haien oder Delfinen.
Quelle: http://www.trendsderzukunft.de/underwater-glove-erfuehle-die-unterwasserwelt-mit-einem-sonarhandschuh/2016/01/15/



Schnurloses Aufladen – Vom Tablet bis zum Strom-Highway

Hohe Wirkungsgrade dank moderner Leistungselektronik – Hersteller von Smartphones und Elektroautos setzen auf Laden ohne Kabel

Hamburg – Ladekabel ade: Das gilt längst nicht mehr nur für die elektrische Zahnbürste. Immer mehr Smartphones bekommen ihren Strom über eine Ladeplatte. Einfaches Ablegen ohne Kabelgewirr reicht aus. Es wird nicht mehr lange dauern, bis auch Kameras und Tablets schnurlose Ladestellen integriert haben. Rasant entwickelt sich auch das schnurlose Aufladen für Elektromobile. Schafft es die Industrie, sich bei den neuen Technologien schnell auf einen gemeinsamen Standard zu einigen, könnte von vornherein das Netz an Stromtankstellen auf der eleganteren Schnurlostechnik aufbauen - und so der derzeit sich nur zäh verbreitenden Elektromobilität einen weiteren Schub geben.
Auf den ersten Blick ist es erstaunlich, dass schnurloses Aufladen nicht längst gängiger ist. Bereits zu Beginn des 20. Jahrhunderts demonstrierte Nikola Tesla, wie sich Strom über



Induktion zwischen zwei Spulen übertragen lässt. Wie in einem Trafo floss ein Wechselstrom durch die erste Spule und erzeugte ein magnetisches Wechselfeld. Wirkte dies auf eine zweite, benachbarte Spule, zirkulierte in dieser wieder Wechselstrom. „Auch in elektrischen Zahnbürsten wirkt dieses klassische Trafoprinzip“, sagt Bernd Eckart, Experte für schnurlose Stromübertragung am Fraunhofer-Institut für Integrierte Systeme und Bauelementetechnologie IISB in Erlangen. Doch lange Ladezeiten, Verluste durch Abwärme und geringe Wirkungsgrade waren die Gründe dafür, dass das Aufladen über Induktion lange Jahre nur wenige Anwendungen fand.
„Der Sprung aus dieser Nische wurde durch bessere Leistungselektronik in den vergangenen Jahren möglich“, sagt Eckart. Dank dieser Schaltkreise mit Materialien wie Siliziumcarbid und Galliumnitrid wird schnurloses Aufladen über die resonante induktive Kopplung zunehmend leistungsfähiger. Wird der Zahnbürstenakku noch immer mit der langsamen Netzfrequenz von 50 Hertz geladen, schnellen in modernen Ladestationen die Übertragungsfrequenzen auf über 200 Kilohertz hoch. Mit den häufigen Wechseln von elektrischen und magnetischen Feldern verringern sich die Ladezeiten auf wenige Stunden mit steigenden Wirkungsgraden von deutlich über 90 Prozent.
Quelle: http://www.wissenschaft-aktuell.de/artikel/Schnurloses_Aufladen___Vom_Tablet_bis_zum_Strom_Highway1771015590051.html

Surface Book nun auch in Deutschland und Österreich verfügbar

Ultradünn, präzise Fertigung und ein brillanter Bildschirm: Der ultimative Laptop.
Surface Book lässt sich in drei verschiedene Varianten verwandeln und passt sich damit perfekt Ihrem Arbeitsstil und Ihrer Umgebung an.
Surface Book ist konzipiert für kreatives Arbeiten. Seine Leistung verdankt es einem Intel® Core™ i5- oder i7-Prozessor der 6. Generation, bis zu 16 GB Arbeitsspeicher und einem optionalen Grafikchip.
Seit Februar dieses Jahres ist das Surface Book nun auch in Deutschland und Österreich erhältlich.

Quelle: https://www.microsoft.com/surface/de-de/devices/surface-book?ocid=OCTEVENT_MSCOM

Samsung zeigt Innovationen tragbarer Technologie

Handsteuerung für Virtual Reality
Eine Innovation ist „rink“. Dabei handelt es sich um eine neue Handbewegungssteuerung für mobile Virtual Reality Geräte. Die Anwender sollen so mit ihren bloßen Händen noch tiefer und kontrollierter in die virtuelle Welt der jeweiligen Anwendung oder eines Spiels eintauchen können.

Ans Ohr tippen zum Telefonieren
Einfach einen Finger ans Ohr drücken anstatt aufwändig ein Headset anzustecken oder Kopfhörer aufzusetzen: Diese Idee steckt hinter „Tip Talk“. So wird die Verständlichkeit von Anrufen erhöht. Tip Talk soll sich auch für den Einsatz in lauten, gesprächigen Umgebungen eignen.

Quelle: http://www.technikneuheiten.com/samsung-zeigt-innovationen-tragbarer-technologie/#prettyPhoto

Microsoft setzt mit Partnern zum Sprung ins vernetzte Auto an

Microsoft will mit einer Reihe von Partnern künftig eine wichtige Rolle bei der Gestaltung des vernetzten Autos spielen.
Gemeinsam mit der VW-Tochter IAV zeigt der Softwarehersteller auf der CES in Las Vegas, wie sich Anwendungen aus Windows 10 über das Smartphone direkt über das Display eines Fahrzeugs nutzen lassen.
Der Autofahrer soll damit während der Fahrt etwa auf Musik oder Anwendungen wie den Kalender oder Outlook zugreifen können. Dabei steht ihm auch die Sprachassistentin Cortana zur Verfügung. Die Integration solle es ermöglichen, diejenigen Geräte mit dem vernetzten Auto zu verbinden, die ohnehin schon genutzt würden, teilte Microsoft am Dienstag mit.
Volvo Cars zeigt gemeinsam mit Microsoft in Las Vegas, wie sich über eine Volvo-App auf Microsofts Fitness-Armband per Spracheingabe schon unterwegs die Heizung im Auto ferngesteuert starten lässt. Seine Bürosoftware Office 365 soll zudem in das Infotainment-System für vernetzte Autos des Audiospezialisten Harman integriert werden.

«Das Auto wird nicht länger nur Mittel zum Zweck sein, um uns von A nach B zu bringen, sondern integraler Bestandteil unseres digitalen Lebens», sagte Alastair Bruce, Geschäftsführer von Microsoft Deutschland.
Quelle: https://news.microsoft.com/de-de/ces2016_connectedcars/

NO SPECTATORS, ONLY PARTICIPANTS!

Ein Wochenende mit anderen Interessierten zu den neusten Trends in der mobilen Welt austauschen, so kann man das MobileCamp Dresden kurz zusammenfassen. Im Mai ist es wieder so weit, wenn sich Entwickler aus der DACH-Region im Osten Deutschlands zusammenfinden.

Wir schreiben das Jahr 2003: Tim O'Reilly veranstaltet eine sogenannte Unkonferenz, um sich mit seinen Gästen, die eine extra Einladung zu dieser Veranstaltung erhielten, auszutauschen. Er nannte das ganze Friends of O'Reilly Camp oder kurz FooCamp, und legte damit den Grundstein für die BarCamps, wie wir sie heute kennen. Einzelne Teilnehmer dieses FooCamps fanden die Idee so gut, dass sie kurzer Hand ein eigenes Camp in den Büroräumen von Socialtext organisierten - das erste BarCamp. Der Unterschied zum FooCamp: Es konnte jeder daran teilnehmen ohne separate Einladung.

Anders als bei herkömmlichen Konferenzen steht bei einem BarCamp zu Beginn noch keine Agenda fest. Diese entwickelt sich an dem jeweiligen Tag. Alle Teilnehmer begegnen sich auf einer Augenhöhe und sind eingeladen, die Tagesordnung gemeinsam mit eigenen Ideen zu füllen. Je nach thematischem Schwerpunkt entwickeln sich auch Veranstaltungen, die gegebenenfalls bereits am Ende konkrete Ergebnisse vorweisen können. Vorrangig dienen BarCamps zum Austausch und zur Diskussion. Die meisten BarCamps setzen auf die Unterstützung von Sponsoren, damit sie die Veranstaltung ohne Eintrittsgelder finanzieren können. Somit haben auch Personen aus finanziell schwachen Verhältnissen, die Möglichkeit an dem Wissensaustausch teilzunehmen und sich einzubringen.

Im Herbst 2006 schwappten dann auch die Barcamps in den deutschsprachigen Raum. Mittlerweile gibt es um die 150 Unkonferenzen in der DACH-Region. Fast jeden Monat findet eine Unkonferenz in Deutschland statt.

Gemeinsam wird über die Agenda abgestimmt. Jeder kann seine Ideen und Vorträge einbringen. Die Teilnehmer entscheiden durch Handzeichen, ob ein Thema interessant ist und es auf den Tagesplan schafft.

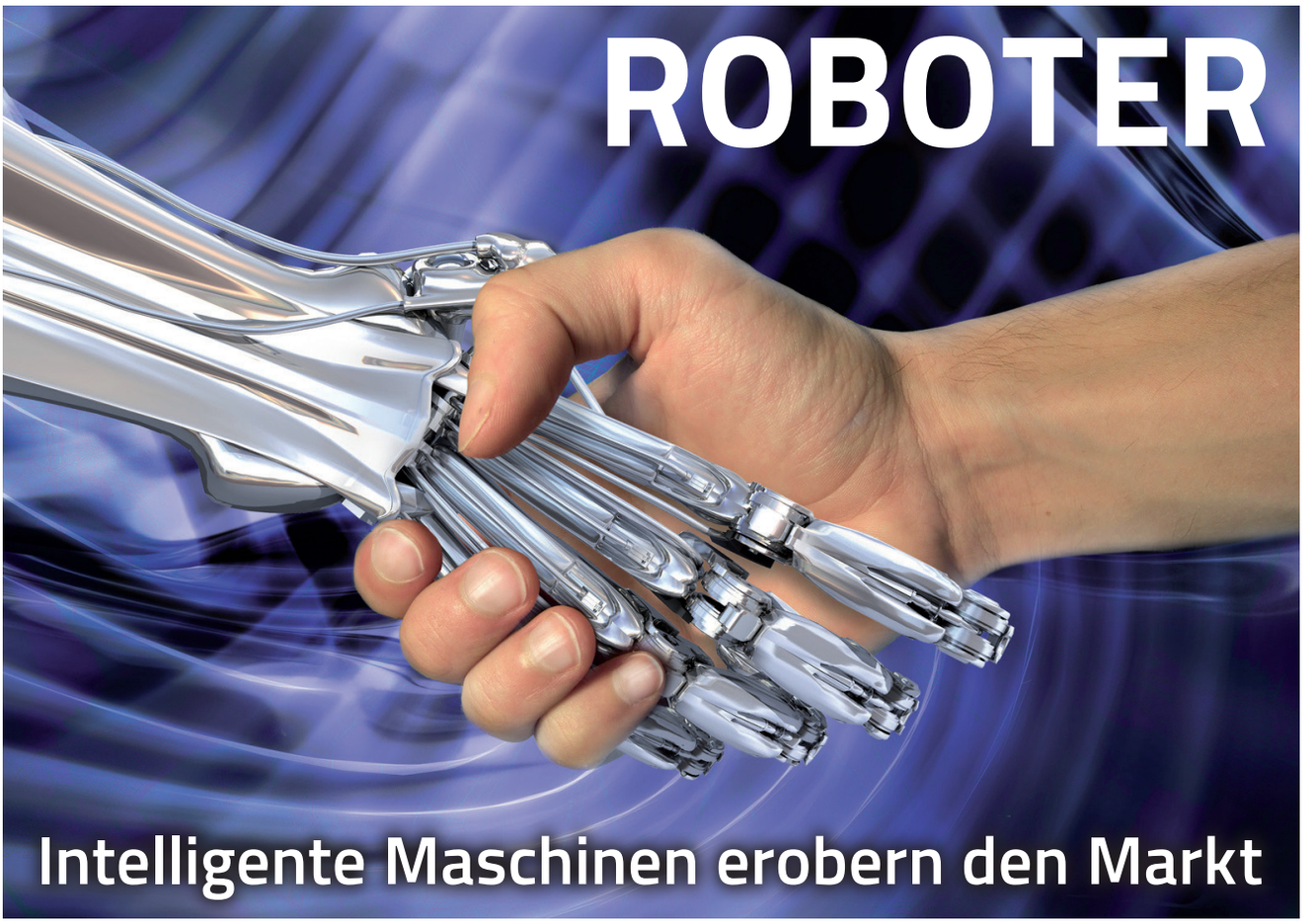
MOBILE+ BARCAMP = MOBILECAMP

Auch für Mobile-Entwickler gibt es natürlich BarCamps. So zum Beispiel das MobileCamp in Dresden. Mit knapp 300 Teilnehmern jährlich gehört es zu den größten Barcamps im deutschsprachigen Raum zum Thema „Mobile Everything". Hinter dem Motto versteckt sich fast alles – von Mobile Development, Mobile Advertising bis hin zu Wearabels und IOT.

Wie bei jedem BarCamp bestimmen die Teilnehmer die einzelnen Sessions. Daher gilt auch beim MobileCamp: „No spectators, only participants!". Seit 2013 organisiert der MobileCamp e.V. die Veranstaltung und kümmert sich ehrenamtlich darum, dass Sponsoren gefunden werden und alles wie gewohnt abläuft.

In diesem Jahr findet am 21. und 22. Mai zum achten Mal das MobileCamp in Dresden statt. Wie die letzten Jahre stellt auch 2016 die TU Dresden die Räume der Informatikfakultät zur Verfügung, um sich das Wochenende entspannt zu den neuesten Themen auszutauschen. Das Ganze wird, wie auch in den letzten Jahren, dank Sponsoren für die Teilnehmer kostenfrei sein.

MobileCamp Dresden
Termin: 21. – 22. Mai 2016 in Dresden
Ort: Nöthnitzer Strasse 46, 01187 Dresden
Teilnehmer: ca. 300 (Innovatoren, Early Adoptor, Multiplikatoren, Gründer, Programmierer, Anwender)
Themen: Alles rund um das Thema „Mobile". Von der Entwicklung bis zur Vermarktung und darüber hinaus.



Robotik und künstliche Intelligenz

Jeder kennt den Begriff Roboter und jeder hat sicherlich im Fernsehen oder auch mittlerweile im echten Leben den ein oder anderen gesehen, jedenfalls eine Art von Roboter. Mittlerweile existieren schon in den kleinsten Ausfertigungen diese kleinen intelligenten Maschinen, die einen putzen und wischen und die anderen mähen den Rasen. Schon vor Jahrzehnten hat man intelligente Maschinen und künstliche Intelligenz in Büchern beschrieben und in Filmen gezeigt, zumeist als Science-Fiction.

Die bekanntesten Beispiele hierfür aus der Star Wars Saga sind die zwei geselligen Droiden R2D2, ein Astromech-Droide und C3PO, welcher zu den humanoiden Droiden gehört. In unglaublich vielen Bereichen werden seit mehr als 40 Jahren Roboter verschiedenster Arten entwickelt und eingesetzt. Hierzu wird im Laufe des Artikels näher eingegangen, zuerst wäre es aber sicherlich interessant die Herkunft und Bedeutung von Robotern und der Robotertechnik näher zu beachten.

URSPRUNG UND BEDEUTUNG DER ROBOTIK

Schon im Jahre 340 v. Christus philosophierte Aristoteles bei seinen Forschungen der Wissenschaftstheorien und Physik über selbstständig arbeitende Werkzeuge. Das folgende Zitat von ihm bildet die heutige Realität ab:

„Wenn jedes Werkzeug auf Geheiß, oder auch vorausahnend, das ihm zukommende Werk verrichten könnte, wie des Dädalus Kunstwerke sich von selbst bewegten oder die Dreifüße des Hephästos aus eignem Antrieb an die heilige Arbeit gingen, wenn so die Weberschiffe von selbst webten, so bedarf es weder für den Werkmeister der Gehilfen noch für die Herren der Sklaven“ *Aristoteles, 339 v. Chr., Athen*

Sicherlich konnte sich Aristoteles noch keine genauen und eindeutigen Vorstellungen von Robotern und künstlicher Intelligenz machen, selbst in der heutigen Zeit sollten immer noch nicht alle Möglichkeiten und Technologien vollständig erforscht und entwickelt sein. Dies bildet einen fortlaufenden Prozess und ist auch stark von der Entwicklung der Menschheit abhängig. Heute sind sich Wissenschaftlicher einig, dass unsere Vorfahren in der Steinzeit und selbst bis ins frühe Mittelalter nicht den geistigen Horizont erbringen konnten, um derartige Entwicklungen und Innovationen vorantreiben zu können. Man dürfte dann relativ sicher davon ausgehen, dass selbst wir noch einen erheblichen zeitlichen Sprung benötigen um weitere bedeutende und innovative Erfindungen herbeiführen zu können.

Im Jahr 1920 hat der Schriftsteller Karel Capek den Begriff Roboter zum ersten Mal für einen Androiden in seinem Schauspiel „Rossums Universal Robots, kurz R.U.R benutzt und damit auch eingeführt. In seinem

Stück ging es darum, dass ein Unternehmen künstliche Menschen erzeugt und diese als billige Arbeiter und ohne jeglichen menschlichen Rechts verkauft. Im Verlaufe der Geschichte wird durch den Einsatz dieser Roboter die gesamte Welt in Aufruhr gesetzt und schließlich vernichten die künstlich erschaffenen Maschinen die Menschheit.

Sicherlich hat man zwischen den 1920er und den 1940er Jahren ganz viel versucht und entwickelt, jedoch erst im Jahr 1947 als der Transistor erfunden wurde kam ein regelrechter Aufschwung und es gelangen große Fortschritte. Schon 13 Jahre später im Jahr 1960 wurde der erste Industrieroboter vorgestellt, welcher hydraulisch betrieben und für das Schweißen von Widerstandspunkten verwendet wurde. Seit diesem Zeitpunkt ging es mit der Kurve und Komplexität der Robotertechnik stark nach oben. Bereits in Asien wurden diese hydraulischen Roboter ab Mitte der 70iger Jahre eingesetzt. In Deutschland setzte Mercedes Benz ab dem Jahre 1970 solche Maschinen in der Produktion schon sehr erfolgreich ein.

Für die internationale Wirtschaft und Produktion bedeutete die Entwicklung und Etablierung solcher Maschinen eine enorme Produktivitätssteigerung. Internationale Unternehmen und ganze Länder verbesserten ihren Durchsatz zwischen 40 – 70 %, da teilweise ganze Hallen mit diesen technischen Hilfsarbeitern ausgestattet wurden und dadurch die Zeit deutlich verkürzt wurde um ein Produkt oder Paket herstellen zu können.

STRUKTUR UND AUFBAU VON INDUSTRIEROBOTERN

Unendlich viele Einsatzzwecke wurden für Maschinen gefunden, neben dem Montieren, palettieren, stapeln, verpacken oder auch das Sägen wurden solche auch in der Lackierung sowie in der Messtechnik eingesetzt. Grundsätzlich ähneln sich der Aufbau und die Struktur solche Industrieroboter immer sehr.
Mit der Entwicklung des weltweit ersten Industrieroboters durch den Robotikpionier KUKA im Jahre 1973 begann der Einstieg in die Kombinierung der Elektrotechnik und Mechanik. Dieser bestand aus sechs elektromechanisch angetriebenen Achsen und trug den Namen „Famulus“, welcher vor allem in der Automobilindustrie zum Einsatz kam, da viele Unternehmen mehr Flexibilität in der Herstellung und Produktion gefordert hatten.

Die Aufbaustruktur eines Industrieroboters setzt sich grundlegend aus folgenden Bereichen zusammen:

- ***Steuerung oder auch Programmierung***

Dieser Bereich enthält sämtliche Logik und Definitionen wie sich der Roboter verhält und agiert. Sämtliche Bewegungen und Interaktionen werden durch die Logik bestimmt und werden an den beweglichen Antrieb weitergeleitet.

- ***Antrieb***

Dieser Bereich enthält die Regelung und den Motor. In Zusammenhang mit dem Getriebe werden so die Bewegungen durch elektrische Impulse oder durch Hydraulik durchgeführt.

- ***Controlling oder Sensorik***

Dieser Bereich enthält die Logik welche Haltepunkte und Bewegungsmustern definiert. Mithilfe von Mustern oder Funktionen liefert das Controlling die Information über die aktuelle Position und kann somit bestimmen was zwischen IST – und SOLL-Wert noch zu tun ist.

- ***Kinematik***

Dieser Bereich ist für die physische Bewegung von Punkten und Teilen zuständig. Durch die Beschreibung der Größen Geschwindigkeit, Position und Beschleunigung in Kombination mit einem Koordinatensystem wird dies festgelegt.

- ***Greifsystem oder auch Gelenkvorrichtung***

Dieser Bereich beinhaltet, vor allem aus der Elektrotechnik und Mechanik enorm viele Funktionen und Bausätze. Grundlegend ist das Greifsystem das operative Arbeitssystem eines Industrieroboters in der Produktion. Vorstellen kann man sich diesen wie ein Armgelenk.

FORMEN UND VERFAHREN DER ROBOTER-PROGRAMMIERUNG

Man unterscheidet in der Industrieroboter-Programmierung zwischen der „Online-Programmierung“ und „Offline-Programmierung“.

Die Online-Programmierung erzwingt den direkten Zugang zum Roboter, da diese direkt an der Maschine selbst durchgeführt werden. Am bekanntesten dürfte hier die „Teach-Methode“ sein, wodurch der Entwickler dem Roboter in die definierten Positionen bringt und die gesamten Produktions – und Arbeitsabläufe manuell mit ihm durchspielt. Die notwendigen Informationen z.B. Koordination, Bewegungen etc. speichert der Roboter in der Steuerung. Dies wird solange durchgeführt bis der Roboter den Produktionsablauf selbstständig durchführen kann.

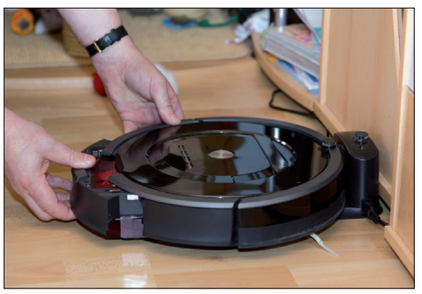
Die Offline-Programmierung, welche sich dadurch auszeichnet, dass diese meist textuell in Form von Programmcode erzeugt wird und sich relativ schnell an individuelle Anforderungen anpassen lässt.

Jeder Entwickler und Programmierer kennt sicherlich die Herangehensweise und Strukturierung von Programmcode und Skripten. Selbstverständlich gibt es hierzu eine große Anzahl an Programmiersprachen mit unterschiedlichen Aufbau und Syntax. Auch in der Roboterprogrammierung gibt es sehr viele verschiedener Programmiersprachen und sogar vereinzelte, welche nur auf bestimmte Robotertypen ausgerichtet sind. Die Herangehensweisen und die Verfahren innerhalb der Roboterentwicklung erleben einen stetigen Wandel. In dem

Bei der Methode der Virtual Reality schlüpft der Entwickler in die „virtuelle Rolle" des Roboters. Meistens mit Hilfe einer 3D-Brille oder eines elektronischen Handschuhes, welcher dazu dient die Steuerung des Roboters direkt selbst durchführen zu können.

schnelllebigen Technologiewandel kommen immer mehr, auch innovative Methoden zum Einsatz z.B. die sensorgestützte Programmierung und die aus Science-Fiction bekannte Virtual Reality.

Bei der sensorgestützten Programmierung ist ein Roboter z.B. in der Lage selbstständig auf Ereignisse zu reagieren und daraufhin einen Ablauf durchzuführen. Diese kommen vor allem in der Produktion zum Einsatz. Ein Szenario welches verbreitet ist, stellt sich so dar, dass der Roboter an einem Bauteil etwa eine raue Kante oder eine Unebenheit entdeckt und diese dann anschließend sofort entfernt oder optimiert.



ROBOTIK IM CONSUMER-BEREICH

Selbst unser alltägliches Leben kann und soll durch kleine Haushaltsroboter erheblich erleichtert werden, so dass wir weniger Energie für haushaltsnahe Arbeiten leisten müssen und uns so auf effektivere Themen fokussieren können. Bereits seit Anfang der 2000er existieren solche Maschinen welche als Serviceroboter bezeichnet werden.

Den Anfang haben die Staubsaugerroboter gemacht, welche meist rundförmig und einen Radius von 28 – 30 cm aufweisen. Diese kleinen Helfer gibt es in etlichen Ausführungen und Varianten. Die Einstiegsmodelle welche für den Konsumenten in einem bezahlbaren Rahmen bleiben sind mit einem einfachen Kollisionssensor ausgestattet und tasten sich somit von Hindernis zu Hindernis. Sobald dieser eine Wand oder einen Gegenstand über den Sensor wahr nimmt, ändert der Roboter seine Richtung und macht seine Arbeit weiter. Zumeist müssen diese Modelle auch manuell geleert und aufgeladen werden. Die erweiterte Stufe des Staubsaugerroboters zeigt sich dadurch, dass diese z.B. mit Ultraschall, Infrarot oder Lasersensoren ausgestattet sind um die Entfernung zu einer Wand oder einem Gegenstand zu messen. Weiterhin gibt es auch schon diverse Helfer dieser Art, welche mit Absturzsensoren (Schutz vor Treppen und Absätzen) ausgestattet sind und durch einen integrierten Speicher den Raum komplett scannen, daraus einen Plan erstellen und so den Saugvorgang beschleunigen. Die absoluten Favoriten sind die völlig autonomen Roboterstaubsauger, welche einen festen Plan haben, zumeist über mehrere Tage hinweg und völlig alleine saugen, fahren, sich entleeren und aufladen. Solche Ausfertigungen liegen dann preislich aber dann schon im kleinen vierstelligen Bereich.

Mittlerweile wurden gemäß der International Federation of Robotics schon mehrere Millionen Staubsaugerroboter weltweit verkauft. Zumeist lässt die Qualität gegenüber klassischen Bodenstaubsaugern nach Stiftung Warentest jedoch aber noch sehr zu wünschen übrig, so dass also für optimale Ergebnisse immer noch selbst der Gang durch die Räume nötig ist.

Seit Mitte der 1990iger Jahre gibt es die Rasenmähroboter, welcher eben-falls zu den Servicerobotern gehört und selbstständig in der Lage ist, den Rasen zu mähen. Diese Art von Maschinen kann man auch als Vorgänger zu den Staubsaugerrobotern sehen.

Die Rasenmähroboter sind überwiegend mit Batterien bzw. Akku-Packs betrieben und wenden das Prinzip des Mulchmähens an. Diese Methode bezeichnet das direkte Aufbringen von gehäckselten Gras-Gut auf der Rasenfläche direkt nach dem Rasenmähen. Bekannt ist dieses Verfahren vor allem in der Landwirtschaft, da dadurch die Nährstoffversorgung des Bodens optimiert werden kann. Der Roboter mäht das Gras so fein, das dieses auf der Grasfläche verbleibt und anschließend auf natürliche weiße verrottet um somit den Kreislauf des Nährstoffaustausches zu beschleunigen.

ROBOTIK IM MILITÄRISCHEN BEREICH

Im militärischen Umfeld sind Roboter oder in diesem Fall auch unbemannte militärische Systeme genannt, vor allem in den Teilgebieten Spionage, Aufklärung oder auch der Zielbekämpfung im Einsatz.

Vor allem in Aufklärungseinsätzen nehmen diverse luftgetriebene Drohnen Bild – und Videoaufnahmen von den kritischen Gebieten auf und können diese direkt drahtlos an die operativen Zentralen übersenden. Diese werden dann durch Computer und Fachpersonal ausgewertet.

Weiterhin sind Minenräumfahrzeuge, welche vollständig autonom agieren können seit einigen Jahren in vielen Ländern im Einsatz. Diese werden entweder mit einer festen Logik programmiert und dann ausgesetzt oder per Fernsteuerung gelenkt. Auch kommt meist eine Kombination aus beiden Varianten zum Einsatz um den größtmöglichen Mehrwert dafür zu schaffen.

Mit einer der bekanntesten solcher Maschinen ist der Talon, welches ein ferngesteuertes Robotersystem darstellt, das primär zur Entschärfung von Sprengstoffen eingesetzt wird. Dieser wiegt je nach Ausstattung zwischen 30 bis 50 kg und wurde bereits am Ground Zero eingesetzt.

Die Überlegungen fortschrittlichere Maschinen zu entwickeln um das Militär technologisch auch voranzubringen, lassen sich jedoch nicht so einfach wie in den Bereichen Haushalt oder Industrie realisieren, da vor allem auch Themen wie Ethik und Moral eine tragende Rolle dabei spielen.

ROBOTIK IN DER FREIZEIT

Selbstverständlich schreckt die Robotik auch nicht davor zurück für die Bereiche Freizeit und Hobby etwas zu bieten und dies seit einiger Zeit schon sehr erfolgreich.

Mittlerweile sind die sogenannten Drohnen, Quadrocopter und auch Multicopter in aller Munde und verbreiten sich sehr stark in allen Altersgruppen. Diese Art von Robotern oder teilweise auch noch als Computer definiert, sind umbenannte Luftfahrzeuge welche über eine Steuerung am Boden betrieben werden. Das Steuern von solchen Drohnen im privaten Bereich fällt rechtlich gesehen unter dem Punkt „Betreiben von Modellflugzeugen“, was sich schon einige Jahre etabliert hat.

Vor allem begeisterte Bastler und Jugendliche erfreuen sich sehr an den leicht zu bedienenden Flugmodellen, welche unkompliziert gewartet und auch erweitert werden können. Neuere Modelle lassen sich mittlerweile vom Smartphone aus über eine dafür vorgesehene App steuern und navigieren. Das Material entspricht, außer der Elektronik und Propellern zu 90% aus Styropor, da dieses vor allem sehr leicht ist und einfach zu handhaben ist.

Seit einiger Zeit wird diese Art von unbemannten Luftfahrzeugen ebenfalls im Marketingbereich, in der Werbung aber auch in der Logistik verwendet. Vor allem aber die Medien nutzen die Drohnen gerne um Bild – sowie Videomaterial von Orten aus der Luft aufzunehmen, da die Technologien mittlerweile auch soweit ausgebaut sind, dass Kameras und drahtlose Übertragungsvorrichtungen unkompliziert an Drohnen jeglicher Bauweise angebracht werden können.

ROBOTER IN DER ZUKUNFT

Was es seit einigen Jahren auf der Welt an Technologien und nun jetzt auch an Robotern gibt ist schon sehr faszinierend, auch wird es im alltäglichen Leben niemals langweilig, da fast täglich Neuerungen gezeigt und gefeiert werden.

Was aber genauso interessant oder fast noch viel interessanter ist, wäre die Weiterentwicklung dieser Thematik und wie sich Roboter in Zukunft als Unterstützung für die Menschheit einsetzen lassen können.

Aktuell wird in fast allen möglichen Bereichen geforscht und entwickelt, die größte Hürde, deren Überwindung mit großen Druck vorangetrieben wird, ist Robotern die Sprache des Menschen beizubringen. Darunterfallen aber auch das Deuten oder verständlich machen bis hin zur wahren künstlichen Intelligenz, so dass Roboter eigene Entscheidungen fällen können und z.B. Reflexe besitzen.

Der Mensch agiert klassisch nach dem Prinzip sehen – verarbeiten – lernen – machen –. Genau dieses Prinzip möchte man auch Robotern beibringen, so dass diese selbstständig Informationen aufnehmen, verarbeiten und dann in Zusammenhang mit anderen Informationen bringen können. Dieses Forschungsfeld ist jedoch mit das komplexeste in der gesamten Robotik.

Wie schon zu Beginn des Artikels beschrieben, existieren in der Science-Fiction sowie in der Phantasie-Welt etliche Möglichkeiten und Ebenen von potentiellen Robotern und Einsatzmöglichkeiten. Ob diese aber irgendwann zur Realität werden können, das ist noch die große Frage. Sicherlich wird es nicht langweilig und man darf gespannt sein, welche innovativen Erfindungen und Weiterentwicklungen in den Bereichen Elektronik und Robotik auf die Menschheit zukommen werden.

BENJAMIN LANZENDÖRFER ist seit einigen Jahren im Microsoft-Umfeld tätig und verbindet damit eine große Leidenschaft. Als Administrator, Consultant und Projektleiter hat er schon viele Projekte rund um SharePoint, Office und Exchange begleiten dürfen.

Neuigkeiten von DreamSpark

DreamSpark bietet sowohl für Lernende als auch Lehrende kostenfreie Software an

Software ist aus dem Lernalltag nicht mehr wegzudenken. Ob zur Organisation von Wissen, Schreiben von Hausarbeiten, zur Kommunikation oder um den Lehrbetrieb aufrechtzuerhalten. Wer auf kostenpflichtige Lizenzen angewiesen ist, muss dafür oft tief in die Tasche greifen. Microsoft DreamSpark erlaubt es Lernenden, Lehrenden und Instituten, kostenfreie Software von Microsoft einzusetzen. Und zu diesem Programm gehört mehr, als vielen bewusst ist.

Während eines Studiums kommt eine ganze Reihe von Softwareprodukten zum Einsatz. Vom Betriebssystem, über E-Mail-Programme und Office Pakete ist alles vertreten. Dann vielleicht noch Tools zur Organisation von Terminen und Wissen, zur Kommunikation und vieles weitere.

Diese Anforderungen unterscheiden sich unter Umständen stark von Fachbereich zu Fachbereich. Während Studenten und Studentinnen eines nicht technischen Studiengangs mit der Aufzählung oben schon sehr glücklich sind, fehlen für ein MINT-Studium (Mathematik, Informatik, Naturwissenschaften und Technik) in der Regel noch eine ganze Menge weiterer Produkte. Entwicklungsumgebungen und Datenbanken sind nur zwei Beispiele für diese fehlenden Produkte.

Und damit sind bisher nur die Anforderungen eines Studiums abgedeckt, denn in Schulen ist der Einsatz von Technologie ebenfalls schon länger Alltag. Auch dort werden Softwareprodukte beispielsweise zum Lernen und zum Erstellen von Hausaufgaben genutzt.

Der Artikel beleuchtet das aktuelle Angebot von Microsoft DreamSpark. Aus eigener Erfahrung mehr aus der Sichtweise eines MINT-Studiums.

DREAMSPARK WAS?

Wer für den Einsatz in Schule oder Studium auf Software von Microsoft setzen möchte oder vielleicht sogar muss, weil es Anforderungen vom Fachbereich gibt, der bleibt schnell auf hohen Kosten sitzen beziehungsweise muss sich im Zweifel gegen ebendiesen Einsatz entscheiden.

Erfreulicherweise ist Microsoft DreamSpark [1] an immer mehr Institutionen vorhanden, so dass viele Lizenzen kostenfrei zur Verfügung stehen.

Hinter dem Namen DreamSpark, vormals Microsoft Developer Network Academic Alliance (MSDNAA), steckt ein Programm von Microsoft, das den Zugriff auf Microsoft-Software für den Unterricht, die Lehre und Forschung, sowie für technische Bildung im Allgemeinen, ermöglicht.

Dieses Angebot gilt für Schüler, Studenten, Lehrende und akademische Einrichtungen. Letztere können bestimmte Abonnements abschließen, um allen Lernenden gleichermaßen Softwareprodukte von Microsoft kostenfrei zur Verfügung zu stellen.

DREAMSPARK (STANDARD) VS. DREAMSPARK PREMIUM

Schüler und Studenten können recht einfach an dem Programm teilnehmen. Notwendig sind nur ein eigenes Konto, das selbstständig erstellt werden kann, sowie die Verifizierung des eigenen Studentenstatus. Anschließend kann der Download beginnen. Den Studentenstatus zu belegen ist ebenfalls sehr einfach. Beispielsweise über die Domäne der eigenen E-Mail-Adresse (Schule oder Hochschule) oder über eine International Student Identity Card (ISIC)-Karte.

Lehrende erhalten Zugriff über das Abonnement der eigenen Einrichtung. Akademische Einrichtungen haben dabei die Wahl zwischen zwei DreamSpark-Programmen beziehungsweise Ausführungen. DreamSpark Standard gilt für alle Einrichtungen von Grundschul- bis Universitätsausbildung. Darin enthalten sich grundlegende Softwareprodukte für diese Einrichtungen. DreamSpark Premium enthält einen umfassenderen Softwarekatalog mit mehr als 500 Produkten. Allerdings gilt dieses Programm nur für berechtigte technische Institute. Microsoft bezeichnet das als sogenannte STEM-Institute. Ein Begriff, der aus dem Englischen kommt, und die Felder Science, Technology, Engineering und Mathematics enthält. Damit entsprechen STEM-Institute den MINT-Instituten hier in Deutschland. Eine Übersicht über die Berechtigungen ist auf der DreamSpark Seite unter Abonnement [2] zu finden. Eine Liste der konkret berechtigten Institute beziehungsweise Fachbereiche ist in der STEM-Definition [3] des DreamSpark Angebots zu finden.

INHALTE VON DREAMSPARK

Die Inhalte des DreamSpark Programms werden von Zeit zu Zeit angepasst und erweitert. Gerade in der jüngsten Vergangenheit gab es einige interessante Anpassungen, insbesondere in Richtung Azure.

Bei DreamSpark Standard sind beispielsweise die Produkte für Windows Server, Entwicklertools wie Visual Studio Community, Professional und Embedded, OneNote und der SQL-Server enthalten. Zum Premium-Angebot von DreamSpark gehören dann auch noch die Client-Betriebssysteme wie Windows 7 bis Windows 10 dazu, die Enterprise Version von Visual Studio, Visio und Project, BizTalk Server und SharePoint Server. Bei beiden DreamSpark Programmen gehören natürlich noch die kostenfreien Produkte wie XNA Game Studio und Microsoft Small Basic mit zum Angebot, um nur zwei Beispiele zu nennen. Diese Produkte stehen allerdings auch so zum Download zur Verfügung und werden nur aus Komfortgründen in einem Portal gebündelt.

Vor einiger Zeit hat dann auch Microsoft Azure in DreamSpark Einzug [4] gehalten. Die Zusammenarbeit erfolgt mit Visual Studio Community 2015 in Kombination mit einigen Azure Angeboten, die Nutzer von DreamSpark kostenfrei zur Verfügung stehen. Visual Studio 2013 ist mit dem Angebot nicht kompatibel. Im Paket enthalten sind unter anderem die Funktionen:

- Web Apps
- MySQL DB
- Visual Studio Online
- Application Insights

Mittlerweile stehen auch Notification Hubs, Azure Mobile Apps und SQL-Datenbanken zur Verfügung.

Auch im Bereich Entwicklung für mobile Geräte hat sich etwas getan. Xamarin für Students ist ebenfalls in DreamSpark enthalten und erlaubt die Zusammenarbeit von Xamarin und Visual Studio Community 2015.

Eine gute Quelle für Informationen rund um DreamSpark und Änderungen an den enthaltenen Softwarepaketen bietet ein RSS-Feed [5], der auch auf Deutsch zur Verfügung steht. Dort werden größere Anpassungen und neue sowie geänderte Veröffentlichungen bekannt gegeben.

WEITERE ANGEBOTE FÜR STUDENTEN

Microsoft ist bei weiten nicht der einzige Anbieter, der kostenfreie Tools, vor allem in Richtung Entwickler-Community, zur Verfügung stellt.

GitHub bietet unter dem Sammelbegriff GitHub Education [6] einen sogenannten Student Developer Pack [7] an. Darin enthalten sind eine ganze Reihe von Softwareprodukten und andere Dienstleistungen, die kostenfrei zur Verfügung gestellt werden.

KRITIK

In einem Artikel über kostenfreie Angebote von Unternehmen muss aber auch Platz für Kritik sein. Natürlich ist es löblich, dass Lernenden und Lehrenden kostenfreie Software und Dienstleistungen zur Verfügung gestellt werden. Damit lassen sich viele Dinge in der Schule und im Studium erleichtert.

An dieser Stelle sei aber noch einmal explizit darauf hingewiesen: All das geht natürlich auch mit Open Source Software, die kostenfrei zur Verfügung steht. Für die genannten Softwarelösungen und Angebote gibt es oftmals nicht nur eine Alternative. Ob sich Anforderungen aus der Lehre mit Open Source oder mit Closed Source besser lösen lassen, sei einmal dahingestellt. Selbst ein Versuch, diese Frage zu beantworten, würde ganz sicher scheitern. Dafür sind die Anforderungen zu unterschiedliche und liegen bereits festgefahrene Meinungen zu weit auseinander. Aber es muss auch klar sein, dass Unternehmen wie

Microsoft und GitHub, um nur die beiden zu nennen, die auch in diesem Artikel vorkommen, nichts zu verschenken haben und ein klares Ziel mit den oben genannten Angeboten verfolgen. Lernende, Lehrende und ganze Einrichtungen sollen schon so früh wie möglich an deren Produkte gewöhnt werden. Zum einen, weil Produkte wie Windows dann genutzt werden und sich in den Alltag integrieren, zum anderen natürlich für den zukünftigen Arbeitsmarkt, weil eine große Anzahl an Absolventen Erfahrungen mit den genannten Produkten und Dienstleistungen haben.

Und auch hier soll die Frage, ob das gut oder schlecht ist, offenbleiben. Alles hat seine Vor- und Nachteile. Auch für diesen Sachverhalt trifft das zu.

FAZIT

Angebote wie DreamSpark von Microsoft treffen auf eine hohe Akzeptanz, da dadurch viele Softwareprodukte kostenfrei zur Verfügung stehen. Das hilft Lernenden und Lehrenden gleichermaßen. Trotzdem sind diese kostenfreien Angebote auch mit Absichten der Anbieter verbunden, eigene Softwareprodukte im Markt zu etablieren. Ein Umstand, der nicht einfach ignoriert werden sollte.

Ob diese Angebote genutzt werden sollten oder ob Open Source Software eine Alternative ist, muss jeder Lernende, Lehrende und jedes Institut für sich selbst beantworten. Darauf kann ich in diesem Artikel keine konkrete Antwort geben.

LINKS UND QUELLEN

[1]: Webseite des DreamSpark Angebots: https://www.dreamspark.com

[2]: Information zu einem DreamSpark-Abonnement: https://www.dreamspark.com/institution/subscription.aspx

[3]: Bei STEM enthaltene Fachbereiche und Forschungsgebiete: https://www.dreamspark.com/Institution/STEM.aspx

[4]: Informationen zu Microsoft Azure in DreamSpark: https://www.dreamspark.com/Product/Product.aspx?productid=99

[5]: Deutscher RSS-Feed zu Neuerungen bei DreamSpark: https://www.dreamspark.com/RSS/HomePageFeed.aspx?Locale=de-DE

[6]: Webseite des GitHub Education Angebots: https://education.github.com/

[7]: Das Student Developer Pack von GitHub: https://education.github.com/pack

FABIAN DEITELHOFF lebt und arbeitet in Dortmund, der Metropole des Ruhrgebiets. Er studiert derzeit den Masterstudiengang Informatik mit dem Schwerpunkt Biomedizinische Informatik an der Hochschule Bonn-Rhein-Sieg in Sankt Augustin. Seine Schwerpunkte liegen in der Entwicklung von Visual Studio Erweiterungen, der Analyse und Beschreibung von Open Source Frameworks sowie im Rapid Prototyping. Beruflich ist er als freier Autor, unter anderem für Pluralsight, Sprecher und Softwareentwickler im .NET Umfeld tätig. Sie erreichen Ihn über seinen Blog *www.fabiandeitelhoff.de*, per E-Mail unter *Fabian@FabianDeitelhoff.de* oder auf Twitter als *@FDeitelhoff*.

DIE MICROSOFT CLOUD IN DEUTSCHLAND

Eine rechtliche und technische Allianz der T-Systems und Microsoft

Am 11.11.2015 hat Microsoft CEO Satya Nadella verkündet, dass es ab Mitte 2016 ausgewählte Microsoft Cloudservices auch aus Rechenzentren in Deutschland geben wird. Ab dem 2. Halbjahr 2016 sollen die neuen Cloud-Dienste stufenweise in Deutschland ausgerollt werden, wobei Microsoft Azure, Office 365 und CRM online dann ausschließlich aus deutschen Rechenzentren der T-Systems heraus betrieben werden sollen. Aber: Liegen wirkliche alle meine Daten im Deutschen Rechenzentrum? Kann ich eine bestehende Umgebung in die Deutsche Cloud migrieren? Und was ist aus rechtlichen Gesichtspunkten zu beachten?

DIE DEUTSCHE MICROSOFT-CLOUD AUS TECHNISCHER SICHT

Nach dem neuen Konzept von Microsoft befinden sich – technisch gesehen – sämtliche Kundendaten und -systeme in den beiden deutschen Rechenzentren Magdeburg (Germany Northeast) und Frankfurt/Main (Germany Central). Zwischen den beiden Rechenzentren in Deutschland erfolgt eine Spiegelung/Datenabgleich, um eine Notfall-Wiederherstellung zu ermöglichen. Zudem besteht zwischen den beiden Rechenzentren ein eigenständiges deutsches, vom öffentlichen Internet getrenntes Datennetzwerk.

ZUGRIFFE AUF DATEN & SYSTEME UND DIE ZUGRIFFSKONTROLLE

Sogenannte „Role Based Access Control (RBAC)"-Tools kontrollieren jeglichen Zugriff auf Kundendaten, während die Mitarbeiter von Microsoft selbst keinerlei administrative Top-Level-Rechte haben, also von sich aus keinen Zugriff auf Kundendaten nehmen können. Selbst im Rahmen von Wartung und Support sowie zur Installation von Software-Updates muss der Zugriff vom deutschen Datentreuhänder zugelassen werden. Dabei wird, basierend auf dem Role Based Access Control-Konzept,

nur Zugriff auf die explizit betroffenen Komponenten wie z.B. betroffene Postfächer, Server oder andere Dienste gewährt. Das Rollen-Konzept regelt dabei auch, ob lesende, teilnehmende oder administrative Rechte er-teilt werden. Im Regelbetrieb hat Microsoft also keinerlei Zugriff auf Kundendaten und die dazu-gehörigen Systeme. Nur im Einzelfall werden zeitlich befristet Rechte gewährt. Die Rechtegewährung nimmt dabei einzig der Datentreuhänder vor, nicht Microsoft selbst. Der Zugriff auf die Kundendaten wird im Bedarfsfall protokolliert und in einem Logbuch festgehalten, auf das Microsoft ebenfalls keinen Zugriff hat. Während eines Zugriffs seitens Microsoft, z.B. im Supportfall, gilt das 4-Augen Prinzip. Es wird also jederzeit von einem Mitarbeiter im Deutschen Rechenzentrum überwacht was genau getan wird und auf welche Daten zugegriffen wird.

DATENVERKEHR IN UND ZWISCHEN DEN RECHENZENTREN

Ein weiterer Sicherheitsaspekt betrifft die Absicherung des Datenverkehrs innerhalb der beiden Rechenzentren und den Datenabgleich zwischen den Rechenzentren. Die komplette Kommunikation ist SSL/TLS verschlüsselt. Die dazu benötigten Zertifikate werden nicht von Microsoft selbst und auch nicht von dem Datentreuhänder T-Systems ausgestellt. Dies geschieht durch einen weiteren unabhängigen Partner. Die Zertifikate mit denen die Kommunikation verschlüsselt wird, werden von der Zertifizierungsstelle D-TRUST der Bundesdruckerei geliefert. Ganz im Sinne "Sicherheit Made in Germany" können Kunden auch ihre eigenen Anwendungen in der Microsoft Deutschland Cloud mit den Zertifikaten der D-TRUST absichern. So wird zusätzlich sichergestellt, dass auch der Datenverkehr innerhalb der Microsoft Cloud in Deutschland sicher vor unbefugten Zugriffen ist.

Anders sieht es mit dem Zugriff der Kunden auf ihre Daten und Systeme aus. Dieser Zugriff geschieht in der Regel weiterhin über das öffentliche Internet. Hier kann das Feature „ExpressRoute" Abhilfe schaffen. „ExpressRoute" ist ein zusätzlicher Service der von Microsoft zusammen mit lokalen Internet Providern angeboten wird. Der Service ermöglicht es, den Datenverkehr vom Firmennetzwerk zu den Microsoft Rechenzentren über explizite, nicht öffentliche Leitungen zu realisieren. Das Angebot steht, abhängig vom lokalen Internet Provider, für den Zugriff auf die Cloud Services von Microsoft zur Verfügung.

LIEGEN WIRKLICH ALLE KUNDENDATEN IM DEUTSCHEN RECHENZENTRUM?

Die Frage, ob wirklich alle Kundendaten sowie sämtliche personenbezogenen Daten ausnahmslos in den deutschen Rechenzentren liegen, kann mit JA beantwortet werden. Im Bereich der Benutzerverwaltung werden die Routinginformationen allerdings voraussichtlich über die Microsoft Rechenzentren weltweit verteilt. Diese dienen aber lediglich der Identifikation, wohin ein Aufruf gelenkt werden muss. Es werden dabei keine personenbezogenen Daten in diesen Indextabellen gespeichert. Die Deutsche Cloud hat demnach auch keinerlei Verbindungen zu anderen Public Cloud Systemen von Microsoft. Daraus ergibt sich u. a. auch, dass es keine gemischten Umgebungen geben kann; es ist

also nicht möglich, einen Teil des Services innerhalb einer Umgebung (z.B. Email-Dienste in der Deutschen Cloud) und andere Services innerhalb einer anderen Umgebung (wie z.B. Dokumentenablage-System in einem anderen Microsoft Rechenzentrum in der Public Cloud von Microsoft) abzulegen. Diese technisch bedingte Einschränkung kann für Bestandskunden, die ihre Systeme und Lösungen in die Deutsche Cloud umziehen wollen, eine Herausforderung darstellen. So wie bei anderen vergleichbaren Szenarien von Microsoft wird es künftig wohl die Möglichkeit einer Migration für Kunden, die bereits Services in Microsoft Rechenzentren nutzen, in die Deutsche Cloud geben. In diesem Fall müssen dann aber ausnahmslos alle Komponenten umgezogen werden. Wie genau so eine Migration ablaufen wird, welcher Aufwand dabei entsteht und ob es während der Migration zu Unterbrechungen und Ausfällen kommen wird, ist dabei noch offen.

ROADMAP UND PREISE

Laut Presseberichten läuft der Betrieb in den beiden Rechenzentren bereits mit den ersten Beta Kunden (Microsoft Cloud Deutschland – Die Azure Preview ist gestartet - https://blogs.technet.microsoft.com/microsoft_presse/microsoft-cloud-deutschland-die-azure-preview-ist-gestartet/). Ein genaues Datum ab wann die Services aus der Deutschen Cloud für alle Interessenten zur Verfügung stehen werden, wurde bislang nicht genannt. Interessenten können sich untern der Mailadresse azuregermany@microsoft.com für die Pilotbetrieb melden. Technisch bedingt werden die ersten Services Azure bzw. Azure AD Dienste sein. Das ergibt sich aus der Abhängigkeit von Office 365 Diensten wie z.B. SharePoint Online oder Exchange Online die ja jeweils ein Azure AD im Hintergrund benötigen. Informationen wie die genaue Roadmap aussieht und ab wann welcher Service zur Verfügung stehen wird, sind aktuell nicht bekannt. Ebenfalls unklar ist, was die Microsoft Services Azure, Office 365 und CRM Online aus der deutschen Cloud kosten werden. Es

ist stark davon auszugehen, dass die Preise über denen der Services aus der Public Cloud liegen werden. Wie hoch die Kosten genau sind bzw. um wie viel höher die Preise sein werden, ist aber noch unklar.

DIE DEUTSCHE MICROSOFT-CLOUD AUS RECHTLICHER SICHT

Mit der „deutschen Cloud“ versprechen Microsoft und T-Systems ihren deutschen Kunden die volle Kontrolle und Entscheidungshoheit über ihre Daten und damit letztlich auch Abhilfe gegen den Zugriff von ausländischen Behörden. Nicht zuletzt aufgrund der Entscheidung des EuGH zum Safe-Harbor-Abkommen und des noch anhängigen „Microsoft Ireland Case“ ist die Nachfrage nach einer rein deutschen Cloud offenbar groß: Nach der KPMG-BITKOM Studie „Cloud-Monitor 2015“ entspricht ein rein nationales Microsoft-Angebot den Wünschen der Kunden, denn 83% der deutschen Unternehmen erwarten, dass ihr Cloud-Anbieter seine Rechenzentren ausschließlich in Deutschland betreibt. Das rechtliche Novum der Microsoft-Cloud: Es wird in dem neuen Modell nicht nur ausschließlich Rechenzentren und Datenleitungen in Deutschland geben, die nur durch einen deutschen Provider betrieben werden, sondern auch einen deutschen „Datentreuhänder“. Dieser Beitrag untersucht, wie das Modell des deutschen Datentreuhänders rechtlich zu bewerten ist.

- Grundsatzfrage: Wie weit reichen die Eingriffsbefugnisse von US-Behörden?
- Eingriffsbefugnisse nach US-Recht

Unter bestimmten Umständen können US-Provider wie Microsoft verpflichtet werden, auch die in europäischen Rechenzentren gespeicherten Daten an US-Behörden herauszugeben.

MICROSOFT VS. UNITED STATES OF AMERICA

In welchem Maß US-Behörden Gebrauch von den beschriebenen Zugriffsrechten machen, ist nicht transparent, weil die Anbieter bei den Herausgabe-Anordnungen häufig durch sog. „Gag Orders“ zur Verschwiegenheit verpflichtet werden und Betroffene daher meist gar nicht von den Eingriffen Kenntnis erlangen.

DAS NEUE MICROSOFT-KONZEPT FÜR EINE SICHERE „DEUTSCHE CLOUD“

Die europäische Tochtergesellschaft Microsoft Irland Ltd. hat ihren europäischen Kunden schon bisher die Möglichkeit gegeben, durch den Abschluss von Zusatzverträgen die Datenverarbeitung im Rahmen ihrer Cloud-Produkte über die EU-Standardvertragsklauseln oder das Safe-Harbor Abkommen zu legitimieren. Die Möglichkeit der ausschließlichen Datenverarbeitung in Rechenzentren innerhalb der Europäischen Union (für Microsoft Azure insbesondere in Irland und den Niederlanden) bestand ebenfalls schon. Nun hat Microsoft für die „deutsche Cloud“ verschiedene technische und vertragliche Neuerungen eingeführt.

DIE RECHTLICHE KONSTRUKTION DER DATENTREUHAND

Zusätzlich zu dem Lizenzvertrag, den ein Kunde (in der Regel über einen Reseller) über die Nutzung der Cloud-Services mit Microsoft schließt, werden bei der deutschen Cloud „Treuhandvereinbarungen“ sowohl zwischen Microsoft und T-Systems als auch zwischen T-Systems und den jeweiligen Kunden geschlossen. Nach diesen Vereinbarungen kontrolliert allein die T-Systems als „deutscher Datentreuhänder“ des Kunden den physischen und technischen Zugriff auf die Kundendaten. Auch nach dem „Partnerschaftsvertrag“ zwischen Microsoft und T-Systems ist allein der Datentreuhänder berechtigt und (aufgrund der zuvor dargestellten technischen Infrastruktur) auch faktisch in der Lage, Zugriff auf die Kundendaten und/oder auf die Infrastruktur, auf der sich Kundendaten befinden, zu nehmen. Microsoft kann dagegen nur mit Hilfe von T-Systems als Datentreuhänder und/oder mit Erlaubnis des Kunden Zugriff erhalten. Der Datentreuhänder überwacht im Auftrag des Kunden jeden Zugriff.

SCHUTZSCHILD GEGEN UNBERECHTIGTEN ZUGRIFF?

Die entscheidende Frage ist also, ob diese Konstruktion der Datentreuhand am Ende einen Zugriff auf in Deutschland abgelegt Daten aus dem Ausland heraus abwenden kann. Anders gefragt: Was passiert, wenn die NSA (oder beispielsweise in einem US-Rechtsstreit ein Prozessgegner) die Herausgabe von Daten wünscht?

DR. MICHAEL RATH wurde 1969 geboren. Er studierte Rechtswissenschaften an den Universitäten in Saarbrücken und Köln und war an beiden Hochschulen als wissenschaftlicher Mitarbeiter tätig. Berufsbegleitend hat Dr. Michael Rath an der Universität Würzburg über das Recht der Internet-Suchmaschinen promoviert. Dr. Michael Rath ist Rechtsanwalt, Fachanwalt für Informationstechnologie-Recht und Partner der Luther Rechtsanwaltsgesellschaft mbH mit Sitz in Köln.

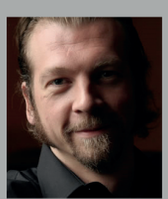
NICKI BORELL ist seit 2003 erfolgreich in der IT Branche tätig. Arbeitsumfeld sind sowohl KMU als auch Enterprise-Umgebungen und der Bereich behördlicher Datenhaltung. Als freier Autor, Trainer und Berater liegt sein Fokus auf dem Microsoft Technologie Stack. Seit 2011 ist Nicki Borell zusammen mit dem Team von Experts Inside in den Bereichen SharePoint, Office 365, und Windows Azure tätig. Dabei erstrecken sich seine Tätigkeitsfelder vom technischen Consulting über Projektmanagement bis hin zur strategische Prozessberatung. Nicki Borell ist professional Member der German Speaker Association e.V., Microsoft zertifizierter System Engineer, Database Administrator, IT-Professional und Microsoft MVP für Office Servers & Services.

Magdeburger Developer Days

18. Mai 2016 - Messehallen Magdeburg

Die Community Konferenz für Entwickler

Speaker:



Daniel Fisher
danielfisher.com

Alrik Schnapke
X-Trade

Dr. Holger Schwichtenberg
IT-Visions

Bernd Gronostay
Tutego

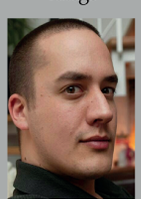
Oliver Zeigermann
zeigermann.eu

Rainer Stropek
software architects

Marius-Schulz
69 Grad

Hendrik Loesch
Saxonia Systems

Jan Fellien
User Group ALT.Net Berlin



Marco-Heimeshoff
Softwareentwickler & Coach

Andreas Richter
advanto Software

Daniel Meixner
Microsoft

Am 18. Mai 2016 ist es so weit: Dann startet die 1. Entwicklerkonferenz in den Messehallen Magdeburg! Wir freuen uns über die zahlreichen hilfsbereiten Unterstützer, die es uns ermöglichen ein erstklassiges Programm mit renommierten Speakern rund um die Themen .NET, JAVA, HTML und ALM zusammenzustellen. Es werden bis zu 20 Sessions, vollgestopft mit wertvollen Erfahrungen und Praxiswissen von Entwicklern für Entwickler, bereitgestellt.

Zusätzlich wird durch Aussteller ein Einblick in aktuelle Tools und Komponenten gewährt. Hier bieten sich optimale Möglichkeiten für Erfahrungsaustausch, Weiterbildung und Networking. Die Magdeburger Developer Days bieten ein umfassendes Themenspektrum aktueller Tendenzen zu OR Mappern, Datenbanken, Webtechnologien und Softwareentwicklungsprozessen. Erleben Sie die Entwickler-Community in Magdeburg am Fuße des Jahrtausendturmes!

www.md-devdays.de info@md-devdays.de @MiB_MD_DevDays

STILLSTAND VERMEIDEN

Mit modernen Managementmethoden den Herausforderungen des Wandels begegnen

Globalisierung, schneller technischer Fortschritt und sich rasch ändernde Lebensbedingungen hinterlassen ihre Spuren. Die IT-Branche war schon immer als dynamisches Universum bekannt, dennoch ist auch hier eine Beschleunigung auf allen Ebenen festzustellen. Der Mensch dagegen agiert am liebsten im gewohnten Umfeld. Veränderungen bedeuten Ungewissheit und können Angst machen. Die Aufgabe des IT-Managements ist es, das Team zielgerichtet auch durch unruhige Gewässer zu führen. Als Fachexperten fällt uns das nicht immer leicht. Eine gute Gelegenheit bei den klassischen Managementmethoden nach Ansätzen zu suchen.

Eine der größten Herausforderungen unserer Zeit ist der Umgang mit den Auswirkungen der Globalisierung. Globaler internationaler Handel, die ständige und sofortige Verfügbarkeit aller Informationen und ein rasanter technischer Fortschritt hinterlassen ihre Auswirkungen auf allen Ebenen. Die Lebens- und Arbeitsbedingungen sind einem ständigen Wandel unterworfen. Die Änderungen auf globaler Ebene haben ihre Auswirkungen auf unsere täglichen Arbeitsaufgaben. Die Ansprüche der Kunden sind wenig stabil und ändern sich rasch. Fast täglich strömen Informationen über neue Methoden und Verfahren auf uns ein, so dass es schwer fällt, den Überblick zu behalten. Verlässt man diese globale Sichtweise, so stellt sich die Softwareentwicklung gänzlich anders dar. Es handelt sich um eine technische Disziplin, welche eine sorgfältige Planung und ein akribisches Vorgehen im Detail erfordert. Programmiersprachen, die Bedienung von Werkzeugen, Methoden und die Entwicklung von Algorithmen müssen erlernt und in Ruhe geübt werden.

Haben Sie bisher primär auf operativer Ebene, zum Beispiel als Softwareentwickler gearbeitet, ist ihnen dieses Dilemma mit Sicherheit nicht unbekannt. Sie hatten vielleicht Glück und haben es nur als

„Hintergrundrauschen“ wahrgenommen. Den direkten Kontakt verspürt man dagegen auf der Ebene des IT-Managements. Hier sind Sie dafür verantwortlich, dass die Projekte erfolgreich bearbeitet werden. Dazu müssen Sie die Projektumwelt stetig beobachten. Kommt es zu Änderungen der Bedingungen, müssen mögliche Auswirkungen auf das Projekt geprüft werden. Dabei dürfen wichtige externe Veränderungen nicht übersehen werden. Hektische Überreaktionen sind jedoch ebenso schlecht. Ein sorgfältiges Antizipieren der Umweltsituation ist also notwendig. Das IT-Management muss sicherstellen, dass das Team gewissermaßen in einem gedanklich geschützten Raum seiner Arbeit nachgehen kann. Die Dynamik der Softwareentwicklung erfordert häufige Kurskorrekturen. Die Methoden und Vorgehensmodelle haben sich dieser Notwendigkeit angepasst. Bestes Beispiel ist die agile Projektführung.

Erfordert eine Projektumsetzung Anpassungsmaßnahmen, so sind diese meist auf einer Vielzahl von Ebenen erforderlich. Es genügt in der Regel nicht, dass lediglich eine technische Angleichung stattfindet. Vielmehr ist das gesamte Projekt mit all seinen Facetten bzw. die gesamte Unternehmung davon betroffen. Der Änderungsbedarf erstreckt sich dann über alle Bereiche und Ebenen und betrifft Fragen der Projektorganisation, der Aufgabenverteilung und der Organisationsstruktur. Die Mitarbeiter haben jedoch ihre eigenen Vorstellungen, Ziele und persönlichen Hintergründe, welche sie in die tägliche Arbeit einbringen. Die Notwendigkeit von Änderungen auf Unternehmensebene hat damit auch immer direkte Auswirkungen auf jeden einzelnen. Es wäre eine Binsenweisheit, wenn man nun glaubt, dass man die offensichtlichen Veränderungen einfach aus der Ebene des Managements administrieren kann und die Umsetzung automatisch erfolgt. In der modernen Managementlehre wird das Thema unter dem Stichwort Change Management diskutiert. Es lohnt sich als ITler mit den Ansätzen, der Denkweise und den Lösungsvorschlägen vertraut zu machen. Nachfolgend finden Sie eine kompakte Einführung in das Thema, welches auch zunehmend eine Bedeutung im Bereich des IT-Services bekommt.

CHANGE MANAGEMENT IM KONTEXT DER IT

Unter dem Begriff Change Management wird eine Anpassung der Unternehmensstrategien und -strukturen an sich ändernde Rahmenbedingungen verstanden. Der Wandel im Unternehmen ist keine außergewöhnliche Sache, sondern eine Regelerscheinung. Betrachtet man die Ansätze des Change Management in den letzten 20 Jahren, so kann man diese gemäß Abbildung 1 darstellen. Eine exakte Abgrenzung der Ansätze ist jedoch nur mit Einschränkungen möglich, da die Kernelemente teilweise parallel oder überlappend entwickelt wurden. Als Teil der IT Infrastructure Library (ITIL) wird unter dem Change Management eine Anpassung der IT-Infrastruktur verstanden, welche effizient, kontrolliert und unter Risikominimierung für den Betrieb durchgeführt wird [2]. ITIL stellt einen Leitfaden zur Planung, Erbringung und Unterstützung von IT-Service-Leistungen dar. Dieser ist mittlerweile ein internationaler Standard. Ursprünglich wurde ITIL durch das Office of Government Commerce im Auftrag der britischen Regierung entwickelt. Unternehmen haben die Möglichkeit sich nach ITIL (ISO 20000) zertifizieren zu lassen. Grundsätzlich wird ITIL im Kontext des IT Service Managements betrachtet (ITSM). Unter ITSM wird die Standardisierung der Prozesse und Methoden im Unternehmen verstanden. Will man bei der Einführung von ITIL erfolgreich sein, so ist es empfehlenswert folgende Fakten zu berücksichtigen:

- Die Kunden der IT sind diejenigen, welche die Leistungen beauftragen und bezahlen.
- Die IT dient zur Unterstützung der Kerngeschäftsprozesse der Kunden, sie hat keinen Selbstzweck.
- Das Kultur- und Organisationsmodell umfasst die Zuordnung von Verantwortlichkeiten und die unternehmensspezifische gewachsene Kultur eines Unternehmens. Dieses ist zu beachten.

Da sich die Unternehmen und die Umwelt ständig ändern, ist davon auch die IT als Dienstleistung betroffen. Sie kann nicht stehen bleiben, eine stetige Anpassung ist unabdingbar.

PROBLEME DES WANDELS

Unternehmen aller Branchen sind ständigen Wandlungsprozessen unterworfen. Erfolgreich abgeschlossene Veränderungen sind dabei die Basis für ein weiteres Wachstum. Dagegen können fehlgeschlagene Veränderungsprozesse im schlimmsten Fall zur Liquidation des Unternehmens führen bzw. eine ernsthafte Krise auslösen. In der medialen Öffentlichkeit ist dazu ein Paradoxon zu beobachten. Geht ein Unternehmen die unabdinglichen Änderungen – ggf. auch mit den notwendigen radikalen Einschnitten in allen Bereichen – an, so ist die Berichterstattung meist von negativen Schlagzeilen geprägt. Der Tenor lautet: Was mutet

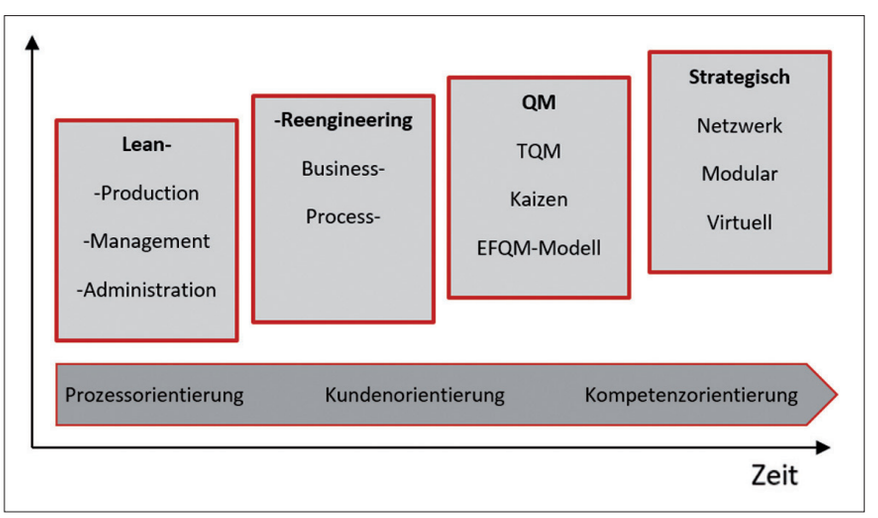


Abbildung 1: Change-Management-Ansätze der letzten zwei Jahrzehnte [1].

das Management den Mitarbeiten nur zu? Werden dagegen Veränderungen, aus welchen Gründen auch immer, nicht zügig umgesetzt, so sind die Folgen in keiner Weise besser. Mittel- bis langfristig wird das Unternehmen in eine wirtschaftliche Krise geraten, welches wiederum Auswirkungen auf alle Unternehmensbereiche und letztendlich auch auf die Beschäftigungsverhältnisse haben wird. Tritt dieser Umstand ein, werden in der Öffentlichkeit das Ausbleiben von Anpassungsmaßnahmen und der damit einhergehende Stillstand mit all seinen negativen Folgen kritisiert. Konkretisieren wir dieses für die IT-Branche und greifen wir die Situation eines langjährig etablierten Softwareentwicklungsunternehmens heraus. Unser fiktives Unternehmen mag sich über Jahre auf die Entwicklung von klassischen Desktop-Applikationen spezialisiert haben. Auf diesem Gebiet sind die Mitarbeiter sehr gut. Sie sind Spezialisten und die anstehenden Projekte können meist effizient und erfolgreich zum Abschluss gebracht werden. Die meisten Mitarbeiter sehen jedoch mit gemischten Gefühlen eine Veränderung in den künftigen Anforderungen auf sich zukommen. Diese speisen sich aus zwei Perspektiven:

(1) Cloud-Computing: Zunehmend lagern Unternehmen alle möglichen IT-Ressourcen in die Cloud aus. Betriebliche Anwendungen müssen eine Anbindung an diese Systeme erfahren, ggf. auch vollständig in die Cloud verlagert werden.

(2) Enterprise mobile Computing: Der notwendigen Flexibilität geschuldet können Teile des Geschäftsprozesses durch Applikationen auf mobilen Endgeräten erledigt werden. Neben klassischen Desktop-und Web-Anwendungen sind zunehmend auch Apps für Smartphones für die unterschiedlichen Betriebssysteme gefragt.

Beide Szenarien sind unausweichlich. Unser fiktives Unternehmen kann sie jedoch (noch) nicht bedienen. Dazu sind weder die Voraussetzungen geschaffen, noch verfügen die Mitarbeiter über die notwendigen Kenntnisse. Es drängt nicht unmittelbar, da das laufende Tagesgeschäft noch einträgliche Einnahmen generiert. Aber erste Gewitterwolken lassen die Veränderung der Großwetterlage erkennen. Nach kurzer Auseinandersetzung mit dem Thema wird beschlossen abzuwarten. Trotz des Reizes bei einigen Mitarbeitern neuen Herausforderungen positiv entgegenzustehen, scheint der notwendige Anpassungsbedarf zu groß. Reorganisation, Weiterbildung der Mitarbeiter, Etablierung anderer Vorgehensmodelle,... wer soll das alles stemmen? Hinzukommt, dass die laufenden Projektarbeiten weiterhin zu erledigen sind. Vielleicht hat das IT-Management auch erkannt, dass die Herausforderungen mit den vorhanden personellen und sachlichen Ressourcen nicht geschafft werden können. Es scheut jedoch die Diskussion in alle Richtungen, denn diese wird unbequem.

Die Literatur benennt vier Grundformen des Wandels: Restrukturierung, Reorientierung, Revitalisierung und Remodellierung. Die vorhandenen Unternehmensstrukturen und -prozesse, sowie eingesetzten Systeme und Ressourcen sind Gegenstand einer Restrukturierung. Daraus ergeben sich die typischen Restrukturierungsmaßnahmen wie z.B. eine Optimierung von Abläufen und ein notwendiger Abbau von Arbeitsplätzen. Letztendlich soll die organisationale Unternehmenshierarchie flacher werden. Bei der Reorientierung geht es um die strategische Neuausrichtung eines Unternehmens, z.B. die Neuentwicklung von Geschäftsfeldern. Ziele der Revitalisierung sind grundlegende Veränderungen hinsichtlich der Fähigkeiten und des Verhaltens der Organisationsmitglieder, z.B. eine Änderung des Führungsstils oder das Aneignen von Fachkenntnissen in bisher nicht abgedeckten Bereichen (siehe eben ausgeführtes Beispiel). Die Remodellierung ist die am tiefsten greifende Form des Wandels. Diese zielt auf eine Veränderung von Werten, Überzeugungen und Einstellungen der Organisationsmitglieder [1]. Bekanntermaßen ist die Änderung dieser „inneren" Eigenschaften von Menschen am schwersten zu erreichen. Selbstverständlich sind diese Grundformen des Wandels nicht trennscharf voneinander abzugrenzen.

Viele notwendige Anpassungen sind langfristig vorherzusehen, es trifft jedoch nicht auf alle zu.

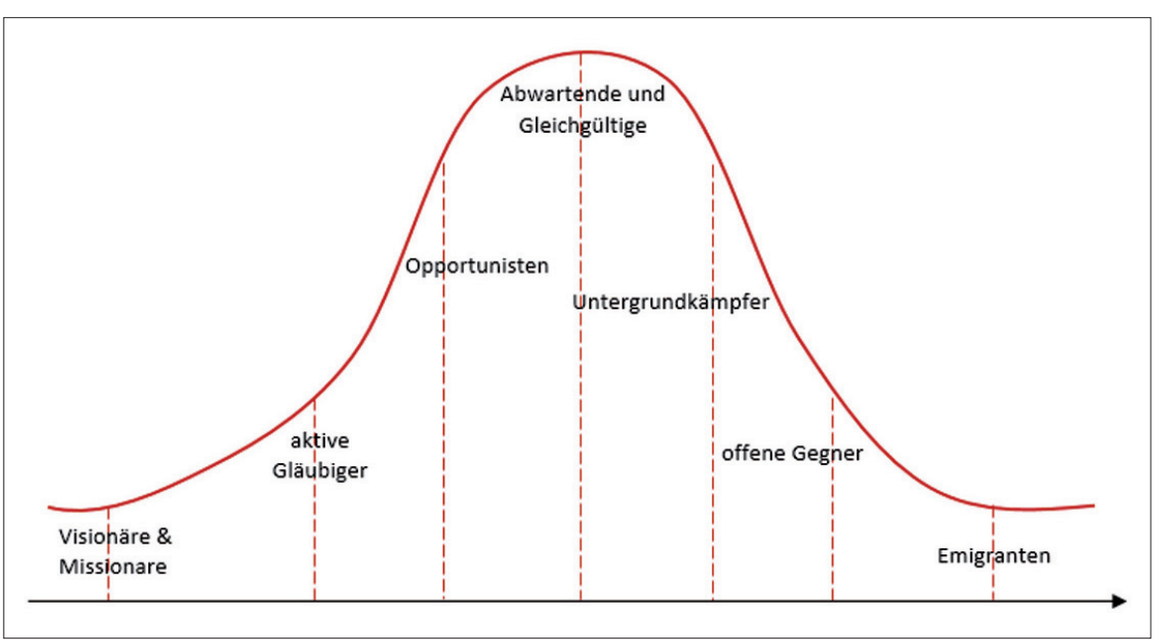


Abbildung 2: Typische Einstellungen Gegenüber dem organisatorischen Wandel [1]

Manche strukturelle Veränderungen sind nicht beabsichtigt, treten zufällig auf und bleiben gelegentlich sogar über einen längeren Zeitraum unbemerkt. Hier handelt es sich um einen ungeplanten organisatorischen Wandel. Das Unternehmen muss schnell reagieren, nicht selten geht es dabei um existenzielle Fragen. Ein planbarer Wandel ist demgegenüber aktiv organisiert, kontrolliert und gesteuert. Das Ziel ist die Steigerung der Effektivität und Effizienz der Unternehmensprozesse.

DEN WANDEL ERFOLGREICH GESTALTEN

Die Erkenntnis lautet: Wer erfolgreich werden oder weiterhin bleiben möchte, muss sich an die ständig wechselnden Anforderungen anpassen! Wichtig ist dabei „immer am Ball bleiben", die notwendigen Anpassungen müssen in kürzester Zeit durchgeführt werden, damit gegenüber der Konkurrenz kein entscheidender Wettbewerbsnachteil entsteht [3]. Der Harvard Professor John P. Kotter weißt in seiner Studie darauf hin, dass ca. 70% aller angestoßenen Change Prozesse scheitern. Die zwei wichtigsten Gründe dafür sind:

- Widerstand bei den Mitarbeitern und
- Rückfall in alte Muster.

Ansatzpunkte für eine erfolgreiche Gestaltung des Change Management Prozesses stellen wir nachfolgend vor. Die erste Voraussetzung für den Erfolg bei Veränderungen ist Etablierung einer umfassenden Kommunikation! Veränderungen bringen Neuigkeiten mit sich. Die betroffenen Mitarbeiter können dadurch sehr verunsichert werden. Aus diesem Grund ist es sehr wichtig die interne Kommunikation im Unternehmen in Bezug auf das Change Management richtig zu gestalten. Folgende drei Grundregeln sind zu nennen:

1. Mitarbeiter frühzeitig informieren.
2. Nachvollziehbare Erklärungen liefern: Dieses hilft die Veränderungen leichter zu akzeptieren.
3. Einen regelmäßigen Dialog mit den Mitarbeitern führen: Die Kommunikation des Change Managements ist keine Einbahnstraße.

Das Change-Management ist ein wiederkehrender Prozess, der sich immer weiterentwickelt. Man kann also nicht sagen, dass mit dem (erfolgreichen) Abschluss eines Veränderungsprozesses dieser nun für alle Zeiten abgeschlossen ist. Die nächste notwendige Veränderung kommt mit Sicherheit. Notwendig sind eine gut durchdachte Planung und ein hohes Maß an Kommunikation bei der Umsetzung. Change Prozesse können nicht erzwungen bzw. von oben administriert werden! Nachfolgend einige Fakten, auf welche es bei der Umsetzung ankommt:

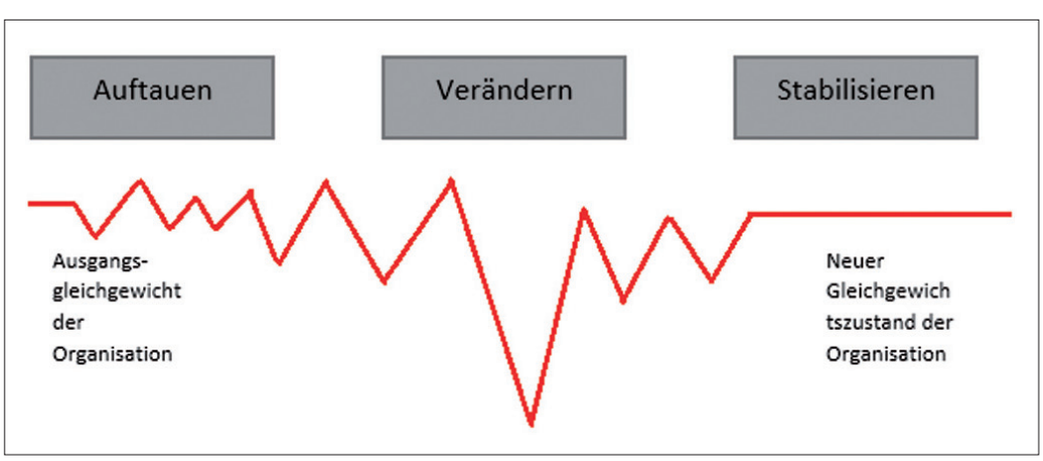
- Die Notwendigkeit der Veränderungen muss richtig vermittelt werden. Am besten gelingt es, wenn die Mitarbeiter klar darüber informiert werden, zu welchen Folgen es kommt, wenn das Unternehmen die notwendigen Veränderungen nicht rechtzeitig angeht.
- Vor der Umsetzung sollte ein konkreter Plan mit klar definierten Zielen vorliegen. Dieses erhöht die Vertraulichkeit, d.h. die Betroffenen fühlen sich in guten Händen.
- Die betroffenen Mitarbeiter sind einzubeziehen. Ihnen ist die Möglichkeit zu geben sich aktiv an den

LEWINS GOLDENE REGEL

Kurt Levin hat 1947 für Veränderungsprozesse in Gruppen und Organisationen ein 3-Phasen-Modell formuliert [4]:

1. ***Auftauen (unfreezing):*** Das System soll den Gleichgewichtszustand aufgeben und damit eine Bereitschaft zur Veränderung signalisieren. Alte Gewohnheiten werden in Frage gestellt, neue Ideen diskutiert
2. ***Verändern (moving):*** Neue Standards werden durch direktes Eingreifen der Verantwortlichen eingeführt.
3. ***Stabilisieren (freezing):*** Es erfolgt eine „Umgewöhnung" der Gruppe auf die neuen Standards. Die Veränderungsbereitschaft steigt, wenn Einverständnis über den notwendigen Wandel hergestellt ist, am Veränderungskonzept selbst mitgearbeitet, die Maßnahmen selbst beschlossen und die Veränderungen begreifbar gemacht wurden.

Wichtig ist dabei, dass nach den Veränderungen wieder ein Gleichgewichtszustand hergestellt wird. Jedoch soll eine neue Ebene erreicht werden, d.h. eine Rückkehr in den alten Zustand ist zu vermeiden. Das Modell von Lewin wurde später durch unterschiedliche Ansätze erweitert, beispielsweise durch Lawrence im Jahr 1954 mit Studien zum Verhalten im Wandel.



Das Änderungsnetz nach Levin [4]

Veränderungen zu beteiligen und diese mitzugestalten.

■ Der Change Prozess muss transparent gestaltet werden. Neue Informationen sollen weitergegeben, wichtige Entscheidungen erklärt und Unsicherheiten beseitigt werden.

Eine Zusammenfassung dieser Handlungsempfehlungen ist als Lewins Goldene Regel (siehe Textkasten) bekannt.

AUF DIE MITARBEITER KOMMT ES AN

Mitarbeiter reagieren auf die anstehenden Veränderungen im Unternehmen sehr unterschiedlich [1]. Die typischen Einstellungen gegenüber dem Wandel zeigt die Abbildung 2 Die Erfahrungen aus der Praxis zeigen, dass etwa ein Drittel der Betroffenen dem Wandel offen und positiv gegenüber stehen, ein Drittel neutral eingestellt und eine abwartende Haltung bevorzugen und das letzte Drittel die Veränderungen ablehnt. Diese Einteilung kann weiter differenziert werden. Wichtige Grundtypen der Mitarbeiter sind.

■ Visionäre und Missionare spielen eine Schlüsselrolle. In der Regel ist es das Top-Management, welches die Ziele und die Maßnahmen des Wandels erarbeitet haben. Diese Personen glauben stark an den Erfolg der Veränderungen und versuchen auch andere von der Notwendigkeit zu überzeugen.

■ Aktive Gläubige sind bereit den Wandel zu akzeptieren und ihre ganze Arbeits- und Überzeugungskraft dafür einzusetzen. Sie bewerten die persönlichen Vor- und Nachteile, welche der Wandel mit sich bringt. Das Verhalten gegenüber den veränderungsbereiten Vorgesetzten ist positiv und änderungsbereit. Im Gegensatz dazu ist das Verhalten gegenüber den Kollegen eher zurückhaltend.

■ Zu den Abwartenden und Gleichgültigen gehören die meisten Mitarbeiter. Sie sind kaum bereit sich an den Veränderungen aktiv zu beteiligen. Sie warten ab und wollen erst sehen was tatsächlich passiert. Spürbare Verbesserungen der persönlichen Arbeitssituation können sie dazu bewegen, aktiver zu werden.

■ Untergrundkämpfer und offene Gegner sind gegen jegliche Veränderungen. Sie sind überzeugt, dass die Entscheidungen falsch und nicht zielführend sind. Untergrundkämpfer gehen verdeckt vor, offene Gegner zeigen ihre Haltung.

■ Emigranten sind Personen, die beschlossen haben den Wandel nicht mitzutragen und verlassen das Unternehmen. Dabei handelt es sich in vielen Fällen um Personen, die keine Perspektive in der die Phase der Veränderung und der darauf folgenden Zeit haben.

FAZIT UND AUSBLICK

Schon Kant wusste um die Bedeutung des Themas. Er schrieb anlässlich eines Rufes an eine Universität: „Alle Veränderung macht mich bange“. Dieser Satz beschreibt auf sehr einfache, aber eindrückliche Art und Weise die Problemlage des Change Managements. Auch die Organisationsstrukturen in der IT erfahren in der letzten Zeit wieder einen neuen Bedarf an Veränderungen um die künftigen Anforderungen bestehen zu können. Denken wir in diesem Zusammenhang nur an Auswirkungen auf die Unternehmensstruktur und die Aufgaben der Mitarbeiter bei Themen wie DevOps, agile Entwicklungsmethoden oder Mobile- und Cloud-Computing. Langjähre Prozesse und Verhaltensweisen sind zu hinterfragen, neue Ansätze sind zu probieren. Doch Stillstand und Abwarten wäre auf jeden Fall falsch. Das äußerte bereits Sir Peter Ustinov mit seinem bekannten Zitat: „Wer nicht zweifelt, muss verrückt sein. Wer sich nicht verändert, ist dumm.“ Zum Glück kennt die etablierte Managementlehre gute Lösungsansätze um dem Problem beizukommen und nicht zu kapitulieren. Als IT-Manager ist es unsere Aufgabe sie an die Belange unseres Metiers zu adaptieren und dabei stets gemeinsam im Team und nicht alleine zu gehen. In diesem Sinne: „Nichts ist so beständig wie der Wechsel“ (Heraklit von Ephesus, etwa 540 - 480 v. Chr., griechischer Philosoph). Gestalten Sie den Wandel aktiv: „Drive change or it will drive you“ (zitiert nach Picot/Freudenberg/Gaßner).

LITERATUR UND LINKS

[1] Vahs, D.: Organisation. Ein Lehr- und Managementbuch, Schäffer-Poeschel Verlag Stuttgart, 2009

[2] http://wirtschaftslexikon.gabler.de/Archiv/381707702/itil-v5.html

[3] http://karrierebibel.de/change-management/

[4] Schreyögg, G.: Organisation. Grundlagen moderner Organisationsgestaltung, Gabler Verlag, 2008

DR. VEIKKO KRYPCZYK
studierte und promovierte Betriebswirtschaftslehre mit dem Schwerpunkt Wirtschaftsinformatik.
Er arbeitet als Fachautor und ist begeisterter Programmierer

OLENA BOCHKOR
studierte Betriebswirtschaftslehre u.a. mit dem Schwerpunkt Wirtschaftsinformatik. Weitere Informationen zu diesen und anderen Themen der IT finden Sie unter http://it-fachartikel.de.

VISUAL STUDIO ERWEITERBARKEIT

Eine vielseitige IDE

Das Visual Studio ist eine sehr komplexe IDE (Integrated Development Environment), die uns als Entwicklern sehr viel Komfort bereitet. Neben C#, Visual Basic.NET und C++ beherrscht Visual Studio eine Menge anderer Sprachen, wie z.B. F#, Java, JavaScript, TypeScript, PHP und Python. Von Hause aus liefert Microsoft einen Blumenstrauß an Projektvorlagen, mit denen auf einfache Art und Weise die verschiedensten Projekte für unterschiedliche Anwendungsfälle angelegt werden können. So können aus dem Stand Projekte für das Web auf Basis von ASP.NET mit C# oder Visual Basic.NET oder für den Desktop, mit dem in die Jahre gekommenen Windows Forms oder der WPF (Windows Presentation Foundation) erzeugt werden. Das ist aber längst nicht alles. Projektvorlagen für SQL Server, SharePoint, LightSwitch, PowerShell, die Microsoft Azure-Cloud, mobile Anwendungen mit Xamarin, iOS-Apps, Android-Apps, Games mit Unity oder der Unreal Engine machen das Visual Studio zu einem Allrounder. Für jeden ist etwas dabei. Zudem sind Projektvorlagen beliebig erweiterbar.

Mit NuGet, NPM oder Bower als Package Manager, können auf einfache Art und Weise Bibliotheken in die eigenen Projekte inkludiert werden. Auch hier bietet Visual Studio eine Unterstützung. Denn die Bibliotheken müssen nicht unbedingt die eigenen sein. Vielmehr bietet Visual Studio die Möglichkeit, Fremdbibliotheken von externen Quellen zu beziehen. Über eine einfache UI in der IDE lassen sich derartige Pakete auf externen Servern durchsuchen und der eigenen Anwendung hinzufügen. Und, last but not least, gibt es da die Visual Studio Extensions. Über das Menü TOOLS\EXTENSIONS & UPDATES lässt sich das Visual Studio mit Erweiterungen ergänzen. Das Menü führt in die Visual Studio Gallery. Hier gibt es eine Unmenge an Plug-Ins, welche die IDE (mal mehr, mal weniger) sinnvoll erweitern. Der ReSharper ist wohl eines der bekanntesten davon. Abbildung 1 zeigt die Galerie, die ebenfalls im Web zur Verfügung steht und dort durchsucht und erforscht werden kann.

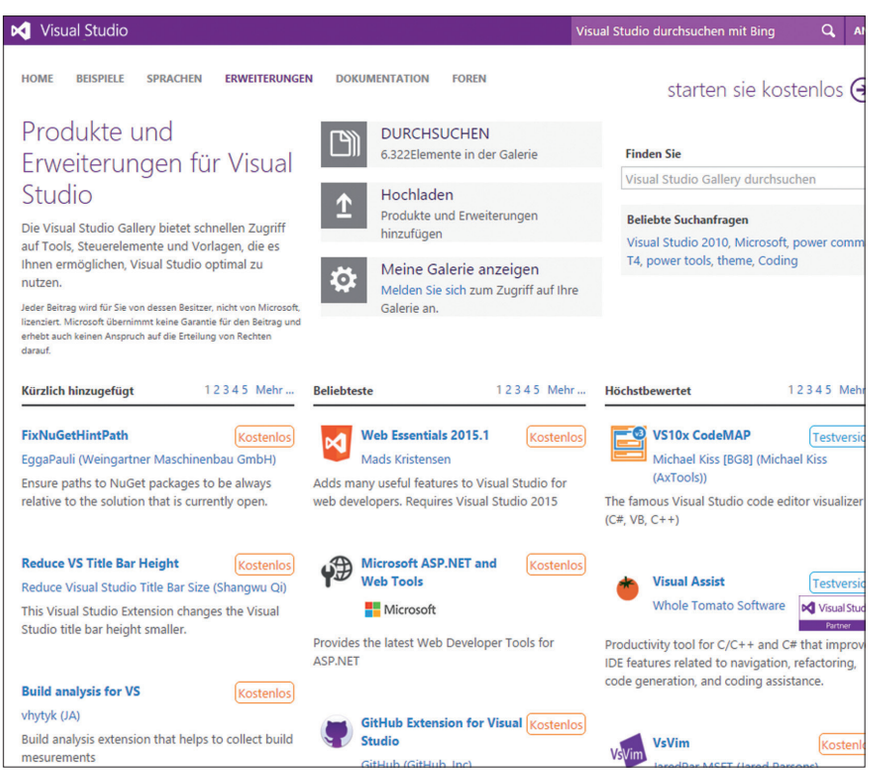


Abbildung 1

All die Flexibilität und Erweiterbarkeit sind den Anforderungen an eine moderne IDE geschuldet. Und genau aus diesem Grund soll die Erweiterbarkeit Kernthema in diesem Artikel sein. Denn so komplex und vielfältig wie die IDE ist, so umfangreich sind auch die Einstiegs- bzw. Erweiterungspunkte. Wer z.B. denkt, der Codeeditor sei lediglich ein simpler Texteditor, der irrt sich gewaltig. In diesem Artikel sollen die Erweiterungsmöglichkeiten des Visual Studios aufgezeigt werden. Insbesondere werden wir einen genaueren Blick auf den Codeeditor werfen. Ein paar Beispiele sollen einen ersten Eindruck vermitteln, auf welchen Ebenen Visual Studio erweitert werden kann. Oberstes Ziel ist dabei immer die Steigerung der Produktivität für uns als Entwickler, der sichere Umgang mit dem Code und die Unterstützung durch die IDE zur Steigerung der Code-Qualität. Am spannendsten wird es dann, wenn wir zusätzlich Roslyn [2] mit auf das Spielfeld schicken. Roslyn – die Microsoft .NET Compiler Plattform – erlaubt uns die Instrumentalisierung des Compilers. Zugriff auf die Ergebnisse aller Phasen der Compiler-Pipeline: auf den Syntaxbaum aus der syntaktischen Analyse, den Symboltabellen aus der semantischen Analyse und den generierten Code

aus der Generierungsphase. Zusammen mit den Visual Studio Extensions eine unschlagbare Mannschaft auf dem Spielfeld der alltäglichen Softwareentwicklung.

ERSTELLEN EINER VISUAL STUDIO-ERWEITERUNG

Wie oben erwähnt, bietet das Visual Studio eine Vielzahl an Erweiterungspunkten an. Dafür liefert Microsoft gleich eine Hand voll Projektvorlagen und Projekttypen mit. Diese sind im Dialog zum Anlegen eines neuen Projektes unter Extensibility zu finden. Von Hause aus kommen Vorlagen für C# Items und C# Projekte daher. Möchte man die Shell des Visual Studios erweitern, so benötigt man das Visual Studio SDK. Dieses muss also zuvor heruntergeladen werden [3]. Auch für Roslyn gibt es ein SDK [4], welches mit 3 zusätzlichen Projektvorlagen daherkommt. Abbildung 2 zeigt den Dialog zum Anlegen eines neuen Projektes nach Installation beider SDKs. Im Rahmen dieses Artikels interessieren wir uns zunächst nur für die VSIX-Projektvorlage. Ein VSIX-Projekt stellt im Wesentlichen einen installierbaren Container für Visual Studio dar. In diesem Container können unterschiedliche Erweiterungen enthalten sein, die nach und nach dem Projekt hinzugefügt werden können. Installiert man Visual Studio Extensions über das Menü TOOLS\EXTENSIONS AND UPDATES, zieht man im Grunde genommen derartige VSIX-Container vom Server und installiert diese in der IDE. Die Installation kann ebenso per Doppelklick auf das Paket im Dateisystem erfolgen, da die VSIX-Dateierweiterung mit dem Visual Studio verbunden ist. Pakete und Erweiterungen lassen sich so also auf bequeme Art und Weise verteilen, ohne diese vorher über einen externen Server bereitstellen zu müssen.

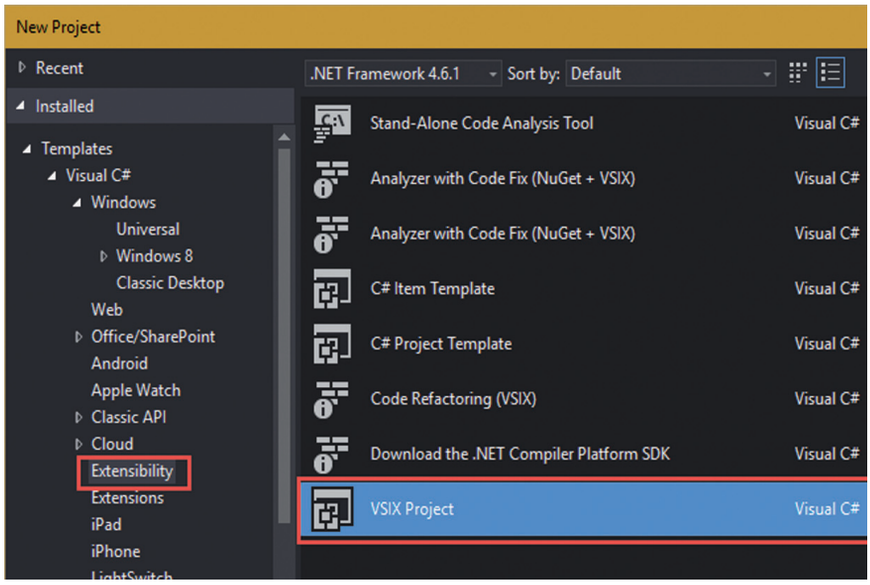


Abbildung 2

Wie schaut so ein VSIX-Projekt nun aber aus? Abbildung 3 zeigt die initiale Struktur des Projektes. Im Wesentlichen beinhaltet dieses Projekt nur eine wichtige Datei – das Manifest. Ebenfalls in Abbildung 3 ist der Editor zu sehen, mit dem das Manifest in Visual Studio bearbeitet werden kann. Hier können Name, Beschreibung, Lizenz, Icon und andere Metadaten hinterlegt werden. Unter „Install Targets“ können die Versionen des Visual Studios hinterlegt werden, in denen die Extension installiert werden kann. Unter „Assets“ können dann zusätzliche Ressourcen hinzugefügt werden, wie z.B.

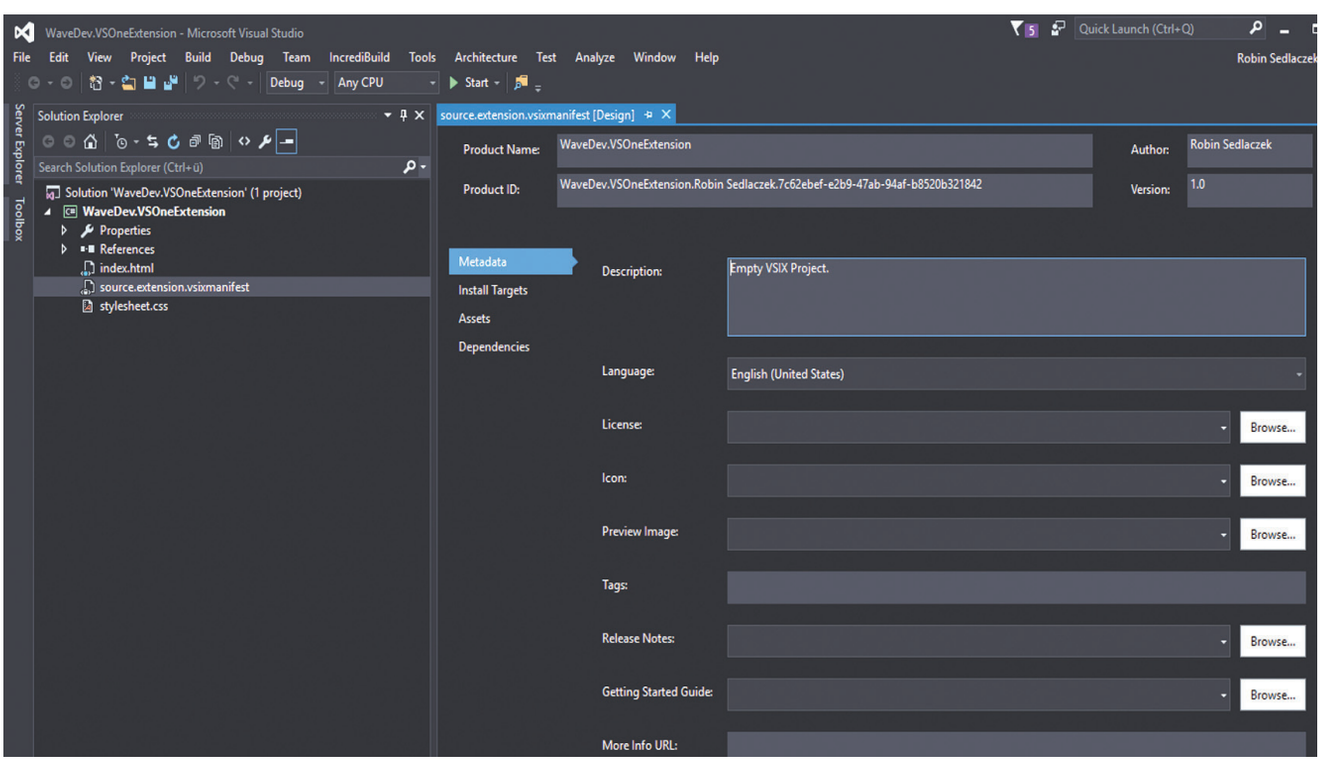


Abbildung 3

Libraries. Diese werden dann beim Build mit in das VSIX eingepackt und ausgeliefert. Somit können verschiedene Extensions modular in Libraries aufgeteilt und mit einem VSIX-Paket zusammengefasst und ausgeliefert werden. Im Bereich „Dependencies“ werden weitere Abhängigkeiten, wie z.B. zu einer bestimmten .NET Framework-Version oder Bibliotheken von Drittanbietern, hinterlegt. Ein VSIX-Paket liefert also die Infrastruktur, die zur Auslieferung von Erweiterungen benötigt wird.

Schauen wir uns an, welche Arten von Erweiterungen es gibt. Dazu soll an dieser Stelle ein Blick in den Dialog zum Hinzufügen eines neuen Elementes zu unserem VSIX-Projekt genügen. Abbildung 4 zeigt den entsprechenden Dialog. Auf der linken Seite wurden die Elemente nach der Kategorie „Extensibility“ gefiltert. Es sei noch einmal darauf hingewiesen, dass die gelisteten Elemente mit den entsprechenden SDKs für Roslyn und Visual Studio daherkommen. Die zur Verfügung stehenden Elemente sind in Tabelle 1 aufgelistet.

Die Liste lässt erahnen, wie mannigfaltig die Erweiterungsmöglichkeiten im Visual Studio sind. Wie aber bereits erwähnt, interessiert uns in diesem Artikel die Erweiterbarkeit des Visual Studio Text- bzw. Codeeditors. Bevor wir uns diesen aber genauer anschauen, bleibt die Frage, wie funktionieren solche Erweiterungen unter der Haube? Wie bindet Visual Studio diese in die Shell nahtlos ein?

EXTENSIONS AUF DEN GRUND GEGANGEN

Für die Erweiterbarkeit setzt Visual Studio auf zwei grundlegende Technologien. Zum einen stellt die IDE Services zur Verfügung, die in den eigenen Erweiterungen verwendet werden können. Derartige Services

Element	Beschreibung	Quelle
Analyzer	Roslyn-Analyzer zur Auffindung von Schwachstellen im Code. Hängt sich direkt in die Compiler-Pipeline von C# oder Visual Basic.	.NET Compiler Platform SDK
CodeFix	CodeFix bietet dem Entwickler eine mögliche Verbesserung des Codes an, passend zu einer Schwachstelle, die durch einen Analyzer identifiziert wurde.	.NET Compiler Platform SDK
Command Extension	Mit Command Extensions können Einträge zum Kontextmenü in Visual Studio UML-Diagrammen hinzugefügt werden.	Visual Studio SDK
Gesture Extension	Mit Gesture Extensions können Erweiterungen für die Reaktion auf Mouse-Ereignisse in Visual Studio UML-Diagrammen hinzugefügt werden.	Visual Studio SDK
Model Validation Extension	Model Validation Extensions erlauben eigene Validierungen in Visual Studio UML-Diagrammen.	Visual Studio SDK
Refactoring	Mittels Refactorings können eigene Code-Umwandlungen implementiert werden, die dann über eine Light Bulb im Visual Studio-Editor zur Verfügung stehen.	.NET Compiler SDK
Custom Command	Mit Custom Commands können eigene Menüeinträge im Visual Studio-Menü hinzugefügt werden. Für derartige Commands ist ein Visual Studio-Package nötig.	Visual Studio SDK
Custom Tool Window	Mit Custom Tool Windows können eigene Werkzeugfenster der Visual Studio-Shell hinzugefügt werden. Für derartige Fenster ist ein Visual Studio-Package nötig.	Visual Studio SDK
Editor Classifier	Editor Classifier bilden den Einstieg in ein eigenes Syntax Highlighting. Hier können zusätzlich Classification Formate exportiert werden, über welche das Syntax Highlighting gesteuert werden kann.	Visual Studio SDK
Editor Margin	Editor Margins bilden ein Rahmen um den Visual Studio Codeeditor. In derartigen Margins sind z.B. Scrollbars, Breakpoint-Symbole und Zeilennummern dargestellt.	Visual Studio SDK
Edit Text Adornment	Mit Editor Text Adornments lässt sich Text im Visual Studio Codeeditor markieren bzw. ergänzen. Die rote Unterkringelung von nicht kompilierbarem Code ist beispielsweise so ein Adornment.	Visual Studio SDK
Edit Viewport Adornment	Mit Editor Viewport Adornments lassen sich, ähnlich zu den Text Adornments, Markierungen relativ zum visuellen Feld des Texteditors implementieren. Diese hängen dann vom Sichtbereich des Editors ab, nicht aber von etwaigem Text.	Visual Studio SDK
Visual Studio Package	Mit einem Visual Studio Package lassen sich Elemente über das Managed Package Framework (MPF [5]) in das Visual Studio integrieren.	Visual Studio SDK
Windows Form Toolbox Control	Ein Windows Form Toolbox Control ist der Einstiegspunkt zur Erweiterung der Toolbox in einem Windows Forms-Designer.	Visual Studio SDK
WPF Toolbox Control	Ein WPF Toolbox Control ist der Einstiegspunkt zur Erweiterung der Toolbox in einem WPF-Designer.	Visual Studio SDK

Tabelle 1: Erweiterungselemente

können ebenfalls zur Verfügung gestellt werden, um anderen Plug-Ins zu ermöglichen, selbst implementierte Services einzubinden. Das Ganze funktioniert über das sogenannte Service Locator Pattern [6]. Möchte man z.B. einen Service zum Durchsuchen einer Quellcodedatei, eines Projektes oder gar einer ganzen Solution benutzen, muss man diesen Service anfordern. Das geschieht über die Methode GetService (siehe Listing 1), die z.B. in einem Visual Studio-Package über die Basisklasse Package zur Verfügung gestellt wird.

Möchte man eigene Services zur Verfügung stellen, so muss man eine Klasse implementieren, die ihrerseits das Interface IServiceProvider realisiert. Mit diesem Mechanismus können also Dienste bereitgestellt und konsumiert werden. Die GetService-Methode ist nicht sonderlich elegant. Object als Rückgabetyp ist unschön und kann zu Fehlern führen. Es ist nicht sichergestellt, dass ein Service des Typs zurückgegeben wird, der im Parameter serviceType angefordert wird. Zusätzlich muss der Rückgabewert gecastet werden. Eine generische Methode wäre hier sicherlich schöner. Das Service Locator Pattern stammt an dieser Stelle aber noch aus älteren Zeiten, als COM noch am Ruder war. Seit einigen Versionen der IDE geht Microsoft aber einen etwas anderen Weg.

Das Zauberwort heißt MEF. MEF ist das Managed Extensibility Framework und wurde mit dem .NET Framework eingeführt. MEF ist das hauseigenen Plug-In-Framework, das nunmehr rege Anwendung im Visual Studio findet. MEF ist ein IoC-Framework und funktioniert ein wenig anders als das Service Locator Pattern (das seinerseits in der Fachwelt sehr gerne heiß diskutiert und dessen Konformität zu den SOLID-Prinzipien gerne in Frage gestellt wird). Im Gegensatz zum Service Locator werden Abhängigkeiten nicht zur Laufzeit angefordert. Vielmehr werden Abhängigkeiten im Code deklariert. Zur Laufzeit kümmert sich MEF dann um die Komposition der Komponenten, auch „Parts“ genannt. Beim Start der Anwendung werden alle Abhängigkeiten aufgelöst, der Abhängigkeitsgraph der einzelnen Parts wird initialisiert und die Parts werden zur Komposition miteinander verbunden. Man sieht das gelegentlich beim Starten von Visual Studio in einem kleinen Dialogfenster mit der Nachricht „Composing MEF Graph“. MEF funktioniert dabei über Attribute. Über ein Import-Attribute (oder Ableitungen davon), kann eine Abhängigkeit angegeben werden, die das Visual Studio auflöst und indiziert. Über Export-Attribute (oder Ableitungen davon) können eigene Komponenten zur Verfügung gestellt werden. Das Framework kümmert sich dann um das Zusammenlinken der Parts selbständig. Auf diese Art und Weise werden also Komponenten (z.B. ein Tool Window oder ein Text Adornment) zur Verfügung gestellt. Auf der anderen Seite können mittels MEF Services importiert werden, die durch das Visual Studio bereitgestellt werden. Service Locator ade.

Listing 1: **GetService**

```
   //
   // Summary:
   //     Gets type-based services from the VSPackage service container.
   //
   // Parameters:
   //   serviceType:
   //     The type of service to retrieve.
   //
   // Returns:
   //     An instance of the requested service, or null if the service
could not be found.
   //
   // Exceptions:
   //   T:System.ArgumentNullException:
   //     serviceType is null.
   protected virtual object GetService(Type serviceType);
```

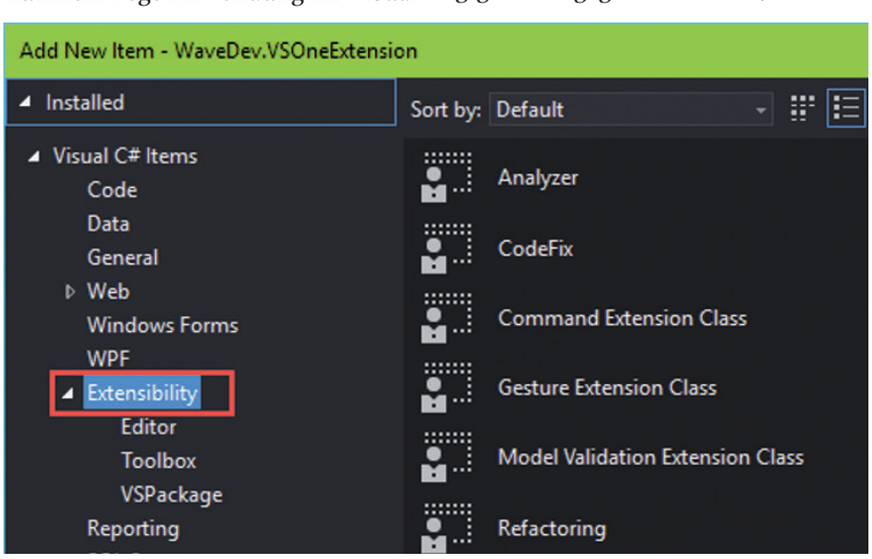


Abbildung 4

EIN BEISPIEL

Ein Beispiel sagt mehr als tausend Worte. Um die Möglichkeiten der Visual Studio Extensions zu verdeutlichen, soll in dem nun folgenden Abschnitt eine Erweiterung implementiert werden. Hinter dieser Erweiterung steckt ein realer Anwendungsfall, um die Produktivität bzw. die Sicherheit im Umgang mit bestimmten Code für den Entwickler zu erhöhen. Nehmen wir einmal an, dass unsere Anwendung mit personenbezogenen Daten umgeht. In der Hotelindustrie ist das beispielsweise keine große Besonderheit, seien es Reservierungs- oder Buchungssysteme, Schlüsselsysteme, Abrechnungssysteme, Multimedianetzwerke oder ein Bewertungssystem. Fast in jeder Software, die diese Systeme steuern und vernetzen, stecken personenbezogene Daten drin. Derartige Daten sind juristisch und moralisch hoch brisant. Geraten solche sensitiven Daten durch einen Bug oder ein Schlupfloch in die falschen Hände, kann das katastrophale Auswirkungen nach sich ziehen. Haftungsansprüche und Verlust des Kundenvertrauens

Listing 2: **Klassen für Kreditkarte und Gast**

```
internal class CreditCard
 {
     public int Id { get; set; }
     public string Number { get; set; }
     public DateTime ValidUntil { get; set; }
     public int VerificationNumber { get; set; }
 }

 internal class Guest
 {
     public int Id { get; set; }
     public string FirstName { get; set; }
     public string LastName { get; set; }
     public CreditCard CreditCard { get; set; }
 }
```

können sehr schnell zum Ende eines Unternehmens führen. Aus diesem Grund muss ich als Entwickler genau darauf achten, wann, wo und wie ich mit Daten in meinem Code hantiere. Und aus der Perspektive des Managements stellt sich die Fragen: wie stelle ich sicher, dass mein Entwicklerteam bewusst, gewissenhaft und sicher mit personenbezogenen Daten umgeht? Und wie kann das Team dabei durch Tooling und Prozesse unterstützt werden?

Schauen wir uns das folgende Beispiel einmal an. In der Problemdomäne gibt es so etwas wie einen Gast. Dieser soll durch einen Typen modelliert werden. Ein Gast verwendet zur Zahlung eine Kreditkarte. Wieder ein Objekt aus der Problemdomäne, die durch einen weiteren Typen dargestellt werden soll. Die daraus resultierende Klassenstruktur für die Kreditkarte und den Gast sind in Listing 2 abgedruckt.

Kreditkarteninformationen, Vor- und Nachname des Gastes stellen dabei die sensitiven Daten dar. Wann immer ein Entwickler nun mit Objekten von diesen Typen arbeitet, soll er auf die Sensitivität dieser Daten hingewiesen werden, so dass er entsprechende Sorgfalt walten lassen kann. Denn vielleicht gibt es ja vorgeschriebene Vorgehensweise, Richtlinien, Best Practices oder Architekturregeln, die das Team bzw. das Unternehmen definiert hat. Beispielsweise für den Umgang mit sensiblen Daten an Systemgrenzen bzw. wenn man diese überschreiten möchte. Und wieso auf den Code Review warten und nicht gleich die IDE in die Pflicht nehmen? Das spart Zeit und ist wesentlich zuverlässiger als das menschliches Auge. Genau hierbei soll die zu implementierende Visual Studio-Erweiterung helfen.

```
namespace RoomsReservationSystem
{
    0 references | 0 changes | 0 authors, 0 changes
    class Program
    {
        0 references | 0 changes | 0 authors, 0 changes
        static void Main(string[] args)
        {
            var creditCard = new CreditCard
            {
                Id = 1,
                Number = "560580100",
                ValidUntil = DateTime.Today.AddYears(1),
                VerificationNumber = 111
            };

            var guest = new Guest
            {
                Id = 1,
                FirstName = "Max",
                LastName = "Muster",
                CreditCard = creditCard
            };

            Console.WriteLine(guest.FirstName);
            Console.WriteLine(guest.CreditCard.Number);

        }
    }
}
```

Abbildung 5

Aufgabe der Erweiterung wird es sein, die Verwendung von sensitiven Daten/Typen im Codeeditor hervorzuheben. Abbildung 5 zeigt, wie das aussehen könnte. Der dort abgebildete Code verwendet beispielhaft Objekte der oben definierten Typen. Jede Verwendung eines solchen sensitiven Objektes im Code wird mit einer roten Box umrandet und damit markiert. Sei es durch Instanziierung, bei der Verwendung als lokale oder Instanzvariable, Zugriff auf einen Member oder bei der Zuweisung von Ausdrücken. Derart markiert, fallen solche kritischen Codeabschnitte sofort ins Auge. Das hilft dem Entwickler, und letztlich auch im Code Review.

DER VISUAL STUDIO EDITOR

Wie eingangs erwähnt, verfügt Visual Studio über eine breite Palette an Erweiterungspunkten. In diesem Artikel konzentrieren wir uns aber nur auf den Visual Studio Texteditor. Denn genau dort möchten wir den Entwickler unterstützen, wie im Beispiel gezeigt. Der Visual Studio Editor ist keinesfalls eine einfache Textbox mit ein wenig Syntax Highlighting – es steckt ein wenig mehr dahinter. Die Editorkomponente besteht aus verschiedenen Subsystemen. Für die generelle Architektur hat das Visual Studio-Team MVC gewählt. Welche Subsysteme zu welcher Schicht gehören, welche Aufgaben sie übernehmen und welche Erweiterungspunkte sie bieten ist in Abbildung 6 dargestellt. Das Model repräsentiert und verwaltet den zu Grunde liegenden Text, unabhängig von der Art des Textes oder seiner Darstellungen. In der Controller-Schicht werden einzelne Textabschnitte klassifiziert (z.B. Schlüsselwörter, Kommentare etc.). Zusätzlich werden für die verschiedenen Textklassen Formatierungen und Font-Eigenschaften definiert, die dann den entsprechenden Textabschnitten zugeordnet werden. Die View-Schicht wertet schließlich alle Informationen aus, um den Text entsprechend Rendern zu können. Zusätzlich zum Text stellt die UI den Cursor dar, rendert Selektionen von Text und den Rahmen um die Editor-Ansicht. Abhängig von Klassifikation und Inhalt, kann der View-Layer weitere UI-Elemente für einzelne Textbereiche anzeigen. Tooltips und Popups für z.B. IntelliSense und Autovervollständigung sind solche UI-Elemente. Diese basieren in Visual Studio ausschließlich auf der WPF. Mit Text- und View-Adornments können freie grafische Formen angezeigt werden (z.B. die rote Wellenlinie unter fehlerhaftem Code). Alle genannten Funktionalitäten können erweitert werden. Dazu bieten die Schichten Interfaces an, die von Erweiterungen implementiert werden müssen. Andere Interfaces dienen der Bereitstellung von Visual Studio-Services und können entsprechend in einer Erweiterung konsumiert werden. Ein paar der wichtigsten Interfaces

aus den einzelnen Schichten sind in den grauen Kästen in Abbildung 6 aufgeführt.

IMPLEMENTIERUNG

In unserem Beispiel werden wir ein Text-Adornment bauen, das den roten Rahmen bei kritischen Code erzeugt. Als Start dient ein Visual Studio Erweiterungsprojekt, das wir bereits kennengelernt haben. Diesem Projekt wird nun ein neues Element vom Typ „Editor Text Adornment" hinzugefügt. Visual Studio erzeugt dafür 2 Klassen: eine Plug-In-Komponente (TextAdornmentTextViewCreationListener) und eine Klasse, die das Adornment implementiert (TextAdornment). Werfen wir einen Blick (Listing 3) auf den generierten Code der Plug-In-Klasse (Xml-Dokumentation und Kommentare wurden für die Übersichtlichkeit entfernt)

Diese Klasse stellt eine Erweiterung für den Code-Editor dar. Um von Visual Studio als Plug-In erkannt zu werden, muss die Klasse zwei Bedingungen erfüllen. Sie muss als MEF-Part veröffentlicht werden und sie muss ein Interface implementieren, das die Schnittstelle für einen Erweiterungspunkt beschreibt. Das ExportAttribute auf der Klasse sorgt für die erste Bedingung und stellt die Klasse TextAdornmentTextViewCreationListener als MEF-Part zur Verfügung. Als Parameter wird dem Attribute eine Typ-Definition mitgegeben, der als Vertrag veröffentlicht wird. In diesem Fall wird der Vertrag durch das Interface IWpfTextViewCreationListener definiert. Folglich muss die Klasse dieses Interface auch implementieren. Das Interface IWpfTextViewCreationListener ist Teil der View-Schicht des Editors und befindet sich im Namespace Microsoft.VisualStudio.Text.UI.Wpf. Damit wäre dann auch die zweite Bedingung erfüllt. Die Klasse stellt eine vollwertige Erweiterung dar. Bleibt die Frage, welchen Erweiterungspunkt dieses Interface nun aber genau repräsentiert. Das Interface beschreibt einen Listener, der sich für das Erzeugen von Text Views, also von Codeeditoren, interessiert. Wann immer ein Codeeditor in Visual Studio geöffnet wird, informiert die IDE alle registrierten Zuhörer darüber. Dafür wird die einzige Methode aufgerufen, die in dem Interface definiert ist: TextViewCreated. Diese Methode bekommt als Eingabeparameter eine Referenz auf den neuen Text View übergeben. Dieser wird über das Interface IWpfTextView repräsentiert und erlaubt z.B. den Zugriff auf die dargestellten Textzeilen, Selektionen, die vorhandenen Adornments und Mouse-Events. Egal welche kreative Idee man auch immer für die Erweiterung des Editors hat, ab hier kann man nun richtig loslegen. Der Kreativität sind (fast) keine Grenzen gesetzt. Und unsere ist es, kritischen Code mit einem roten Rahmen zu versehen. Dieser Rahmen soll durch

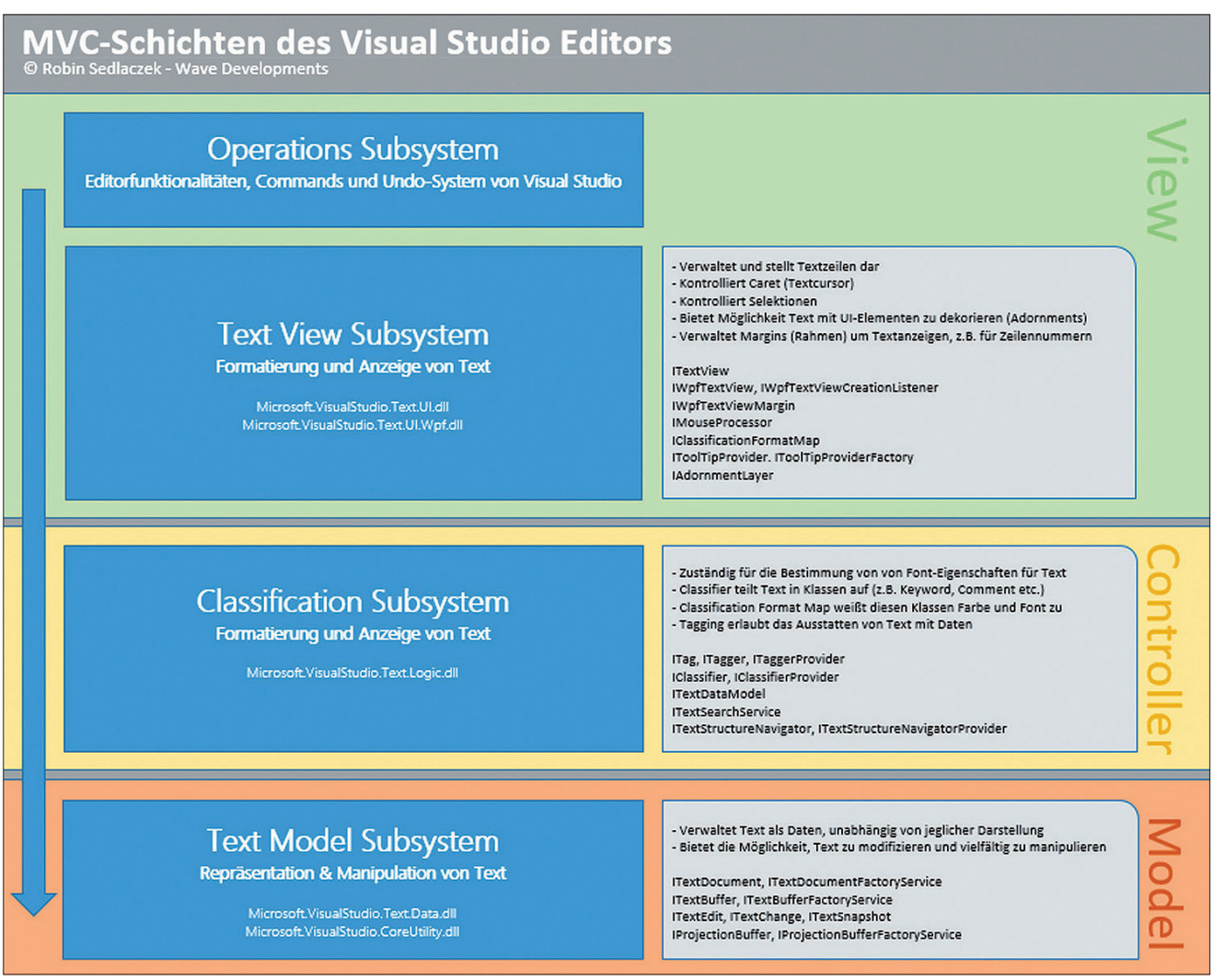


Abbildung 6

Listing 3: **Klassen für ein Text-Adornment**

```
using System.ComponentModel.Composition;
using Microsoft.VisualStudio.Text.Editor;
using Microsoft.VisualStudio.Utilities;

namespace WaveDev.SensitivCodeMarker.Adornments
{
    [Export(typeof(IWpfTextViewCreationListener))]
    [ContentType(„text“)]
    [TextViewRole(PredefinedTextViewRoles.Document)]
    internal sealed class TextAdornmentTextViewCreationListener :
IWpfTextViewCreationListener
    {
#pragma warning disable 649, 169

        [Export(typeof(AdornmentLayerDefinition))]
        [Name(„TextAdornment“)]
        [Order(After = PredefinedAdornmentLayers.Selection, Before =
PredefinedAdornmentLayers.Text)]
        private AdornmentLayerDefinition editorAdornmentLayer;

#pragma warning restore 649, 169

        #region IWpfTextViewCreationListener

        public void TextViewCreated(IWpfTextView textView)
        {
            new TextAdornment(textView);
        }

        #endregion
    }
}
```

ein Adornment umgesetzt werden. In der Implementierung der Methode TextViewCreated sehen wir, dass ein solches Adornment für jeden neuen View erzeugt wird. Damit kommen wir zur zweiten Klasse namens TextAdornment.

Die Klasse TextAdornment ist dafür zuständig, kritischen Code nach obiger Definition zu finden und diesen zu markieren. Und genau hier schicken wir nun das Tag-Team gemeinsam aufs Spielfeld. Zweiter Star auf der Bühne ist Roslyn, die Microsoft .NET Compiler Platform. Mit Roslyn sind wir in der Lage, Quellcode syntaktisch und semantisch zu analysieren und zu interpretieren. Perfekt für unseren Anwendungsfall. Somit sind wir faktisch uneingeschränkt in der Lage, kritischen Code zu finden. Dabei ist es egal, wie man „kritisch“ bzw. „sensitiv“ auf technischer Ebene definiert. Das ist ganz von der Problemdomäne (und der eigenen Kreativität) abhängig. Für das Beispiel in diesem Artikel soll dazu ein einfaches Interface genügen: ein sogenanntes Marker-Interface [8]. Wann immer ein Typ dieses Interface implementiert, markiert es sich damit als „sensitiv“. Das Interface hat in diesem Beispiel die Bezeichnung ISensitiveObject. Weiter oben wurde gezeigt, dass die Klassen Guest und CreditCard personenbezogene Daten, also sensitive Daten repräsentieren. Folglich markieren wir beide Typen mit dem Marker-Interface. (Listing 4)

Listing 4: **Klassen mit Interface**

```
public interface ISensitiveObject
{
    // ...
}

internal class CreditCard : ISensitiveObject
{
    // ...
}

internal class Guest : ISensitiveObject
{
    // ...
}
```

Mit dem Wissen, wie sensitive Typen identifiziert werden, können wir nun die semantische Analyse von Roslyn bemühen, um Verwendungen dieser Typen im Code zu finden. Werfen wir nun einen Blick in die Implementierung der Klasse TextAdornment.

Wir erinnern uns: in unserem Listener wird in der Methode TextViewCreated eine neue Instanz des Adornments erstellt. Übergeben wird der neue Text View. Dieser stellt das Event LayoutChanged zur Verfügung, auf das sich das Adornment registriert. Das Event wird immer dann gefeuert, wenn sich das Layout des Text Views verändert (z.B. durch Zoomen, Ändern der Fenstergröße oder Scrollen). Wenn dieser Fall eintritt, muss das Adornment die Position der kritischen Codes (Textabschnitte) neu berechnen und die grafische Darstellung anpassen. Die Ereignisbehandlungsroutine ist in Listing 5 dargestellt.

Zuerst sammeln wir mit der Hilfe von Roslyn alle sensitiven Codestellen in Form von Syntaxknoten. Die Syntaxknoten geben uns daraufhin Auskunft über ihre Position und Länge im zugrundeliegenden Text. Für die gefundenen Textabschnitte können wir schließlich die Bildschirmkoordinaten berechnen und entsprechende visuelle Darstellungen (rote Rahmen) erzeugen. Soweit der Ablauf. Schauen wir uns ein paar Details an.

Die Methode CollectSensitiveSyntaxNodes benutzt die semantische Analyse von Roslyn zum Auffinden von sensitiven Code im aktuellen Editor. Dazu wird der Text des aktiven Dokumentes (inklusive der Code-Dateien des Projektes und seiner Referenzen) zuerst durch den Compiler geschickt (Listing 6)

Listing 5: **OnLayoutChanged-Methode**

```
internal void OnLayoutChanged(object sender, TextViewLayoutChangedEventArgs e)
{
    var sensitiveSyntaxNodes = CollectSensitiveSyntaxNodes(e.NewSnapshot);
    var sensitiveTextSpans = GetTextSpans(sensitiveSyntaxNodes, e.NewSnapshot);

    CreateSensitiveCodeMarkerVisuals(sensitiveTextSpans, e.NewSnapshot);
}
```

Anschließend haben wir Zugriff auf den Syntaxbaum und das semantische Model des Codes und können uns diesen zu Nutze machen. Zunächst stellt sich die Frage, welche Code-Elemente geprüft werden

Listing 6: **Erstellen des Syntaxbaumes**

```
var compilation = CSharpCompilation
    .Create(„CodeInCurrentProject“)
    .AddReferences(references)
    .AddSyntaxTrees(trees);

var tree = currentDocument.GetSyntaxTreeAsync().Result;
var semanticModel = compilation.GetSemanticModel(tree);
var treeRoot = tree.GetRoot() as CompilationUnitSyntax;
```

sollen. Beispielsweise könnte eine Variablendeklaration einen sensitiven Typen verwenden. Oder der Typ eines Methodenparameters ist sensitiv. Im Prinzip kann sich hinter jedem Identifier (Bezeichner im Code) ein sensitiver Typ verbergen. Also prüfen wir auch jeden Identifier, den uns der Syntaxbaum liefert. (Listing 7)

Über das semantische Model versuchen wir dann den Typen eines Identifiers herauszufinden. Schließlich wird der gefundene Typ geprüft, ob dieser das Interface „ISensitiveObject“ implementiert. Dabei wird die gesamte Vererbungshierarchie berücksichtig, denn das Interface kann auch in einer Basisklasse implementiert sein. Ist der Check positiv, merken wir uns den aktuellen Syntaxknoten in einer Liste.

Die Liste der gefundenen Syntaxknoten wird dann der Methode GetTextSpans übergeben. Diese Methode berechnet die Textabschnitte für jeden Syntaxknoten in der Liste und liefert seinerseits eine Liste mit den Textabschnitten zurück. Diese definieren sich über ihre Startposition und Länge. An dieser Stelle haben wir den sensitiven Code dokumentenweit ausfindig gemacht und können genau sagen, welche Bereiche im Text sensitiv sind. Das ist genau das, was wir erreichen wollten. Mit diesen Informationen können wir nun Visuals erzeugen. Visuals sind visuelle Elemente, die schließlich vom Code-Editor gerendert werden.

Listing 7: **Prüfung der Identifier**

```
var identifiers = treeRoot.DescendantNodes().OfType<IdentifierNameSyntax>();

foreach (var identifier in identifiers)
{
    var typeInfo = semanticModel.GetTypeInfo(identifier);

    if (typeInfo.Type != null)
    {
        if (typeInfo.Type.AllInterfaces.Where(namedInterfaceType => namedInterfaceType.Name ==
„ISensitiveObject“).Any())
            sensitiveSyntaxNodes.Add(identifier);
    }
    else
    {
        var symbolInfo = semanticModel.GetSymbolInfo(identifier);

        if (symbolInfo.Symbol != null)
        {
            var namedType = symbolInfo.Symbol as INamedTypeSymbol;

            if (namedType != null && namedType.AllInterfaces.Where(namedInterfaceType => namedInterfaceType.Name
== „ISensitiveObject“).Any())
                    sensitiveSyntaxNodes.Add(identifier);
        }
    }
}
```

Listing 8: **Erstellen der Visuals**

```
foreach (var span in sensitiveTextSpans)
{
    IWpfTextViewLineCollection textViewLines = _view.TextViewLines;

    var geometry = textViewLines.GetMarkerGeometry(span);

    if (geometry != null)
    {
        var drawing = new GeometryDrawing(_brush, _pen, geometry);
        drawing.Freeze();

        var drawingImage = new DrawingImage(drawing);
        drawingImage.Freeze();

        var image = new Image
        {
            Source = drawingImage,
        };

        // Align the image with the top of the bounds of the text geometry
        Canvas.SetLeft(image, geometry.Bounds.Left);
        Canvas.SetTop(image, geometry.Bounds.Top);

        _layer.AddAdornment(AdornmentPositioningBehavior.TextRelative, span, null, image, null);
    }
}
```

Die Visuals werden in der Methode CreateSensitiveCodeMarkerVisuals (Listing 8) erzeugt. Dort werden für jeden Textabschnitt die Geometriekoordinaten bestimmt. Dabei hilft uns der Text View, also der View-Layer mit seinen Services. Mit den Koordinaten können dann die visuellen Elemente erzeugt und dem Adornment-Layer hinzugefügt werden. Um den Rest kümmert sich dann das Visual Studio und rendert die Adornments auf dem Bildschirm (siehe Abbildung 5).

Sicher ist die gezeigte Implementierung nicht optimal. Beispielsweise ist das Interface, welches als Marker für sensitiven Code benutzt wird, in dem Visual Studio-Plug-In hart-codiert. Wünschenswert wäre ein flexiblerer Ansatz. Mit jedem Layout-Change des Views wird auch der Code komplett durch den Compiler geschickt und die kritischen Textabschnitte werden jedes Mal neu berechnet. Das muss aber nicht immer sein, wenn sich z.B. der Code gar nicht geändert hat.

Der komplette Quellcode für die Visual Studio-Erweiterung steht auf GitHub [9] zur Verfügung. Auch ein Projekt zum Ausprobieren der Extension kann dort heruntergeladen werden [10].

FAZIT

Der Artikel hat gezeigt, wie vielseitig Visual Studio erweitert werden kann. Dazu wurde der Code-Editor etwas genauer unter die Lupe genommen und die Erweiterungspunkte wurden aufgezeigt. Zusammen mit Roslyn eröffnen sich undenkbar viele Anwendungsfälle und Einsatzmöglichkeiten, um die Produktivität im Entwickleralltag zu steigern, Entwicklungsprozesse zu beschleunigen und sicherer zu machen. An Hand eines Beispiels aus der realen Welt wurde dies anschaulich dargestellt und sollte dem Leser einen Eindruck von den Möglichkeiten verschaffen, die uns Erweiterungen für Visual Studio bieten.

QUELLEN

[1] https://visualstudiogallery.msdn.microsoft.com/

[2] https://github.com/dotnet/roslyn

[3] https://msdn.microsoft.com/en-us/library/bb166441.aspx?f=255&MSPPError=-2147217396

[4] https://visualstudiogallery.msdn.microsoft.com/2ddb7240-5249-4c8c-969e-5d05823bcb89

[5] https://msdn.microsoft.com/en-us/library/bb164709.aspx

[6] https://msdn.microsoft.com/en-us/library/ff648968.aspx

[7] https://msdn.microsoft.com/de-de/library/dd460648(v=vs.110).aspx

[8] https://en.wikipedia.org/wiki/Marker_interface_pattern

[9] https://github.com/robinsedlaczek/WaveDev.SensitiveCodeMarker

[10] https://github.com/robinsedlaczek/WaveDev.RoomsReservationSystemDemo

ROBIN SEDLACZEK hat mehr als fünfzehn Jahre Erfahrung als professioneller Softwareentwickler. Seit 7 Jahren tätig bei der Fairmas GmbH in Berlin, hat er das auf Finanzplanungstools spezialisierte Unternehmen mit aufgebaut. Als CTO ist er dort zuständig für den gesamten Entwicklungsbereich, beschäftigt sich mit der Produktentwicklung, Softwarearchitektur und -spezifikation, Technologieevaluierung, Prozess- und Teammanagement mit Ausrichtung auf die Microsoft .NET Plattform. In seiner Freizeit organisiert er die .NET User Group Berlin-Brandenburg und vermittelt sein Wissen auf Konferenzen, in User Groups, Onlinekursen, Fachartikeln und in seinem Blog *(http://robinsedlaczek.com/)*. Er ist erreichbar via Twitter *(https://twitter.com/RobinSedlaczek)* und per E-Mail *(robin.sedlaczek@live.de)*.

IMPRESSUM

Herausgeber
ppedv AG
Marktler Straße 15 b
84489 Burghausen
Vorstand: Hannes Preishuber
redaktion@visualstudio1.de
www.visualstudio1.de
Tel. +49 8960665044
Fax: +49 (0)8677 9889 4335

Chefredakteur (V.i.S.d.P.)
Hannes Preishuber
HannesP@ppedv.de

Redaktion
Martin Groblschegg
Chief Technical Officer
ppedv AG
MartinG@ppedv.at

Anzeigenleitung
Visualstudio1@ppedv.de
+49 8677 9889-60

Abobetreuung
Kathleen Ergezinger
KathleenE@ppedv.de
+49 8677 9889-60

Freie Mitarbeiter
Heiko Angermann
Olena Bochkor
Nicki Borell
Fabian Deitelhoff
Berndt Hamböck
Dr. Veikko Krypczyk
Benjamin Lanzendörfer
Naeem Ramzan
Dr. Michael Rath
Jan Schweda
Robin Sedlaczek
Remigiusz Suskiewicz
Daniel Wolf
Michael Zöhling

Layout/Satz: U. Ammann
Burghausen, ulli.am@web.de

Druck
Allgäuer Druck
Mehringer Straße 12
D-84489 Burghausen
Tel: +49 8677 9125-60
Fax: +49 8677 9125-65
info@allgaeuer-druck.de
www.allgaeuer-druck.de

Bildquellen
Seiten 1, 8, 10, 11, 12, 16, 20, 34, 55, 58: Fotolia
S. 39: venimo/Shutterstock
S. 13, 50: Microsoft

Bezugspreise
Einzelhefte:
8,50 EUR (D),
9,50 EUR (EU),
Jahresabo: 32 EUR (D),
35 EUR (EU)

Erscheinungsweise
4 x jährlich

Haftung
Für Fehler in Programmcode oder Text kommt eine Haftung nur bei grober Fahrlässigkeit in Betracht. Für unaufgefordert eingesandte Manuskripte, Datenträger, Produkte und Fotos wird keine Haftung übernommen.
Alle Pressemitteilungen wurden im Originaltext übernommen.

© 2016

Alle Rechte vorbehalten, Vervielfältigung nur mit Genehmigung der ppedv AG

Amtsgericht Traunstein:
HRB 12703

AUF DEN KUNDEN ZUGEHEN

Der Erfolg einer IT-Firma misst sich wesentlich daran, wie gut die Problemdomäne des Kunden verstanden wird. Business-Applikationen müssen dabei oft die Geschäftsprozesse abbilden und sollen somit helfen bisher manuell durchgeführte Vorgänge automatisiert zu erledigen. In der Anforderungsanalyse müssen Sie sich daher mit den Geschäftsvorgängen des Kunden beschäftigen, man muss seine „Sprache" sprechen!

Um dieses möglichst gut zu meistern existieren Hilfsmittel und methodische Werkzeuge, welche sowohl von den Mitarbeitern der Fachabteilungen, als auch von den IT-Entwicklern verstanden werden. Zu nennen sind beispielsweise: Ereignisgesteuerte Prozesskette (EPK), Swimlane und Diagramme aus der UML. Mit deren sicherer Beherrschung kann man den Kunden im Beratungsgespräch beeindrucken und noch viel wichtiger: Man findet eine gemeinsame Diskussionsebene.

GESCHÄFTSPROZESS UND GESCHÄFTSPROZESSMANAGEMENT

Unter einem Geschäftsprozess wird eine Folge von inhaltlich abgeschlossenen Aktivitäten verstanden, welche in einem logischen Zusammenhang stehen. Unter Zuhilfenahme von Ressourcen und eingehenden Informationen

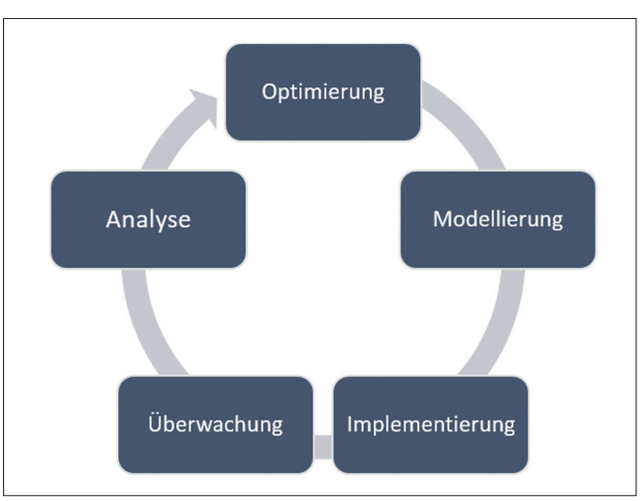


Abbildung 1: Lebenszyklus von Geschäftsprozessen [2].

Ebene	Tätigkeit	Ergebnis	Akteur
strategische Ebene	Strategieentwicklung	Geschäftsfeldstrategie	strategisches Management
fachlich-konzeptionelle Ebene	Prozessmodellierung	Prozessmodell	Geschäftsprozessmanagement
operative Ebene	Workflowmodellierung	Workflowmodell	Workflowmanagement

Tabelle 1: Ebenenkonzept nach Gehring [3]

werden diese durch Menschen oder Maschinen mit dem Ziel ein bestimmtes Ergebnis zu erzeugen ausgeführt.
Ein Geschäftsprozess hat einen definierten Anfang und ein definiertes Ende und kann einen funktions-, hierarchie- und standortübergreifenden Ablauf haben. In den letzten Jahren hat die Bedeutung des Geschäftsprozessmanagements im Unternehmen stark an Bedeutung zugenommen. Dieses beinhaltet Methoden, Konzepte und Technologien für die Analyse, das Design, die Implementierung und die Steuerung operativer Geschäftsprozesse [1]. Die Prozesse werden auf diese Weise analysiert, optimiert, automatisiert und transparent gemacht. Die zugrunde liegende Theorie ist ein Teilgebiet der Wirtschaftsinformatik. Der Lebenszyklus von Geschäftsprozessen kann anhand folgender Phasen erklärt werden (Abbildung 1)[2]:

- *Modellierung:* In dieser Phase werden Prozesse ausgewählt, definiert und optimal beschrieben.
- *Implementierung:* Die Geschäftsprozesse werden aufgenommen, dokumentiert und in eine Prozesslandschaft implementiert.
- *Überwachung:* Hier erfolgen die Überwachung und die Visualisierung der Prozesse.
- *Analyse:* Nun werden die in der Überwachungsphase ermittelten Daten ausgewertet. Die Bewertung orientiert sich an den Unternehmenszielen und -strategien.
- *Optimierung:* Aus den gewonnenen Erkenntnissen werden die Verbesserungsmaßnahmen und die Vorgaben für die Gestaltung der neuen Geschäftsprozesse abgeleitet. Die Folge ist die Anpassung der Prozesse und somit eine erneute Design- bzw. Modellierungsphase.

GESCHÄFTSPROZESS VS. WORKFLOW

Wie bereits erwähnt, ist der Geschäftsprozess eine zielgerichtete, zeitlich abgestimmte und logisch zusammenhängende Abfolge von Aufgaben und Aktivitäten, welche arbeitsteilig von mehreren Organisationseinheiten unter Nutzung von Informations- und Kommunikationstechnologien ausgeführt werden. Dagegen ist ein Workflow ein formal beschriebener, ganz oder teilweise automatisierter Geschäftsprozess. Dazu gehören zeitliche, fachliche und ressourcenbezogenen Spezifikationen, welche für eine automatische Steuerung des Arbeitsablaufs auf operativer Ebene erforderlich sind. Je nach Grad der Strukturierung unterscheidet man zwischen dem allgemeinem, dem fallbezogenem und dem ad hoc Workflow. Im Fokus der Geschäftsprozessmodellierung stehen somit die betrieblichen Systeme und Strukturen, bei der Workflow-Modellierung die Informationssysteme und Anwendungen im Vordergrund (Tabelle 1) [3].

GESCHÄFTSPROZESS-MODELLIERUNG

Unter der Geschäftsprozessmodellierung versteht man ein Unterstützungsinstrument des Prozessmanagements, welches die Konstruktion, Wartung und Anwendung konzeptionelle Modelle und Geschäftsabläufe von Unternehmen umfasst [4]. Je nach Autor, wird Geschäftsprozessmodellierung von verschiedenen Sichten betrachtet.

Wie in Tabelle 2 ersichtlich, existieren viele Möglichkeiten Geschäftsprozesse zu modellieren. Nachfolgend geben wir einen Überblick über die wichtigsten Ansätze [3]:

Scheer	Ferst/Sinz	Gehring	Gadatsch
Organisationssicht	Leistungssicht	Organisationssicht	Prozesssicht
Funktionssicht	Lenkungssicht	Funktionssicht	Organisationsstruktursicht
Datensicht	Ablaufsicht	Datensicht	Aktivitätsstruktursicht
Steuerungssicht			Applikationsstruktursicht
Leistungssicht			Informationsstruktursicht

Tabelle 2: Sichten auf Geschäftsprozessmodellierung [3]

- *Organisationssicht* beschreibt die Struktur und die Beziehung von Organisationseinheiten. Somit soll die Aufbauorganisation in reduzierter Komplexität dargestellt werden.
- *Datensicht* beschreibt Informationsobjekte und deren Beziehung zueinander. Die Modellierung erfolgt mit einem Entity-Relationship-Modell (ERM).
- *Funktionssicht* wird zur Beschreibung betriebswirtschaftlicher Funktionen und deren Beziehung zueinander verwendet. Die einzelnen Funktionen können in einem Funktionsbaum dargestellt werden.
- *Leistungssicht* beschreibt materielle und immaterielle Input- und Output-Leistungen einschließlich der Geldflüsse. Dieses kann anhand eines Produktmodells dargestellt werden.
- *Steuerungssicht* übernimmt die Integration der anderen Teilschichten. Das Wertschöpfungskettendiagramm (WKD), die ereignisgesteuerte Prozesskette (EPK) und das Vorgangskettendiagramm (VKD) sind dafür geeignet.

EREIGNISGESTEUERTE PROZESSKETTEN (EPK)

Ereignisgesteuerte Prozessketten (EPK) sind aufgrund der anschaulichen und intuitiven Repräsentation, sowie einer durchgängigen Werkzeugunterstützung als Modellierungsinstrument in der Praxis weit verbreitet. Die EPK ist ein gerichteter Graph, welcher aus passiven und aktiven Komponenten besteht. Passive Komponenten repräsentieren den Zustand eines Informationsobjekts, sind Ergebnisse von Funktionen und gleichzeitig lösen sie diese aus. Anderes ausgedrückt, beschreibt das Ereignis das Eintreten eines betriebswirtschaftlichen Zustandes. Jeder Geschäftsprozess beginnt mit einem Startereignis und endet mit einem Endereignis. Aktive Komponenten besitzen Entscheidungskompetenz; es sind Funktionen. Funktionen verbrauchen

Ressourcen und Zeit und beschreiben was nach einem auslösenden Ereignis passiert. Bei der Beschreibung der Funktionen sollen Verben verwendet werden. Die weiteren Komponenten zur Modellierung des Kontrollflusses sind Verknüpfungsoperatoren. Die drei logischen Operatoren ermöglichen es Verzweigungen zwischen Ereignissen und Funktionen bzw. umgekehrt einzufügen:

- „UND“ bedeutet, dass alle Ereignisse eintreten müssen, damit der Kontrollfluss fortgesetzt wird.
- „ODER“ steht dafür, dass mindestens ein Ereignis oder eine Funktion eintreten muss, damit der Kontrollfluss fortgesetzt wird.
- „exklusives ODER“ heißt, dass genau ein Ereignis eintreten muss, damit der Kontrollfluss fortgesetzt wird.

Ereignisse, Funktionen und Verknüpfungsoperatoren werden mithilfe von Kontrollflusskanten verbunden. Dies wird mithilfe eines gerichteten Pfeils abgebildet. Der Kontrollfluss legt die logische und zeitliche Reihenfolge fest. Mittels Operatoren kann dieser in mehrere Kontrollflüsse aufgespaltet werden.
Bei einer Abwägung von Vor- und Nachteilen von EPKs, überwiegen die positiven Eigenschaften [5].

Vorteile:

- EPKs sind verständlich und mit wichtigen anderen Notationen, wie z.B. dem Funktionsbaum oder dem Klassendiagramm kombinierbar.
- EPKs bieten die Möglichkeit einer detaillierten Darstellung unter vielen unterschiedlichen Aspekten. Ebenso erlauben sie eine durchgängige Betrachtung von der Organisation bis zur IT.
- EPKs sind eine gute Grundlage zum Testen von Sollkonzepten.
- Anhand von EPKs können die Prozesse anwendungsunabhängig und über verschiedene Systeme im Unternehmen dargestellt werden.
- Es existieren zahlreiche Werkzeuge und Tools zum Erstellen von EPKs.

Nachteile:

- Für EPKs existieren keine Standards: Je komplexer die dargestellten Prozesse, desto höher ist der Aufwand zur Erstellung und Schulung der Modellierer.
- Der Fokus liegt auf der Dokumentation und nicht auf der Ausführung der Prozesse.
- Der Aufwand zur Auswahl der passenden Modellierungsinstrumente ist sehr hoch.

In der Praxis sollte auf die Einhaltung folgender allgemeiner Regeln geachtet werden [6]:

- Symbole sind ausschließlich reserviert für: Ereignisse, Funktionen, Prozesswegeweiser, logische Verknüpfungsoperatoren, Organisationseinheiten und Informationsobjekte.
- Ein EPK kann nur mit einem Ereignis oder mit einem Prozesswegweiser beginnen oder enden.
- Alle Symbole müssen verbunden sein.
- Ein EPK muss mindestens eine Funktion enthalten.
- Ereignisse können nicht direkt mit Ereignissen verbunden werden.

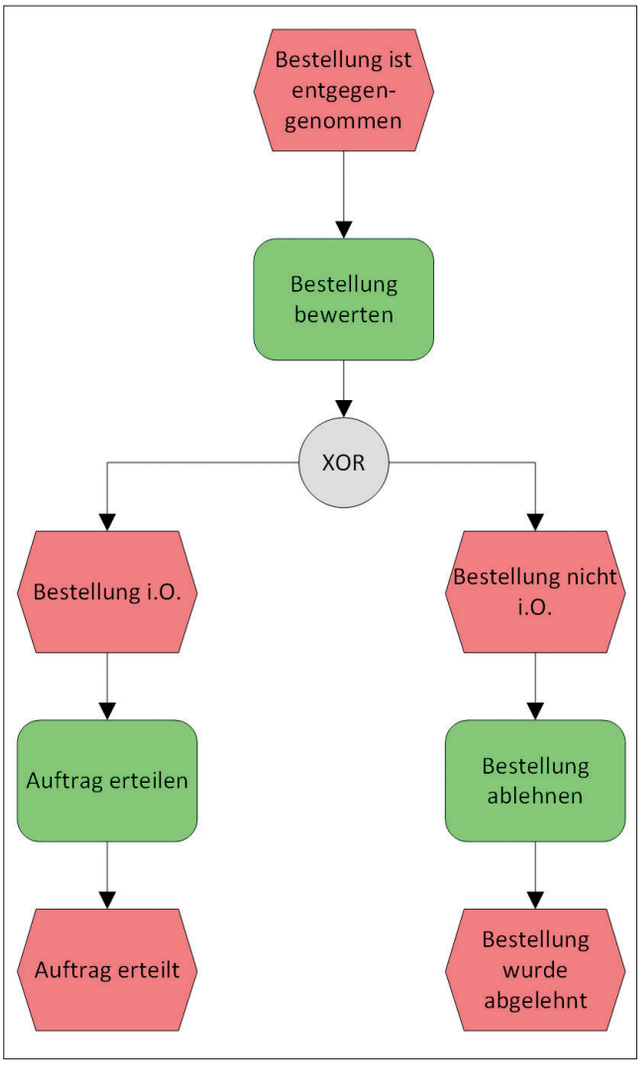


Abbildung 2: Ereignisgesteuerte Prozesskette

- Funktionen können nicht direkt mit Funktionen verbunden werden.

Abbildung 2 zeigt ein einfaches Beispiel für eine EPK. Eine Swimlane ist eine Modifikation des EPKs. Dessen Bezeichnung erinnert wegen der Darstellungsform an die Bahnen eines Schwimmbades. Die Swimlane zeigt die Abfolge einzelner Schritte. Die Logik der Schrittfolge als auch das „Springen“ des Prozesses zwischen den beteiligten Organisationseinheiten ist gut erkennbar. Vorteilhaft gegenüber der EPK ist ihre höhere Genauigkeit und größere Aussagekraft [7].

IT INFRASTRUCTURE LIBRARY (ITIL)

Unter der ITIL wird eine Sammlung von vordefinierten und standardisierten Prozessen, Funktionen und Rollen verstanden. Anderes ausgedrückt, ist die ITIL eine „Library“ von Best Practices für das IT Service Management, welche Regeln zu Aufbau- und Ablauforganisation und die dafür benötigten Werkzeuge im IT-Dienstleistungsbereich beschreibt. Mittlerweile ist ITIL weit verbreitet und hat den Einzug in das Tagesgeschäft gefunden. Die ITIL besteht aus fünf Haupt-Publikationen, welche gemeinsam den ITIL Service-Lebenszyklus abbilden [8]:

- Servicestrategie
- Service Design

- Service Überführung
- Servicebetrieb
- Kontinuierliche Serviceverbesserung.

Die Einführung verläuft in mehreren Phasen, die aufeinander aufgebaut sind. Das Projekt beginnt mit der Konzeption der Abläufe und Verfahren, danach werden die konkreten Schritte zur Einführung definiert:

Schritt 1: Das ITIL-Projekt vorbereiten

Schritt 2: IT-Services definieren

Schritt 3: ITIL-Rollen auswählen und besetzen

Schritt 4: Ist-Prozesse analysieren

Schritt 5: Soll-Prozess-Struktur entwickeln

Schritt 6: ITIL-Prozess-Schnittstellen definieren

Schritt 7: Prozess-Controlling einrichten

Schritt 8: ITIL-Prozesse im Detail ausarbeiten

Schritt 9: Anwendungssysteme auswählen und implementieren

Schritt 10: Prozess-Implementierung und Schulung

Im Internet findet man viele hilfreiche Informationen rund um ITIL und ISO 20000, z.B. die komplette Beschreibung der ITIL-Prozesse, ein navigierbares ITIL-Glossar, ITIL-Checklisten, die Definition von ITIL-Rollen u.v.m.

UNIFIED MODELING LANGUAGE (UML)

Die UML definiert eine allgemein verwendbare Modellierungssprache. Sie stellt Diagramme und Notationselemente zur Verfügung, mit deren Hilfe sowohl statische wie auch dynamische Aspekte beliebiger Anwendungsgebiete modelliert werden können. Sie beinhaltet mehrere Diagramarten [9].

- Zur Definition der Anforderungen: Anwendungsfalldiagramme
- Zur Definition der Struktur: Klassendiagramme, Paketdiagramme und Kollaborationsdiagramme
- Zur Abbildung der Abläufe: Sequenzdiagramme und Zustandsdiagramme
- Zur Implementierung: Komponentendiagramme und Einsatzdiagramme.

UML Diagramme sind durch ihre Eindeutigkeit, Verständlichkeit, Ausdrucksstärke, Plattform- und Sprachunabhängigkeit bekannt. Es folgen Ausführungen zum Anwendungsfalldiagramm und Aktivitätsdiagramm, da beide Diagrammtypen insbesondere bei der fachlichen Modellierung verwendet werden.

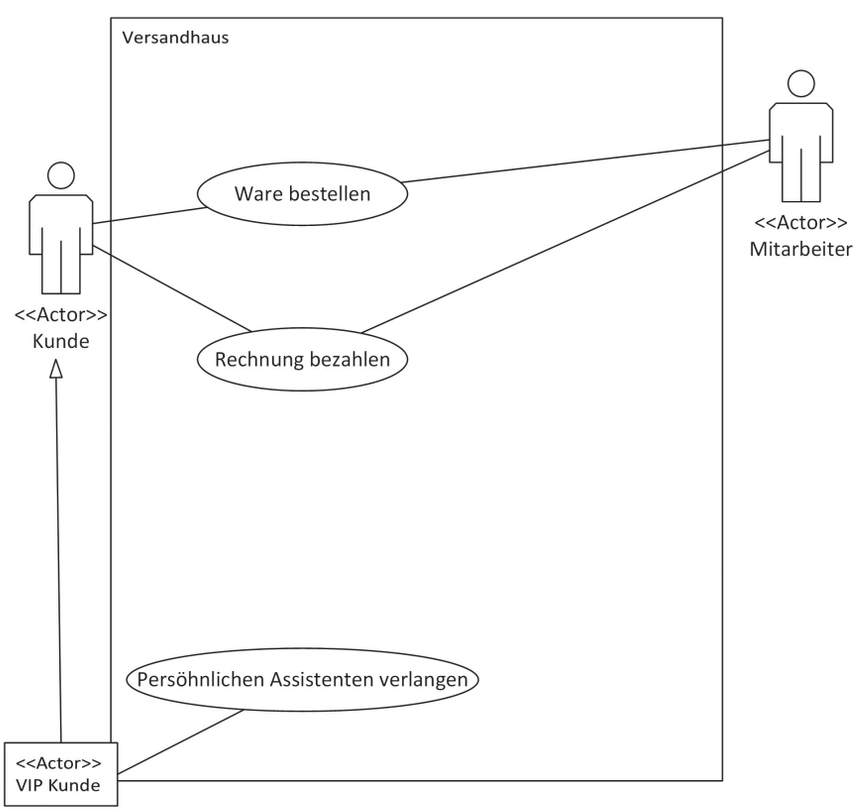


Abbildung 3: Anwendungsfalldiagramm zeigt die wichtigsten Funktionen einer Software.

Anwendungsfalldiagramme (Use Case Diagrams) modellieren die Funktionalität des Systems auf einem hohen Abstraktionsniveau aus der Black-Box-Sicht des Anwenders (Abbildung 3). Durch die Modellierung wird beschrieben, welche Funktionen das System anbietet. Die Notationselemente sind:

- *Systemgrenze* umfasst das System, das die benötigten Anwendungsfälle bereitstellt und mit dem die Anwender interagieren.
- *Akteur* modelliert ein Typ oder eine Rolle, die ein externer Benutzer oder ein externes System während der Interaktion mit dem System einnimmt.
- *Anwendungsfall* spezifiziert eine abgeschlossene Menge von Aktionen, die von einem System bereitgestellt werden und einen erkennbaren Nutzen für einen oder mehreren Akteure erbringen.
- *Assoziation* modelliert in Anwendungsfalldiagrammen eine Beziehung zwischen Akteuren und Anwendungsfällen.
- *Generalisierung/Spezialisierung*
- *Include-* und *Extend-Beziehung.*

Aktivitätsdiagramme sind in allen Phasen der Softwareentwicklung einsetzbar: während der Analyse/Definition-Phase, in der Entwurf/Design Phase, bei der Implementierung und in der Test-Phase. Die Bestandteile sind Aktivitätskanten und Aktivitätsknoten. Aktivitätskanten modellieren gerichtete Verbindung zwischen den Aktivitätsknoten und legen die Reihenfolge des Ablaufs eines Aktivitätsdiagramms fest. Die Aktivitätsknoten können in drei Gruppen unterteilt werden (Abbildung 4):

1. Ausführbare Knoten führen die Tätigkeiten aus.
2. Objektknoten sind alle Notationselemente, welche Daten speichern können.
3. Kontrollknoten koordinieren den Ablauf zwischen anderen Knoten und bestimmen den Ablauf von Aktivitätsdiagrammen.

FAZIT UND AUSBLICK

Als IT-Servicedienstleister ist die Fachabteilung unser Kunde. Damit ist die Rollenverteilung klar, d.h. wir müssen uns mit den Problemen des

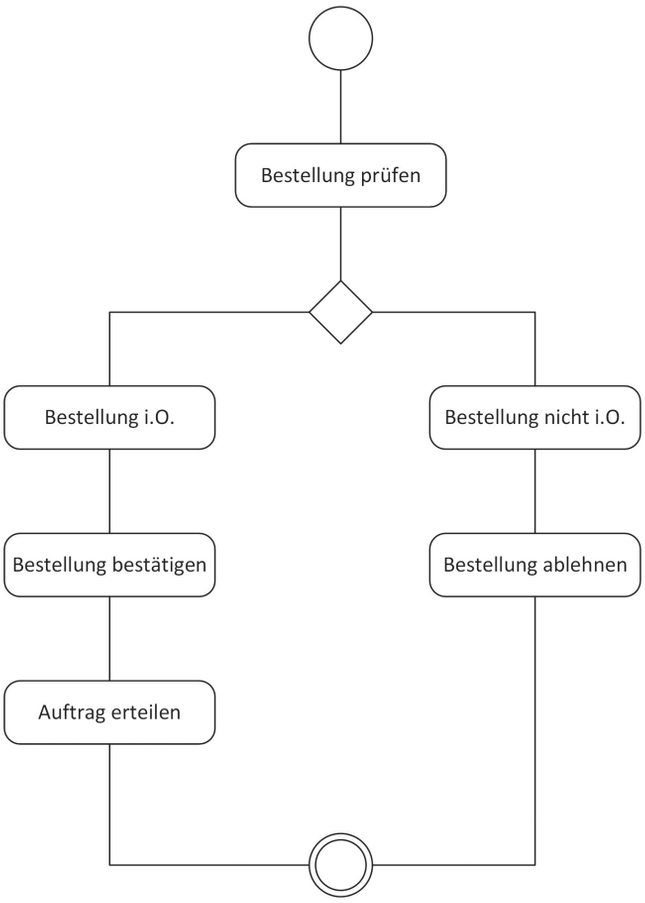


Abbildung 4: Abläufe lassen sich mithilfe eines Aktivitätsdiagramms darstellen

Kunden exakt beschäftigen und deren Geschäftsprozesse bestmöglich verstehen. Gelegentlich kann es dabei sogar zu Synergieeffekten kommen, in dem die Digitalisierung der Abläufe auch zu einer Verbesserung dieser führen kann. Die hier vorgestellten grafischen Visualisierungsmethoden sind dafür bestens geeignet für eine reibungslose Kommunikation zwischen den Beteiligten zu sorgen.

LITERATUR UND LINKS

[1] http://wirtschaftslexikon.gabler.de/Definition/geschaeftsprozessmanagement.html

[2] http://www.manager-wiki.com/unternehmensanalyse/84-business-process-management-bpm

[3] http://winfor.uni-wuppertal.de/fileadmin/wolff/Downloads/Hauptstudium/WIModul1/IKS/IKS-Geschaeftsprozessmodellierung.pdf

[4] http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/Geschaftsprozessmodellierung

[5] http://www.ruffing-consult.de/wp-content/uploads/2014/08/%C3%9Cbersicht-Notationen-zur-Prozessmodellierung.pdf

[6] Baumgärtner, H., u.a.: Regeln zur Modellierung von ereignisgesteuerten Prozessketten

[7] Vahs, D.: Organisation. Das Lehr- und Managementbuch, Schäffer Poeschel, 2009

[8] http://de.it-processmaps.com/itil/ueber-itil.html

[9] Kecher, C.: UML 2. Das umfassende Handbuch, Galileo Computing, 2011

DR. VEIKKO KRYPCZYK
studierte und promovierte Betriebswirtschaftslehre mit dem Schwerpunkt Wirtschaftsinformatik.
Er arbeitet als Fachautor und ist begeisterter Programmierer

OLENA BOCHKOR
studierte Betriebswirtschaftslehre u.a. mit dem Schwerpunkt Wirtschaftsinformatik. Weitere Informationen zu diesen und anderen Themen der IT finden Sie unter http://it-fachartikel.de.

Finally: MCSA Windows Server 2016

9 Tage
8—20 Uhr
inkl. Prüfungen
Trainer: Torsten Fischer

Voranmeldung unter:
minerva@ppedv.de oder
030—201 430 41

Jetzt schon Platz sichern!

IT-Akademie MINERVA
Weitere Informationen unter:
www.minerva.ppedv.de

E-COMMERCE MIT SMARTSTORE.NET

Bewertung des durch Microsoft vertriebenen Shop-Systems

Der Versandhandel hat während des letzten Jahrzehnts einen drastischen Wandel erlebt. Durch die steigende Anzahl verfügbarer Medien bieten Händler ihre Produkte mittlerweile auf mehreren Kanälen (*Multi-Channel, Multi-Channel-Publishing*) an. Dies hat den Vorteil, dass viele Kunden erreicht werden und diese maximal personalisiert angesprochen werden können. Da gedruckte Kataloge beinahe der Vergangenheit angehören, stiegen die durch digitale Katalogsysteme (öfters bezeichnet als *E-Commerce, Web-Shop*, oder Shop-System) erzielten Umsätze explosionsartig an. Laut Statista.com konnte der Onlinehandel zwischen 2005 und 2015 ein Umsatzplus von 302,17 % erzielen. Microsoft hat diesem Trend widersprochen, indem es die Entwicklung des "Microsoft Commerce Server" vor ein paar Jahren abgegeben hat. Jedoch bewirbt und unterstützt Microsoft auf der offiziellen Webseite sowie auf der eigenen *Open-Source* Plattform "Codeplex" Shop-Systeme, welche auf .NET-Technologien basieren. Der renommierteste Vertreter für den deutschsprachigen Raum ist hierbei "SmartStore.NET". Dieser Artikel durchleuchtet die mittlerweile stark verbesserte Anwendung hinsichtlich marktrelevanter und marktaktueller Anforderungen. Im Anschluss wird das System mit anderen renommierten *E-Commerce* Anwendungen mittels eines neutralen Kriterienkataloges verglichen. Die Analyse zeigt, dass sich die .NET-Lösung auf Augenhöhe mit dem Klassenprimus "Magento" befindet und eine ausgereifte Anwendung für verschiedene Anwendungsszenarien darstellt, insofern Abstriche im Design möglich sind.

1. HINTERGRUND

Die im Jahre 1999 gegründete "SmartStore AG"[1] (Dortmund, Deutschland) entwickelt *E-Commerce* Lösungen für kleine und mittelgroße Onlinehändler. Im Juni 2013 wurde die erste Version des "SmartStore.NET" auf der durch Microsoft betriebenen *Open-Source* Plattform "Codplex"[2] vorgestellt. Mittlerweile wurde das System mehrfach überarbeitet und hat dessen Vorgänger "SmartStore.biz" endgültig abgelöst. Das System greift hierbei auf renommierte Technologien hinsichtlich *Backend* und *Frontend* Funktionalitäten zurück, z.B. "Model-View-Controller 4", "ASP.NET", "Entity Framework 5", "Twitter Bootstrap" und "LESS CSS". Hauptmerkmale in der Entwicklung wurde auf eine einfache Bedienbarkeit gelegt, die Unterstützung von sozial-medialen Anforderungen wie etwa ein interner *Blog* oder die Möglichkeit Produkte aktiv bewerten zu können, sowie das Angebot eines durch Metadaten (Kategorien, Schlagworte, Eigenschaften, Inhaltstypen, Medien, Preise, Dateiformate, etc.) unterstützten Suchcenters. Das System stellt Basis-Vorlagen bereit, um auch für *Multi-Channel-Publishing* (Deutsch: Publizieren auf mehreren Kanälen), gewappnet zu sein. Multi-Shop (Deutsch: Mehrere (Sub)-Shops, z. B. für die Trennung B2B und B2C) ist nicht möglich. Hinsichtlich Support kann der potentielle Shop-Betreiber zwischen vier verschiedenen Paketen variieren. Das kostenfreie Support-Programm "Basic" ermöglicht den Zugang zu einer Online-Community und eines *Support*-Forums, sowie das Bereitstellen

von regelmäßigen Sicherheitsupdates und verschiedenster Demo-Shops[3].

Um zu durchleuchten, was das auf .NET-basierende Shop-System kann und ob es mit seinen Konkurrenten auf Augenhöhe ist, bewerten wir "SmartStore.NET" auf Basis eines neutralen Kriterienkataloges. Dieser entstand aus einem Vergleich aktueller Web-Shop Anwendungen, welche bei deutschen Onlinehändlern überwiegend Verwendung finden (Intershop[4], Heiler[5], Demandware[6], Hybris[7], Magento[8], Commerce Server[9], Strato[10], xt:Commerce[11], Omeco[12], Websale[13]). So erlaubt der aus verschiedenen Kriterien bestehende Katalog einen aussagekräftigen Vergleich zwischen den Systemen:

- ***System und Performanz*** bewertet serverseitige Eigenschaften:
  - **Daten Schnittstellen** zur Bewertung unterstützter Import/Export Formate "Extensible Markup Language" (XML) und "Structured Query Language" (SQL). Ferner werden Schnittstellen zu Systemen "Customer Relationship Management" (CRM), "Enterprise Resource Planning" (ERP) und "Product Information Management" (PIM) untersucht.
  - **Modul Erweiterungen** zur Evaluierung, ob kostenlose Erweiterungen (Englisch: *Plug-Ins*) angeboten werden. Diese können eingesetzt werden, um z. B. weitere Zahlverfahren oder Mehrsprachigkeit zu unterstützen.
  - **Generelle Kriterien** um enthaltene Testumgebungen (A/B Testing) und Sicherheitsfunktionen wie "Secure Sockets Layer" (SSL) festzustellen.
- ***Backend Funktionen*** bewertet für das Marketing relevante Eigenschaften:
  - **Editierbarkeit** um die Einfachheit hinsichtlich der Bearbeitung von Inhalt und Design zu ermitteln sowie ob Funktionen hinsichtlich *Multi-Channel-Publishing* unterstützt werden.
  - **Bestell Administration** bewertet die Funktionen hinsichtlich des Bestellprozesses. Beispielsweise ob verschiedene Zahlverfahren unterstützt werden, eine Lagerverwaltung und die Verwaltung von Retouren berücksichtigt ist, oder ein Mahnwesen angeschlossen ist.
  - **Kampagnen Administration** bewertet z.B. die Fähigkeiten hinsichtlich personalisierter Kampagnen, oder das Anbieten alternativer Produkte.
- ***Frontend Funktionen*** bewertet für den Kunden relevante Eigenschaften:
  - **Sucherfahrung** analysiert Eigenschaften, um das Suchcenter zu verbessern, z.B. mit Hilfe von Filterungstechniken (Kategorie, Preis, etc.).
  - **Soziale Aktivitäten** analysiert die sozial-medialen Fähigkeiten des Shop-Systems, z.B. ob ein *Blog*/Form integriert ist, oder Produkte vom Kunden bewertet und kommentiert werden können.

2. SMARTSTORE.NET

2.1 SYSTEM UND PERFORMANZ

2.1.1 Daten Schnittstellen

Als Datenbanken werden der "Microsoft Compact Server" sowie der "Microsoft SQL Server" unterstützt. Importiert und exportiert werden kann das Format "Microsoft

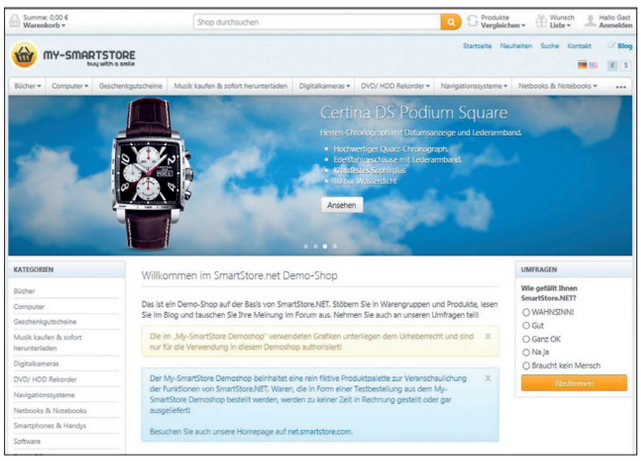


Abbildung 1: "SmartStore.NET" Demo-Shop Frontend User Interface.

Excel". Inhalte von "SmartShop.biz" können auch importiert werden. XML Dateien können lediglich exportiert werden, z.B. zum Herunterladen des Produktkataloges. Schnittstellen zu anderen Managementsystemen werden nicht unterstützt, da das Shop-System bereits selbst über die grundlegenden CMS und CRM Funktionalitäten verfügt.

2.1.2 Modul Erweiterungen

Viele kostenfreie *Plug-Ins* werden angeboten, um die Shop-Funktionalitäten zu verbessern. Beispielsweise werden *Backend* Erweiterungen angeboten, um länderspezifische Discounts zu realisieren, oder um die angebotenen Zahlverfahren zu erweitern. Die *Frontend* Erweiterungen erstrecken sich von einer Schnittstelle zu weiteren Preisportalen (z.B. "Billiger.de") bis hin zur Realisierung eines *Live*-Chats.

2.1.3 Generelle Kriterien

Das System bietet regelmäßige Sicherheitsupdates an, jedoch keine Testumgebung.

2.2 BACKEND FUNKTIONEN

2.2.1 Editierbarkeit

Keine Kenntnisse hinsichtlich Programmier- oder Seitenbeschreibungssprachen sind für die Verwaltung von Inhalten (z.B. Kategorisierung von Inhalten), deren Editierung (z.B. Löschen von Text) oder die Modifikation von Texten (z.B. Ändern von Farbe oder Schriftschnitt) nötig. Die Browseroberfläche ist analog der "Microsoft Word" Oberfläche aufgebaut und verspricht dadurch ein rasches

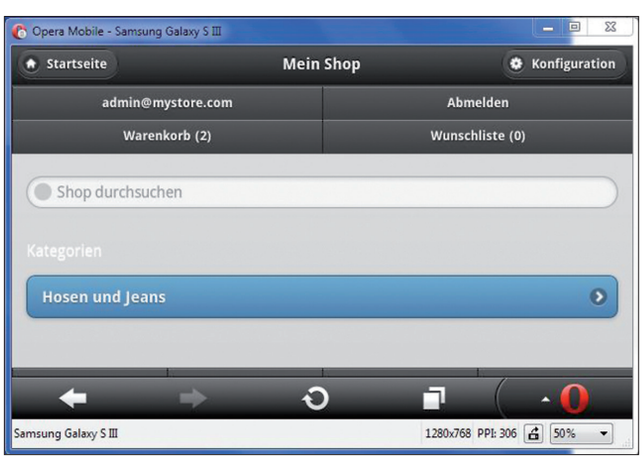


Abbildung 2: "SmartStore.NET" Demo-Shop Frontend User Interface ("Mobile").

	Kriterium	SmartStore.NET	Magento	xt:Commerce
Daten Schnittstellen				
Import/Export	XML, SQL	✓	✓	✓
System Schnittstellen	ERP, CRM, PIM	×	×	×
	Preisportale	✓	✓	✓
Modul Erweiterungen				
Plug-Ins	Plug-Ins	✓	✓	✓
	Skalierbarkeit	✓	✓	✓
Mehrsprachigkeit	Verschiedene Sprachen	✓	✓	✓
	Verschiedene Währungen	✓	✓	✓
Generelle Kriterien				
Sicherheit	Updates	✓	×	✓
	Testumgebung	×	✓	×

(✓ = unterstützt, oder mittels kostenfreier Plug-Ins unterstützt; ×=Nicht unterstützt).

Tabelle 1: "SmartStore.NET" System und Performanz; Vergleich von Open-Source Anwendungen

Einlernen in das System. Technisch versiertere Benutzer können zur *Code-View* wechseln und so direkt auf den "Hypertext Markup Language" (HTML) und "Cascading Style Sheet" (CSS) Quellcode zugreifen.

Produkte und Kategorien können mit mehreren Bildern ausgezeichnet werden. Im Gegensatz zu der Textbearbeitung können die Bilder allerdings nicht mittels *Drag and Drop* hochgeladen werden. Ergänzende Medienformate wie etwa Videos oder Audio Dateien werden nicht unterstützt. Des Weiteren können die Bilder nicht im Browser editiert werden. Änderungen bezüglich Auflösung oder Größe müssen zwingend außerhalb des Shop-Systems vorgenommen werden, z.B. mittels "Microsoft Visio".

Multi-Channel-Publishing ist mittels sogenannter *Themes* (Deutsch: Designvorlagen) möglich. Standardmäßig werden die Ausgabekanäle *Desktop* und *Mobile* unterstützt, siehe Abbildungen 1 und 2. Die Vorlagen sind auf die Bearbeitung von Textfarben, Schriftfamilien und CSS-Boxen limitiert. Das *Theme* für mobile Endgeräte wird automatisch dem Desktop Erscheinungsbild angepasst und einer geringeren Ausgebequalität optimiert. Bilder werden für *Mobile* standardmäßig nicht angezeigt.

2.2.2 Bestell Administration

Die Verfügbarkeit von Produkten kann mittels der integrierten Lagerverwaltung administriert werden. Hierbei lässt sich beispielsweise auch eine automatisierte Mitteilung einrichten, wenn ein Mindestbestand unterschritten werden sollte.

Dem Kunden steht neben dem konventionellen Warenkorb auch die Möglichkeit zur Pflege einer Wunschliste bereit. Darin kann der Kunde Produkte platzieren an denen sie/er interessiert ist und kann diese Liste auch an Dritte (z.B. Freunde, Familie) weiterleiten. Sollte der Shop global betrieben werden, so kann der Dienstleister auch länderspezifische Steuerklassen hinterlegen. Dem Kunden werden nur die Verfügbarkeitsanzeigen "Im Lager" und "Nicht im Lager" angeboten. Eine Notiz, wie viele Artikel noch verfügbar sind um z.B. auch die Verkaufschance zu erhöhen, wird dem Kunden nicht angezeigt. Die Lieferkosten (Porto) können automatisch vom System errechnet werden analog dem Wegfallen des Porto ab einer gewissen Bestellmenge. Standardmäßig werden die Zahlmethoden "Per Nachnahme", "Vorkasse" und "Kreditkarte" angeboten. Zahlmethoden mittels Drittanbieter "PayPal Express" und "Sofortüberweisung" müssen aktiviert werden.

2.2.3 Kampagnen Administration

Dem Marketingverantwortlichen stehen vielfältige Möglichkeiten zur Verfügung, um die Verkaufschancen für die angebotenen Produkte zu erhöhen. Es können E-Mail Kampagnen generiert werden sowie ein Bonusprogramm und Gutscheinaktionen gestartet werden. Die Effizienz verwirklichter (Absatz)-Kampagnen kann mittels dem vorinstallierten "Google Analytics" [15] *Plug-In* bewertet werden. Dieses analysiert den *Web-Shop* hinsichtlich Zugriffszahlen und misst somit die Relevanz der Kampagnen.

2.3 FRONTEND FUNKTIONEN

2.3.1 Sucherfahrung

Das Suchcenter liefert bei Eingabe von Schlüsselwörtern automatisierte Vorschläge zurück. Des Weiteren kann der Kunde die Resultate hinsichtlich Produktkategorie und Preisspanne eingrenzen. Fehler in der Eingabe werden nicht toleriert. Bei der Pflege neuer Produkte wird automatisch ein für die Suchmaschine "Google"[16] optimierter *Uniform Resource Locators* (URL) generiert.

2.3.2 Soziale Aktivitäten

Schnittstellen zu existierenden Social-Media Plattformen (z.B. Facebook, Twitter) sind nur mittels zusätzlicher *Plug-Ins* möglich. Jedoch bietet das System selbst eine breite Palette an gängigen sozial-medialen Funktionen. Kunden können die Produkte mittels Kommentaren und einem *Rating* bewerten. Kommunikation zwischen den Kunden sowie zwischen Kunden und dem Betreiber, wird

	Kriterium	SmartStore.NET	Magento	xt:Commerce
Editierbarkeit				
Einfachheit	Browserdarstellung	✓	✓	✓
	Templates	✓	✓	✓
	Inhalt/Layout getrennt	✓	×	✓
Multi-Channel Publishing	Medienformate	✓	✓	✓
	Multi-Channel	✓	✓	×
	Multi-Shop	×	✓	×
Bestell Administration				
Bestellprozess	Zahlverfahren	✓	✓	✓
	Lagerverwaltung	✓	×	✓
	Retourmanagement	×	×	×
	Mahnwesen	×	×	×
Kampagnen Administration				
Personalisierte Promotionen	Personalisierter Inhalt	✓	×	✓
	Staffelpreise	✓	✓	✓
	Verwandte Produkte	✓	✓	✓
Warenkorb	Warenkorb	✓	✓	✓
	Wunschliste	✓	✓	×

(✓ = unterstützt, oder mittels kostenfreier Plug-Ins unterstützt; ×=Nicht unterstützt).

Tabelle 2: "SmartStore.NET" Backend Funktionen; Vergleich von Open-Source Anwendungen

	Kriterium	SmartStore.NET	Magento	xt:Commerce
Sucherfahrung				
Fortgeschrittene Suche	Automatische Vervollständigung	✓	✓	×
	Fehlertoleranz	×	×	×
	Weiteres Filtern	✓	✓	✓
Suchoptimierung	Suchfreundliche Links	✓	✓	✓
Soziale Aktivitäten				
Blog/Forum	Blog/Forum	✓	×	×
Kunden-Interaktion	Kundenkommentare	✓	×	✓
	Kundenbewertung	✓	×	✓
	Produkt Verschlagwortung	✓	✓	✓
	Social-Media Schnittstellen	✓	✓	✓

(✓ = unterstützt, oder mittels kostenfreier Plug-Ins unterstützt; ×=Nicht unterstützt).

Tabelle 3: "SmartStore.NET" Frontend Funktionen; Vergleich von Open-Source Anwendungen

mittels eines internen Blogs möglich. Ferner kann der Kunde eigene Attribute, sogenannte Tags (*tagging*, Deutsch: das Verschlagworten, Beifügen von Schlüsselworten) den Produkten beifügen.

3. VERGLEICH

Studien, welche sich mit der Untersuchung der *E-Commerce* Landschaft im deutschsprachigen Raum auseinandersetzen, liefern leider inhomogene Ergebnisse bezüglich Einsatzzahlen von Shop-Systemen (z.B. "t3n.de"[17] oder "statista.com[18]). Die Studien verweisen jedoch alle auf zwei am häufigsten eingesetzte *Open-Source* Lösungen, nämlich "Magento" und "xt:Commerce". Analog des bereits untersuchten Systems, wurde der Kriterienkatalog auf die beiden alternativen *Open-Source* Systeme angewandt. Diese wurden im Folgenden benutzt, um eine neutrale Bewertung von "SmartStore.NET" zu ermöglichen.

3.1 SYSTEM UND PERFORMANZ VERGLEICH

Keines der drei untersuchten Systeme kann mit anderen Management Systemen (ERP, CRM, PIM, etc.) kommunizieren, bieten jedoch den manuellen Austausch für verschiedene Dateiformate sowie die Interaktion mit Preisportalen (siehe Tabelle 1). "Magento" liefert aufgrund der größten *Community* die besten Eigenschaften hinsichtlich *Support* und Erweiterungsmöglichkeiten.

3.2 BACKEND FUNKTIONEN VERGLEICH

Alle Systeme stellen *Templates* bereit (siehe Tabelle 2). Diese ermöglichen das Bearbeiten von Inhalten und dessen Design auch ohne Kenntnisse der dahinterliegenden Technologien (HTML, CSS). Die Editierung erwies sich mit "Magento" am aufwändigsten, da die Trennung von Inhalt und Layout nicht konsequent ist. Jedoch liefert "Magento" das umfangreichste Angebot für *Multi-Channel-Publishing*. Retour Management und ein Mahnwesen sind nur in *E-Commerce* Systemen möglich, welche als Kaufprodukt angeboten werden (z.B. "Intershop"), oder müssen in den *Open-Source* Systemen kostenpflichtig erweitert werden.

3.3 FRONTEND FUNKTIONEN VERGLEICH

Die Suchcenter der analysierten Systeme liefern verschiedene Möglichkeiten, um eine intuitive Sucherfahrung zu gewährleisten. Fehler bei der Suchworteingabe werden allerdings von keinem System toleriert. Im Vergleich zu den anderen *Open-Source* Lösungen bietet "SmartStore.NET" die meisten *Social-Media* Funktionen an.

4. FAZIT

Dieser Artikel untersuchte das *Open-Source E-Commerce* System "SmartStore.NET". Der in der vorliegenden Studie enthaltene Kriterienkatalog erlaubt eine aussagekräftige Untersuchung des Systems hinsichtlich der *State-of-the-Art* Funktionen. Mittels der durchgeführten neutralen Bewertung von "SmartStore.NET" und dessen relavantester Wettbewerbssysteme ist abschließend ein neutrales Fazit möglich.

"SmartStore.NET" erwies sich als ein umfangreiches *E-Commerce* System, in welchem sich auch unerfahrene Benutzer rasch zurechtfinden. Es werden verschiedene *Templates* mitgeliefert, welche einem geringen Designanspruch gerecht werden. Daher ist auch *Multi-Channel-Publishing* nur bedingt möglich und muss gegebenenfalls außerhalb des Systems mittels eigener HTML und CSS Bausteine realisiert werden. Voller Funktionsumfang hinsichtlich *Social-Media* und Kampagnen ist gesichert.

Die Autoren werden in zukünftigen Projekten u. a. an einem Empfehlungssystem arbeiten, welches innerhalb des untersuchten Systems implementiert werden kann.

HEIKO ANGERMANN ist Dozent an der Hochschule der Medien und University of the West of Scotland in den Bereichen "Metadaten Management" und "Content Management". Er ist Betreiber des Beratungshauses "HA-ECM" (Internet: www.ha-ecm.com; E-Mail: angermann@ha-ecm.com), sowie Autor, Sprecher und Trainer im Themenumfeld Microsoft SharePoint, war als SharePoint Administrator für einen Microsoft Gold Partner tätig und leitete internationale SAP-Projekte im Bereich Produkt-Daten-Management.

NAEEM RAMZAN ist Dozent an der University of the West of Scotland und Autor von "Social Media Retrieval". Er ist in europaweite Industrie- und Forschungsprojekte im Themenfeld "Social Media" involviert und im Bereich "Netzwerke" tätig.

[1] http://www.smartstore-net.com/
[2] https://www.codeplex.com/
[3] http://demo.smartstore.com/
[4] http://www.intershop.com/
[5] https://www.informatica.com/
[6] http://www.demandware.com/
[7] https://www.hybris.com/
[8] https://magento.com/
[9] http://www.sitecore.net/
[10] https://www.strato.com/
[11] http://www.xt-commerce.com/
[12] http://www.omeco.de/
[13] https://websale.de/
[14] http:/www.billiger.de/
[15] https://analytics.google.com/
[16] https://www.google.com
[17] http://www.t3n.de
[18] http://www.statista.com
[19] http://www.intershop.com/

EINSTIEG IN AZURE MACHINE LEARNING

Maschinelles Lernen für Softwareentwickler

Spätestens seit dem Sieg von AlphaGo über Lee Sedol [1] ist Machine Learning (kurz ML) auch der breiten Masse ein Begriff. Im Folgenden wird Microsofts ML Dienst Azure Machine Learning vorgestellt. Im Besonderen wird gezeigt, wie dieser Dienst genutzt wird und wie er in eigene Projekte integriert werden kann.

WAS IST MACHINE LEARNING?

Das Ziel von maschinellem Lernen ist es, anhand von historischen oder künstlich erstellten Daten eine Aussage zur Lösung eines Problems zu treffen. Wichtig hierbei ist es, dass die Daten nicht einfach auswendig gelernt oder gespeichert werden, sondern dass die Software Muster erkennt und somit auch in der Lage ist, Probleme für noch unbekannte Eingabeparameter zu lösen.

Mit ML können unterschiedlichste Probleme gelöst werden. Eine Auswahl der Probleme, die heute schon mit ML gelöst werden, sind: Erkennung von Kreditkartenbetrug, Aktienmarktanalyse oder Spracherkennung. Es gibt unterschiedliche Algorithmen, die für den Lernprozess genutzt werden können. Die Algorithmen können nach ihren Ansätzen gruppiert werden. Diese Ansätze sind unter anderem Überwachtes Lernen, Bestärkendes Lernen und Unüberwachtes Lernen.

WAS IST AZURE MACHINE LEARNING?

Azure Machine Learning ist ein verwalteter Cloud Dienst. Azure ML ermöglicht die einfache Nutzung von maschinellem Lernen ohne die Anschaffung teurer Software oder Hardware. Unterschiedliche Lernvorgänge werden dabei in Versuche (engl. Experiment) aufgeteilt. Ein Versuch enthält in Azure ML unterschiedliche Module, die in einem grafischen Editor wie in einem Datenflussdiagramm verbunden werden können.

WIE KANN ICH AZURE MACHINE LEARNING NUTZEN?

Microsoft bietet Azure ML mit zwei Preismodellen an; Free und Standard. Wie bei vielen Diensten in Azure wird auch hier durch die kostenlose Version des Dienstes ein schneller und einfacher Start ermöglicht. Für den produktiven Einsatz ist die kostenlose Version allerdings nur

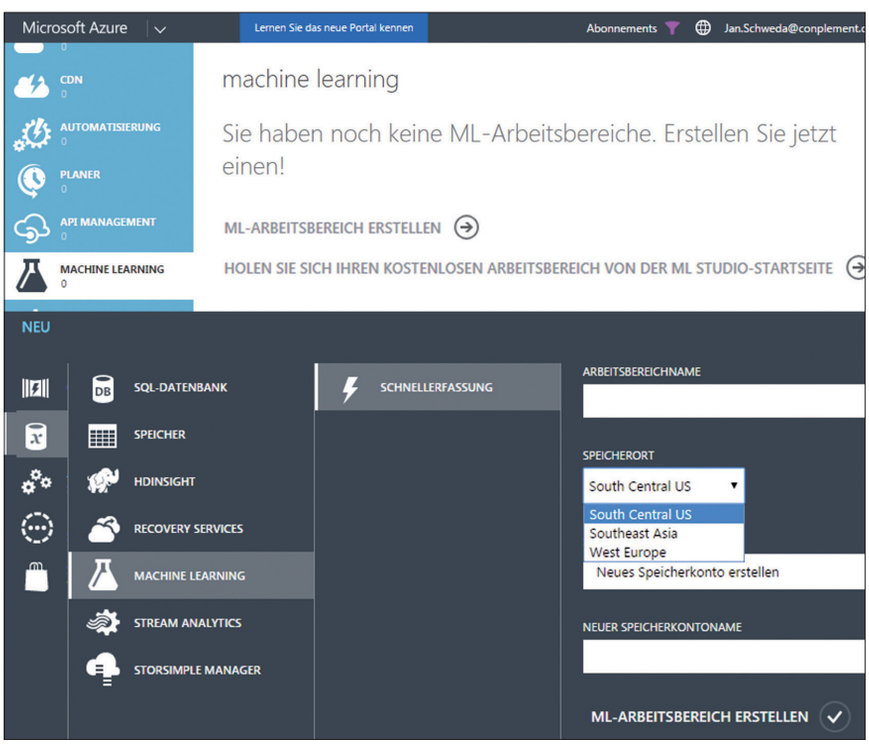


Abbildung 1: Erstellen eines neuen ML Workspaces über das Azure Portal.

Überwachtes Lernen und Unüberwachtes Lernen

Überwachtes Lernen (engl. Supervised Learning) und Unüberwachtes Lernen (engl. Unsupervised Learning) sind die beiden wohl am weitesten verbreiteten Kategorien von maschinellem Lernen.
Algorithmen des Überwachten Lernens leiten aus einer Datenmenge Strategien zur Problemlösung ab. In der Datenmenge sind sowohl Eingabewerte wie auch Ergebniswerte für das Problem enthalten. Wurde die Strategie, eine nichtlineare Funktion, abgeleitet, kann sie auf noch unbekannte Eingaben angewendet werden.
Algorithmen des Unüberwachten Lernens erkennen Muster in einer Datenmenge. Durch die erkannten Muster können die Daten geclustert oder komprimiert

bedingt nutzbar.
Um die kostenlose Version zu nutzen, ist bei Azure ML keine Azure Subscription notwendig. Unter *https://studio.azureml.net/* kann man mit einem Microsoft Konto einen Workspace einrichten. Bei Nutzung der kostenlosen Version gibt es jedoch die folgenden Einschränkungen: Die Anzahl der maximalen Module innerhalb eines Versuches ist auf 100 begrenzt, die Dauer des Versuches darf eine Stunde nicht überschreiten, es kann nur ein Knoten für die Ausführung genutzt werden und der Speicherplatz für Datasets und eigene Module ist auf 10 GB begrenzt.

Entscheidet man sich für die kostenpflichtige Variante, ist der Einstiegspunkt zum Erstellen eines Workspaces derzeit noch das alte Azure Portal (https://manage.windowsazure.com). Abbildung 1 zeigt den Menüpunkt für ML und den Dialog zum Anlegen eines Workspaces. In diesem Angebot ist kein Speicherplatz enthalten. Es wird ein zusätzlicher Azure Speicher-Account benötigt. Der Speicher muss im selben Rechenzentrum angelegt sein wie der ML Workspace. Aktuell wird ML lediglich in den Rechenzentren Westeuropa, Südostasien und Südzentral-Amerika angeboten.

WIE WIRD EIN VERSUCH DURCHGEFÜHRT?

Ein Versuch wird im Azure ML Studio erstellt. Das ML Studio ist eine einfach strukturierte Entwicklungsumgebung. Im Grunde besteht sie aus fünf Bereichen:

- *Ganz Links:* Navigation zum Wechseln zwischen unterschiedlichen Bereichen des ML Studios: Versuche, Webservices, Notebooks, Datasets, Trainierte Modelle und Einstellungen.
- *Links:* Alle Module, die in einem Versuch genutzt werden können. Die Module sind logisch gruppiert.
- *Mittig:* Die Visualisierung des Versuchs in einem Datenflussdiagramm
- *Rechts:* Details und Einstellungen zum aktuell selektierten Modul.
- *Unten:* Aktionen für den Versuch.

Zum Erstellen eines Versuches werden beliebig viele Modulen innerhalb eines Datenflussdiagramms verknüpft.

Im Folgenden wird ein Versuch erstellt, der mit den Daten eines verfügbaren Sample-Datasets ein Modell trainiert. Das trainierte Modell soll die Frage, ob eine Person mehr als oder weniger als eine definierte Summe im Jahr verdient, beantworten. Das Einkommen der Person ist in der Spalte income festgehalten, die Werte sind mehr als 50.000$ (>50k) oder genau 50.000$ oder weniger (<=50k).

Das Dataset besteht aus ca. 32.500 Datensätze. Die Spalten des Datasets sind: *age, fnlwgt (Final sampling weight: Kennzahl wie viele Haushalte durch diesen Wert repräsentiert werden), education-num (Bildungsabschluss als Zahl), education, marital-status, relationship, race, sex, capital-gain, capital-loss, hours-per-week* und *income*.

DATEN LADEN

Um mit dem Versuch zu beginnen, werden zuerst die Daten geladen.

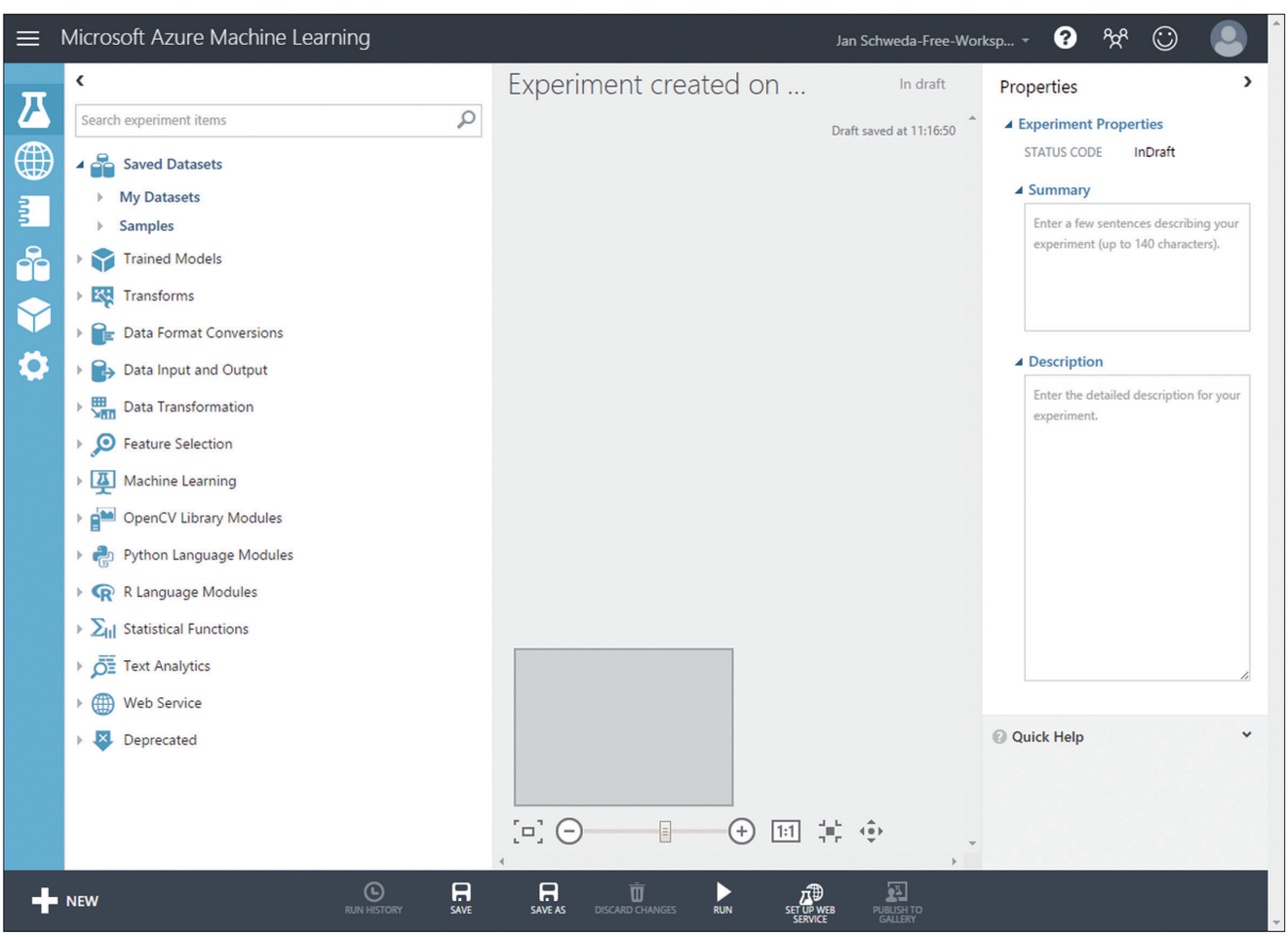


Abbildung 2: Die ML Studio Entwicklungsumgebung.

Daten können entweder aus einem Dataset importiert werden oder zur Laufzeit aus unterschiedlichsten Datenquellen geladen werden. Ein Dataset kann hinzugefügt werden, indem eine Datei hochgeladen und als Dataset gespeichert wird oder als Export innerhalb eines Versuchs. Mögliche Quellen für das Laden zur Laufzeit sind unter anderem Hive Queries, Azure Tables, Azure Blobs, Webservices oder Azure SQL Datenbanken.

Die Daten für unseren Versuch kommen wie bereits erwähnt aus den vorhandenen Beispieldaten, genauer aus dem Adult Census Income Binary Classification Dataset. Dieses Dataset ist unter Saved Datasets -> Samples -> Adult Census... verfügbar.

Nachdem das Dataset per Drag & Drop dem Arbeitsbereich hinzugefügt wurde, kann es als Modul im Datenflussdiagramm genutzt werden. Alle Module in einem Versuch verfügen über mindestens einen Eingang oder Ausgang, viele Module auch über mehrere Eingänge oder Ausgänge. An den Ausgängen können mit der Maus Kontextmenüs geöffnet werden. Das Kontextmenü für Datasets enthält die folgenden Aktionen:

- *Download:* Das Dataset wird heruntergeladen und kann lokal gespeichert oder bearbeitet werden.
- *Visualize:* Ein Fenster mit Informationen zu dem Dataset wird geöffnet
- *Generate Data Access Code:* Erstellt einen Code mit dem sicher aus anderen Entwicklungsumgebungen auf dieses Dataset zugegriffen werden kann.
- *Open in a new Notebook:* Öffnet ein Jupyter Notebook, in dem mit Python 2 oder Python 3 auf den Versuch zugegriffen werden kann.

In der Visualisierung für ein Dataset werden statistische Werte zu dem Dataset angezeigt. So sind unter anderem die Anzahl der Zeilen, die Spalten mit ihren Überschriften, Werten und Verteilungen erkennbar.

DATEN AUFBEREITEN

Nicht immer können Daten in dem Format importiert werden, das für das Training des Modells optimal ist. Um aber ein optimal trainiertes Modell zu erhalten, müssen die Daten so aufbereitet sein, dass keine Zeilen, Spalten oder Felder mehr enthalten sind, die das Ergebnis

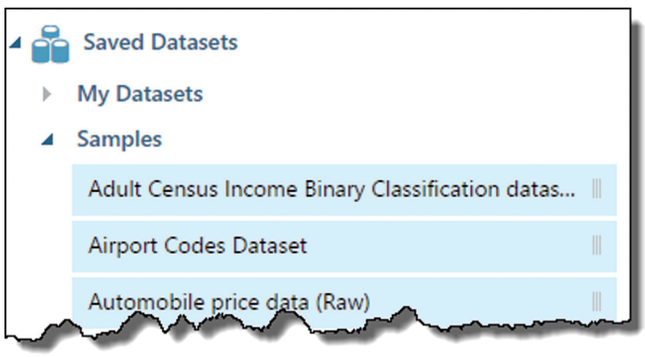


Abbildung 3: Das Dataset auswählen.

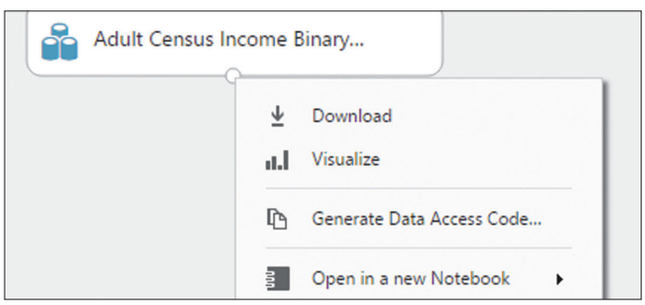


Abbildung 4: Das Kontextmenü für ein Dataset.

verfälschen. An dieser Stelle muss erwähnt werden, dass ein gutes Modell nicht einfach so entsteht. Es steckt viel Erfahrung und Geduld in einem wirklich guten Modell. Eine große Hilfe bei der Auswahl der relevanten Spalten kann das Modul Filter Based Feature Selection sein. Dieses Modul kann die relevantesten Spalten ermitteln.

Die Reihenfolge der nächsten Schritte kann von Versuch zu Versuch variieren.

Für diesen Versuch werden zuerst alle Spalten, die nicht benötigt werden oder redundant sind, entfernt. Das Modul für diesen Zweck heißt Project Columns. Nachdem das Modul dem Datenflussdiagramm hinzugefügt wurde, wird der Ausgang des Datasets mit seinem Eingang verbunden. Abbildung 6 zeigt das vorläufige Ergebnis des Versuchs.

Das Ausrufezeichen auf rotem Grund zeigt an, dass in diesem Modul ein Fehler vorliegt. Solange ein Fehler vorliegt, kann der Versuch nicht durchgeführt werden und auch kein Ergebnis eines Moduls visualisiert werden.

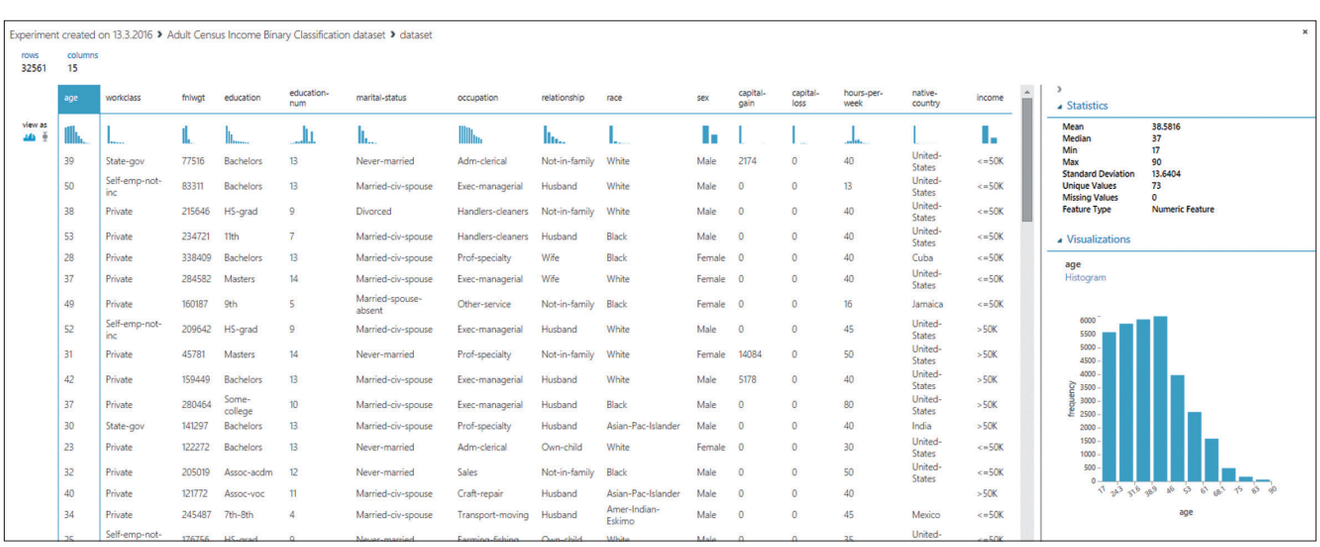


Abbildung 5: Visualisierung für das Dataset.

Der Fehler in dem Project Columns Modul ist, dass die Projektion bis jetzt noch nicht definiert wurde. Die Konfiguration für das Modul ist im rechten Teil des ML-Studios vorzunehmen. Über den Column Selector können die Spalten, die im Ergebnis enthalten sein sollen, definiert werden. Hierzu kann entweder mit allen Spalten begonnen werden und einzelne Spalten werden exkludiert oder es werden keine Spalten genommen und einzelne hinzugefügt. Abbildung 7 zeigt diesen Dialog mit dem Ansatz mit allen Spalten zu beginnen und gezielt Spalten zu entfernen.

Als nächstes wird auf verbliebene Felder reagiert, die nicht befüllt sind. Hier gibt es unterschiedliche Möglichkeiten, wie mit solchen Feldern verfahren werden soll. Die zu wählende Möglichkeit wird jedoch mehr durch den Kontext der Daten als durch die technischen Möglichkeiten bestimmt. Wenn genug Daten vorliegen, können Zeilen oder Spalten mit fehlenden Werten komplett aus dem Dataset ausgeschlossen werden. Fehlende Felder können aber auch durch statistische Funktionen befüllt werden.

DAS MODELL TRAINIEREN

Die aufbereiteten Daten werden auf zwei Teilmengen aufgeteilt. Mit der ersten und größeren Teilmenge wird das Modell trainiert, mit der zweiten kleineren wird das trainierte Modell validiert. Es bietet sich an, 70 – 80 Prozent der Daten zum Trainieren zu verwenden und den Rest der Daten für die Validierung aufzusparen. Beim Aufteilen der Daten kann konfiguriert werden, ob die Daten bei jeder Versuchsdurchführung gleich oder zufällig aufgeteilt werden sollen. Wenn man unterschiedliche Algorithmen oder Veränderungen an

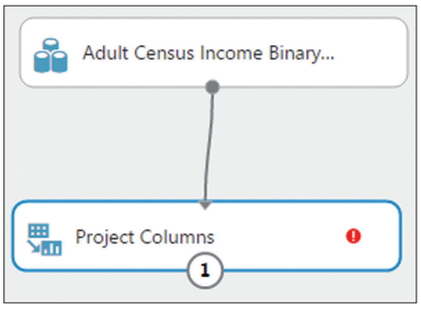


Abbildung 6: Die ersten Schritte mit dem Modul.

den Daten vergleichen möchte, sollte man gleich und nicht zufällig aufteilen lassen, damit die Ergebnisse nicht durch eventuelle Unregelmäßigkeiten in den Daten verfälscht werden.

Um das Modell zu trainieren, wird ein Algorithmus ausgewählt. Die Algorithmen befinden sich unter Machine Learning → Initialize Model. Der selektierte Algorithmus und die Daten zum Trainieren dieses Algorithmus werden mit einem Train Model verbunden. Im Train Model Modul wird definiert, welche Spalte oder Spalten der Daten das Ergebnis darstellt.

DAS MODELL BEWERTEN

Um ein Modell zu bewerten, sind zwei Module in das Datenflussdiagramm hinzuzufügen, Score Model und Rate Model. Score Model erwartet zwei Eingänge. Das trainierte Modell und die Daten zum Verifizieren des Modells. Das Ergebnis von Score Model wird an das Modul Evaluate Model übergeben.

- *True Positive – Richtig positiv:* Anteil der richtig als positiv eingestuften Ergebnisse. Je größer, desto besser.
- *False Positive – Falsch positiv:* Anteil der fälschlich als positiv eingestuften Ergebnisse. Je kleiner, desto besser.
- *False Negative – Falsch negativ:* Anteil der fälschlich als negativ eingestuften Ergebnisse. Je kleiner, desto besser.
- *True Negative – Richtig negativ:* Anteil der richtig als negativ eingestuften Ergebnisse. Je größer, desto besser.
- *Accuracy – Exaktheit:* Anteil der insgesamt richtigen Ergebnisse. Je größer, desto besser.
- Recall – Rückruf: Definiert den Anteil aller richtigen Ergebnisse, die vom Modell zurückgegeben werden. Je kleiner, desto besser.
- Precision – Genauigkeit: Der Anteil von wahren Ergebnissen an den positiven Ergebnissen. Je größer, desto besser.
- F1 Score - F-Bewertung: Eine Maßgröße für die Exaktheit, die Genauigkeit und Rückruf ausgleicht. Je näher an der 1, desto besser.
- AUC – Fläche unter der Kurve: Die Fläche unter der Kurve bietet eine einzelne Zahl um Modelle zu vergleichen. Je größer, desto besser.

ZWEI MODELLE VERGLEICHEN

Wenn die Ergebnisse von zwei unterschiedlichen Algorithmen mit denselben Daten verglichen werden sollen, kann dies in einem Versuch kombiniert werden. Hierzu werden dem Datenflussdiagramm parallel zu den bereits bestehenden Modulen ein weiteres Train Model, sowie ein weiteres Score Model Modul hinzugefügt. Die neu hinzugefügten Module werden mit denselben Daten verbunden wie bereits die Module aus den vorherigen Schritten verbunden wurden. Allerdings mit der Ausnahme, dass Train Model mit einem anderen Algorithmus initialisiert wird. Das Ergebnis von Score Model wird dem zweiten Eingang von Evaluate Model zugewiesen.

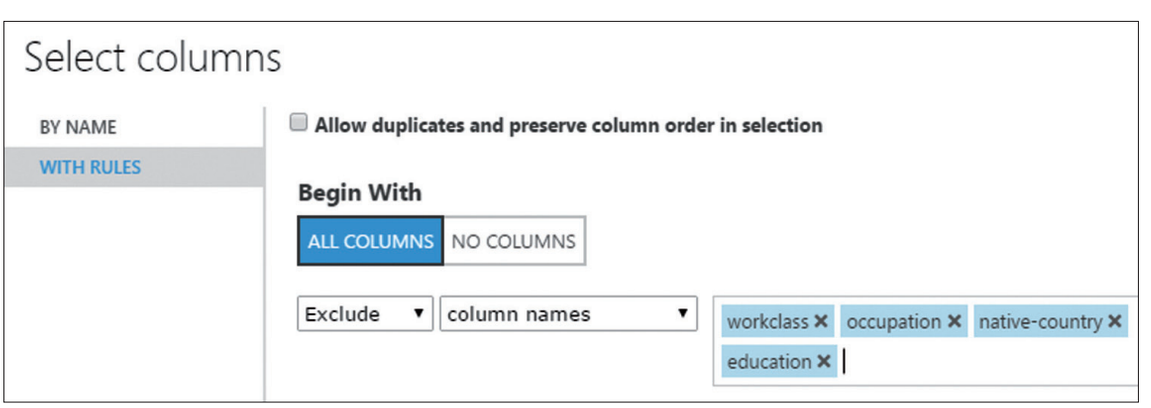


Abbildung 7: Der Select Columns Dialog.

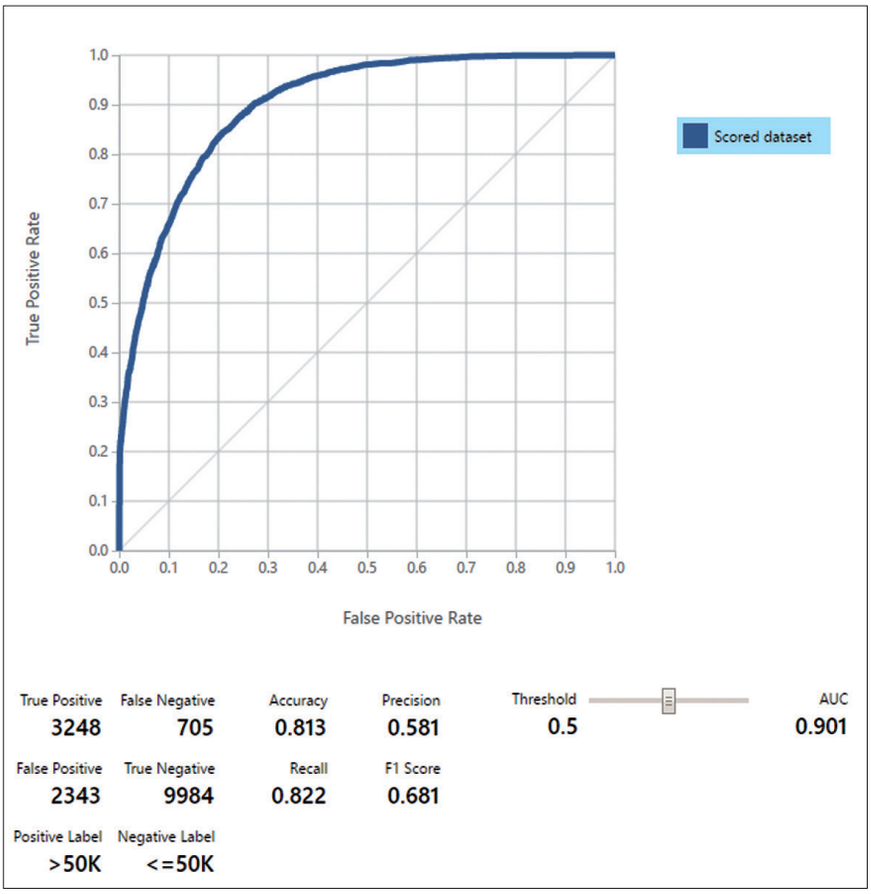


Abbildung 8: Die Auswertung des Modells.

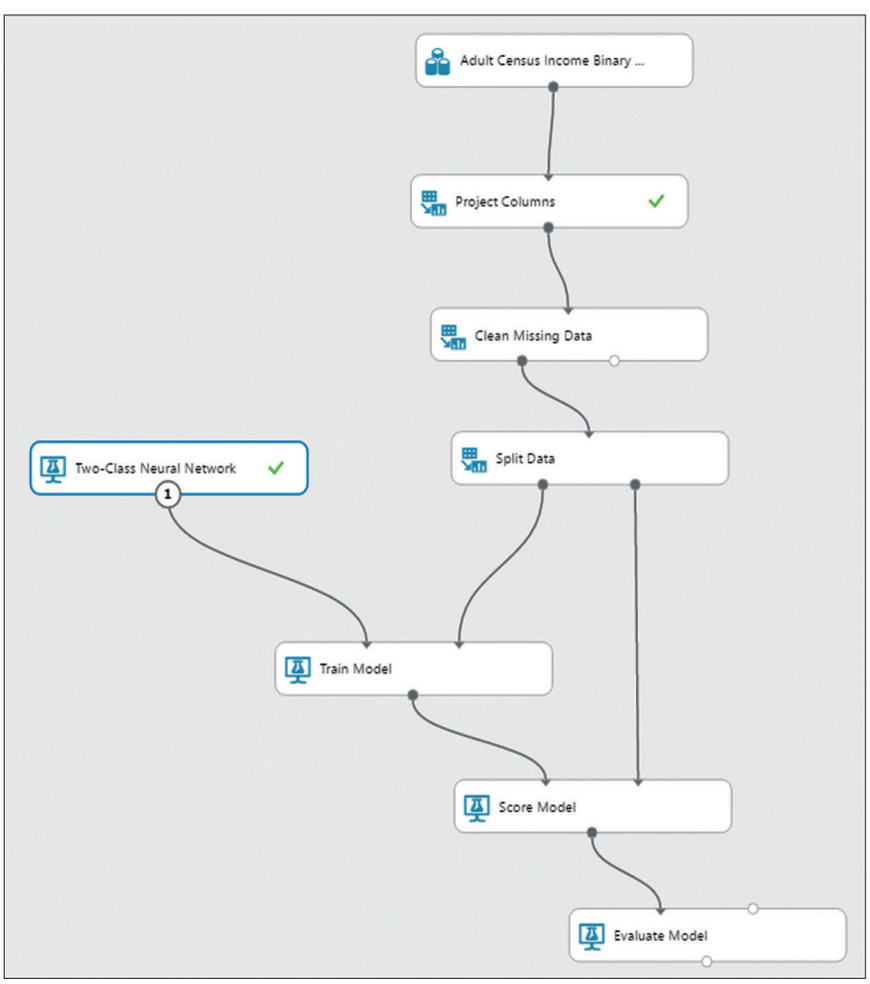


Abbildung 9: Der fertige Versuch.

WAS PASSIERT MIT EINEM TRAINIERTEN MODELL?

Das trainierte Modell kann auf zwei Arten weiter genutzt werden. Zum einen kann das Modell gespeichert werden und in anderen Versuchen genutzt werden. Zum anderen kann aus einem trainierten Modell ein Webservice erstellt werden, über den das Modell einfach angesprochen werden kann. Der Webservice wird über den Befehl Setup Web Service im unteren Menü erstellt. Bei mehreren trainierten Modellen in einem Versuch muss das Modell, das veröffentlicht werden soll, selektiert sein. Setup Web Service erstellt aus dem Versuch zwei Komponenten, ein Training Experiment, welches sich nicht von dem ursprünglichen Versuch unterscheidet und ein Predictive Experiment. Das Predictive Experiment enthält zusätzlich einen Webservice-Input und einen Webservice-Output. Das Predictive Experiment muss einmal per Run ausgeführt werden, bevor durch den Befehl Deploy web service ein Webservice erstellt werden kann.

Beim Erstellen eines Webservice wird automatisch das Dashboard für diesen geöffnet. Das Dashboard enthält alle relevanten Informationen zu dem Webservice. Neben dem API-Key, den Clients benötigen wir, um diesen Dienst zu konsumieren, auch die Beschreibung für die beiden Endpunkte, die der Dienst veröffentlicht. Über den Request/Response Endpunkt können Anfragen direkt an das Modell geschickt werden. Der Batch Execution Endpunkt hingegen bearbeitet Daten, die in Azure Storage hinterlegt wurden. Der Request/Response Endpunkt ist sofort über eine einfache Oberfläche im Browser aufrufbar. Zusätzlich werden Excel Dateien zum Download angeboten, die ebenfalls diesen Endpunkt konsumieren.

Wenn der Dienst im Rahmen einer größeren Anwendung verwendet werden soll, kann aus dem Dashboard auch Sample Code für C#, Python und R heruntergeladen werden. Jetzt steht der Verwendung unseres Modells für maschinelles Lernen durch Anwendungen per Webservice Aufruf nichts mehr im Wege.

FAZIT

Azure Machine Learning macht den Start mit ML einfach. Versuche sind

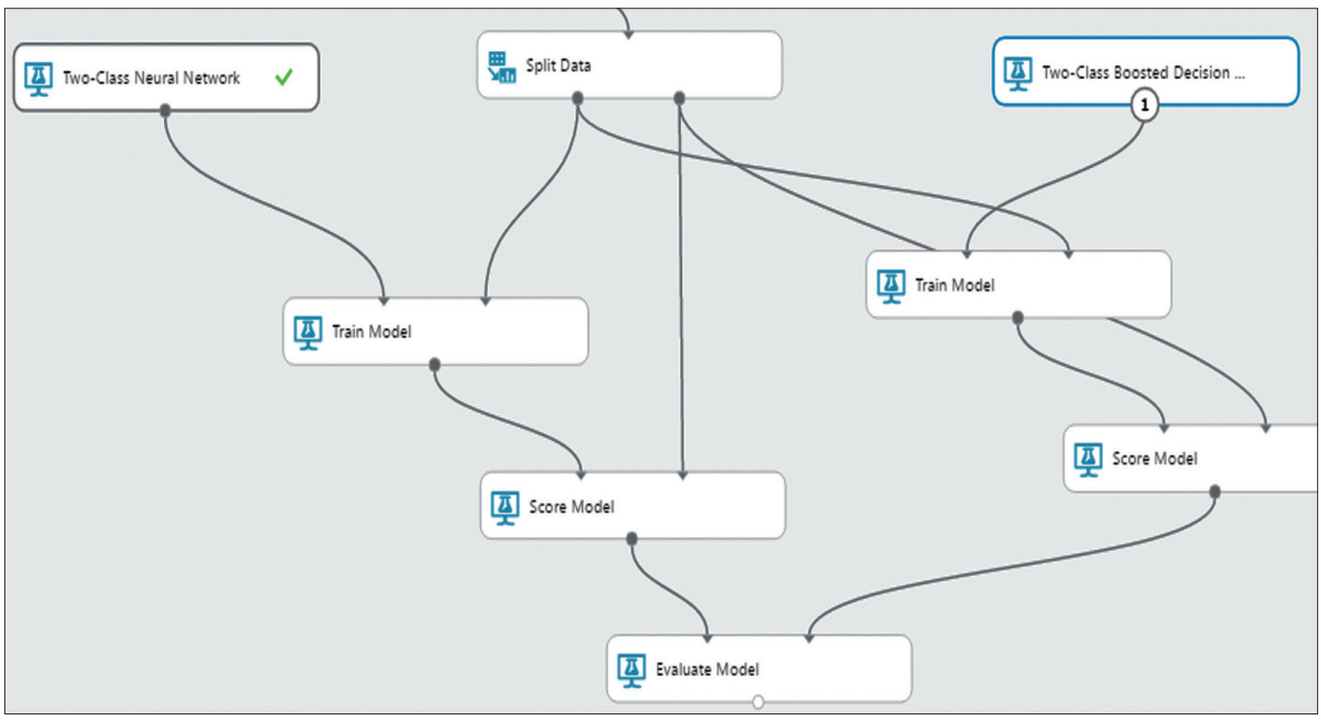


Abbildung 10: Training von zwei Modellen.

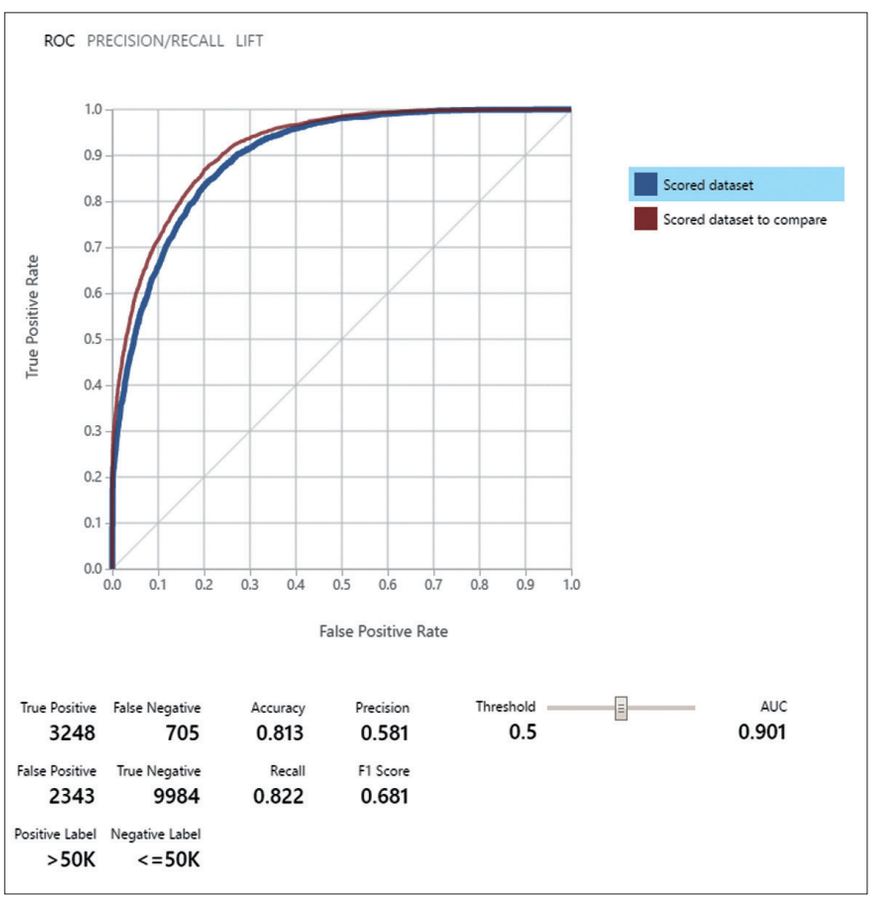


Abbildung 11: Die Visualisierung von Rate Model mit zwei trainierten Modellen.

auch ohne viel Statistikwissen durchführbar. Durch die vielen vorgefertigten Module ist sehr viel Analyse ohne eine einzige Zeile Code möglich. Zusätzlich können für wirklich spezielle Anforderungen eigene Module in SQL, R oder Python erstellt werden.

LINKS UND QUELLEN

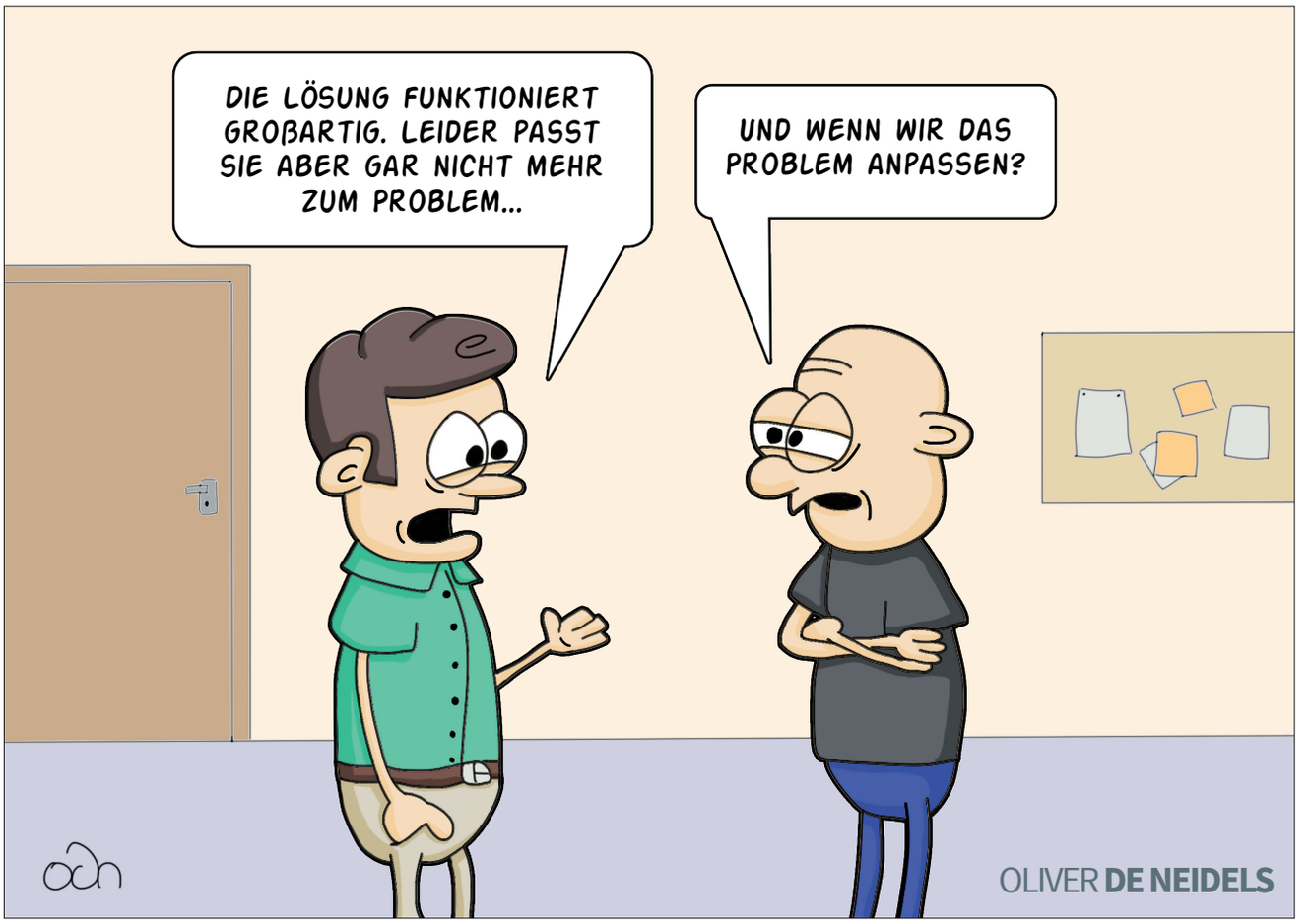
[1] Mensch gegen Maschine: Google-KI AlphaGo schlägt Lee Sedol
http://www.heise.de/newsticker/meldung/Mensch-gegen-Maschine-Google-KI-AlphaGo-schlaegt-Lee-Sedol-3-0-3133401.html

[2] Maschinelles Lernen
https://de.wikipedia.org/wiki/Maschinelles_Lernen

[3] Preise Azure ML
https://azure.microsoft.com/de-de/pricing/details/machine-learning/

[4] Machine Learning Studio
https://studio.azureml.net/

JAN SCHWEDA ist Technology Specialist bei der conplement AG in Nürnberg. Sein Fokus liegt vor allem auf verteilten Systemen, Cloud-Architekturen und serviceorientierter Architektur. Jan entwickelt seit mehreren Jahren Consumer und Enterprise-Anwendungen im .NET-Umfeld und hat hierbei Erfahrungen in den unterschiedlichen Bereichen des .NET Technologie-Stacks gesammelt. Sein Wissen gibt er regelmäßig und gerne als (Microsoft Certified) Trainer, Fachautor und Sprecher weiter.



SCHULUNGS-TERMINE

SOFTSKILLS FÜR IT-MANAGER

IT-Teams führen und motivieren, 2 Tage
Berlin ab 21. April

Management von erfolgreichen IT-Projekten, 2 Tage
Düsseldorf ab 09. Mai
Karlsruhe ab 24. Mai

PPEDV.DE/COACH

.NET UND VISUAL STUDIO 2015

.NET - Moderne Architekturen und Designprinzipien, 3 Tage
Stuttgart ab 02. Mai
Leipzig ab 11. Mai

C# - Grundlagen der Programmierung, 4 Tage
Wien ab 19. April
Burghausen ab 19. April
Frankfurt ab 17. Mai

C++ - Einstieg, 4 Tage
Wien ab 19. April

Visual Basic .NET für Einsteiger, 3 Tage
Frankfurt ab 02. Mai
Düsseldorf ab 13. Juni

Windows Presentation Foundation, 4 Tage
Berlin ab 07. Juni
Karlsruhe ab 14. Juni
Berlin ab 05. Juli

Windows Communication Foundation, 3 Tage
München ab 27. April
Karlsruhe ab 18. Mai

Power-Woche: ADO.NET, LINQ, Entity Framework, 5 Tage Berlin ab 09. Mai

Softwaretests mit Visual Studio, 2 Tage
Berlin ab 10. Mai
Burghausen ab 07. Juni

PPEDV.DE

SHAREPOINT SERVER 2016

SharePoint – Power-Woche für Entwickler, 5 Tage
Wien ab 20. Juni
München ab 04. Juli

SharePoint - Lösungen entwickeln und anpassen, 3 Tage
Wien ab 27. April

PPEDV.DE/SHAREPOINT

WEB-DEVELOPMENT

JavaScript, HTML und CSS, 3 Tage
Berlin ab 18. April
München ab 18. April
Leipzig ab 30. Mai

AngularJS, 2 Tage
München ab 17. Mai

jQuery, 2 Tage
Nürnberg ab 21. April

ASP.NET - WebForms entwickeln, 4 Tage
Berlin ab 26. April

ASP.NET MVC, 3 Tage
München ab 18. Mai

ASP.NET Web API, 2 Tage
Nürnberg ab 28. April

Powerwoche ASP.NET MVC und WebAPI, 5 Tage
Frankfurt ab 13. Juni

Xamarin: Native Cross-Plattform-Apps, 4 Tage
München ab 07. Juni und
Köln ab 21. Juni

Responsive Design mit CSS3 und Bootstrap, 2 Tage
Stuttgart ab 25. April und
ab 12. Mai

Azure - Cloud Lösungen entwickeln, 4 Tage
Karlsruhe ab 26. April

PPEDV.DE/WEB

INFO & ANMELDUNG
ppedv AG · +49-8677-988 90 · ppedv.de · schulung@ppedv.de · twitter.com/ppedv · facebook.com/ppedvAG

Präferenzanalyse mit "Microsoft Social Engagement"

Big Data und Web 3.0 erreicht Microsoft Dynamics CRM

"Microsoft Dynamics CRM" wurde 2015 um das *Social Listening* Modul "Microsoft Social Engagement" erweitert. Im Zeitalter des Internet-Paradigmas "Web 3.0" (auch: *Semantic* Web) werden solche Module genutzt, um die Popularität und Präferenzen gegenüber eigenen Marken und Produkten, aber auch gegenüber Marktwettbewerbern zu analysieren. *Social-Media* Kanäle wie Facebook, Twitter und Blogs werden hierbei zweifach genutzt: Zum einen zur Durchführung der oben erwähnten Analysen und zum anderen zur Veröffentlichung personalisierter Werbekampagnen und Sichtung deren Effizienz. Der vorliegende Artikel präsentiert eine Übersicht, wie "Microsoft Social Engagement" innerhalb von "Microsoft Dynamics CRM" funktioniert und inwiefern es mit etablierten Wettbewerbern vergleichbar ist. Der Artikel basiert zum Zwecke einer technischen und benutzerrelevanten Veranschaulichung auf einem beispielhaften Anwendungsfall und gibt abschließend ein Fazit hinsichtlich Pros und Kontras des Systems.

1. HINTERGRUND

Das "Semantic Web"[1] (Deutsch: semantisches Internet) ermöglicht die Darstellung von Beziehungen zwischen Objekten (z.B. Produktkategorien, Synonyme, Schlüsselworte) und ist der aktuellste Quantensprung im Internet. Durch die semantische Strukturierung von Relationen zwischen Objekten und Objekten, aber auch zwischen Objekten und Personen, wird die Generierung von neuem Wissen möglich. Aufgrund der täglich steigenden Anzahl an verfügbaren Datenmengen (auch: *Big Data*) werden immer detailliertere Erkenntnisse möglich, welche von Unternehmen zur Steigerung von Absatzchancen sowie zur Verbesserung der Außenwirkung eigener Marken und Produkte genutzt wird. Einerseits werden Standards wie das "Resource Description Framework" [2] (RDF) durch das "World Wide Web Consortium" (W3C) etabliert, welches semantische Beziehungen mittels sogenannter Drei-Tupel (Subjekt, Prädikat, Objekt) modelliert (z.B. "[Kunde Mustermann] [kauft] [Bohrmaschine]", oder [Bohrmaschine] [ist ein] [Elektrogerät]). Andererseits werden sogenannte *Social Listening* (Deutsch: sozialen Medien zuhören) Systeme von "NetBreeze"[3] oder "Brandwatch"[4] präsentiert. Diese dienen dem Ziel, unstrukturierte Daten aus *Social-Media* Kanälen eben derartig semantisch zu strukturieren und um hieraus gewinnbringende Maßnahmen (z.B. Werbekampagnen, technische Produktverbesserungen) abzuleiten. Die Kommentare in sozialen Netzwerken liefern Informationen über die Stimmung von einzelnen Kunden und Kundengruppen gegenüber den Produkten und Marken mittels sogenannter Sentiment Analysen (z.B. Kunde Mustermann schreibt: [Bohrmaschine] [bohrt] [super]). Diese Sentiment Analysen werden, durch Metadaten gestützte Suchanfragen, an das *Social Listening* System gestellt. Üblicherweise enthalten diese Suchanfragen Metadaten wie Produktbezeichnungen (z.B. [Bohrmaschine]), Hersteller (z.B. [Festool]), zu durchsuchende Quellen und Sprachen (z.B. [Facebook], [Deutsch], [Englisch]), aber auch weitere Filter wie Zeitspannen (z.B. [Kalenderwoche 52]) oder auszuschließende Texte (z.B. [Akkuschrauber]). Das Resultat der Suchanfrage sagt letztendlich aus, inwieweit die Metadaten und damit unser Produkt in der Web-Community repräsentiert werden.

Da es in unserer heutigen Internetlandschaft beinahe jedem möglich ist in Form von *Postings* und *Tweets* seine Meinung ungefiltert zu äußern, müssen derartige Systeme verschiedenste Funktionen und Kriterien berücksichtigen:

- ***Daten Konnektoren*** für die Anfrage an möglichst viele *Social-Media* Kanäle, z.B. Facebook, Twitter, Youtube, Google+, aber auch weniger frequentierte Blogs.
- ***Sprach Selektierung*** um den Kommentar einer natürlichen Sprache zuzuordnen, z.B. Deutsch, auch wenn die Kommentare oftmals englische Wörter enthalten.
- ***Inhalts Analyse*** leitet die Stimmung (Sentiment) aus unstrukturierten Daten ab. Gegenwärtige Systeme unterschieden hierbei in: positiv, neutral, negativ.
- ***Nutzung Analyse*** stellt die Anzahl an Weiterleitungen (Shares, Retweets) fest.
- ***Beeinflusser Selektierung*** (Englisch: *Influencers*) um besonders einflussreiche Blogger zu identifizieren.

Diese tragen oftmals wesentlich zu einer positiven oder negativen Außenwirkung von Produkten, Marken und Unternehmen bei.

- ***Inhalts Benachrichtigung*** weist den Betreiber auf wichtige Vorgänge hin, z.B. bei einem Shitstorm, oder wenn Wettbewerber plötzlich stark diskutiert werden.

In den letzten Jahren wurden Systeme wie "Brandwatch Analytics"[5], "BuzzSumo Pro"[6] oder "Social Searcher"[7] vorgestellt, welche natürlich unterschiedliche Stärken und Schwächen in den genannten Funktionen aufweisen. Anfang 2015 entschied sich Microsoft, das damals durch "NetBreeze" vertriebene *Social Listening* Modul in die "Microsoft Dynamics CRM" Plattform zu integrieren. Mittlerweile ist das Modul unter dem Namen "Microsoft Social Engagement" bekannt. Hierdurch zollt Microsoft dem *Hot-Topic Social-Media* Tribut und erweitert seine CRM-Lösung wesentlich. Im Gegensatz zu den oben erwähnten Konkurrenzprodukten ist "Microsoft Dynamics CRM" nämlich eines von vielen Unternehmen bereits eingesetztes "Customer Relationship Management" (CRM) System. Dadurch verspricht das zusätzliche Modul eine rasche Einarbeitungsphase auf Seiten des Benutzers. Auf Seiten des Unternehmens verspricht es eine lückenlose Einbettung in die bestehende IT-Landschaft und letztendlich auch eine perfekte Kommunikation zu dem bereits vorhandenen CRM System.

Um zu sehen, wie "Microsoft Social Engagement" funktioniert, welche Funktionen erfüllt werden und ob das Modul mit seinen *Standalone* Wettbewerbern auf Augenhöhe steht, evaluieren wir dieses basierend auf einem konkreten Anwendungsfall. Aus der Perspektive eines Marketingverantwortlichen des deutschen Maschinenherstellers Festool[8], analysieren wir die Komponenten des Moduls. Hierbei betrachten wir das Starten von *Social Listening* Aktivitäten, das Modifizieren von Suchanfragen, Bewerten die Qualität der Resultate und gehen auf die Bedienbarkeit ein. Die Fähigkeit des Moduls wird abschließend mit dessen Wettbewerbern verglichen.

2. STARTEN VON "MICROSOFT SOCIAL ENGAGEMENT"

"Microsoft Social Engagement" ist in der Lage 19 Sprachen zu identifizieren und die Texte in Echtzeit zu analysieren. Innerhalb von "Microsoft Dynamics CRM" wird dies von Benutzern genutzt, um die Außenwirkung von Produkten oder Marken festzustellen, hieraus Maßnahmen zu erschließen, aber auch zum Vergleich mit Wettbewerbern. Die Cloud-basierende CRM Erweiterung bietet zusammengefasst die folgenden Funktionen:

- Übersicht für das Monitoring von Web-Gemeinschaften,
- Individuelle Benachrichtigungen in Echtzeit,
- Bereitstellung durch das "Microsoft Office 365" Paket,
- Individuelle Suchanfragen durch Metadaten,
- Interaktive und visuelle Interpretation von Resultaten,
- Sentiment-Analyse mit drei Stimmungskategorien (positiv, neutral, negativ),
- Direkte Schnittstelle zu allen gängigen Social-Media Kanälen,

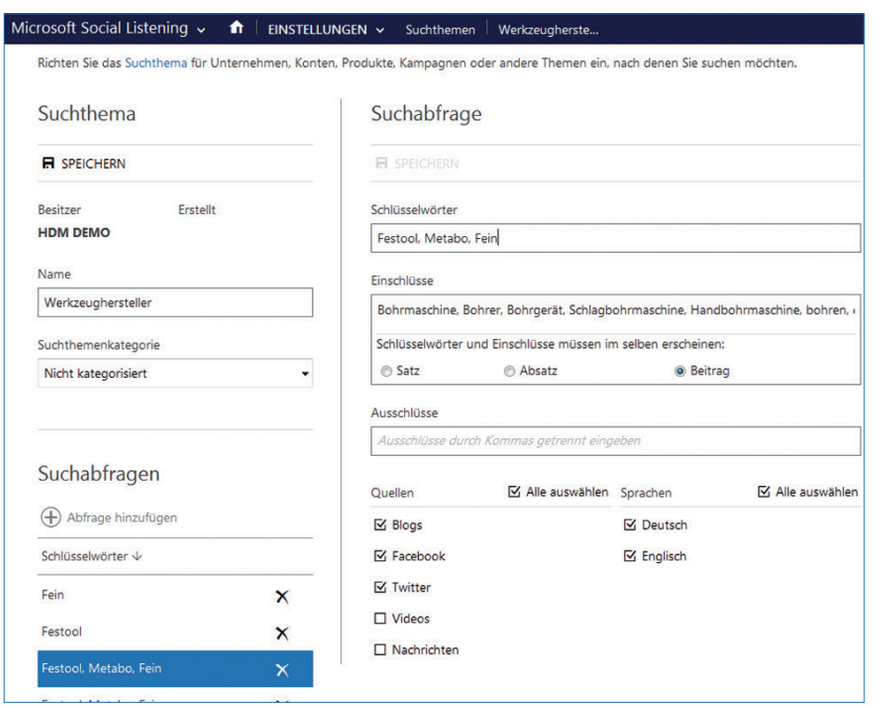


Abbildung 1: Starten von "Microsoft Social Engagement" – Definieren von Suchanfragen.

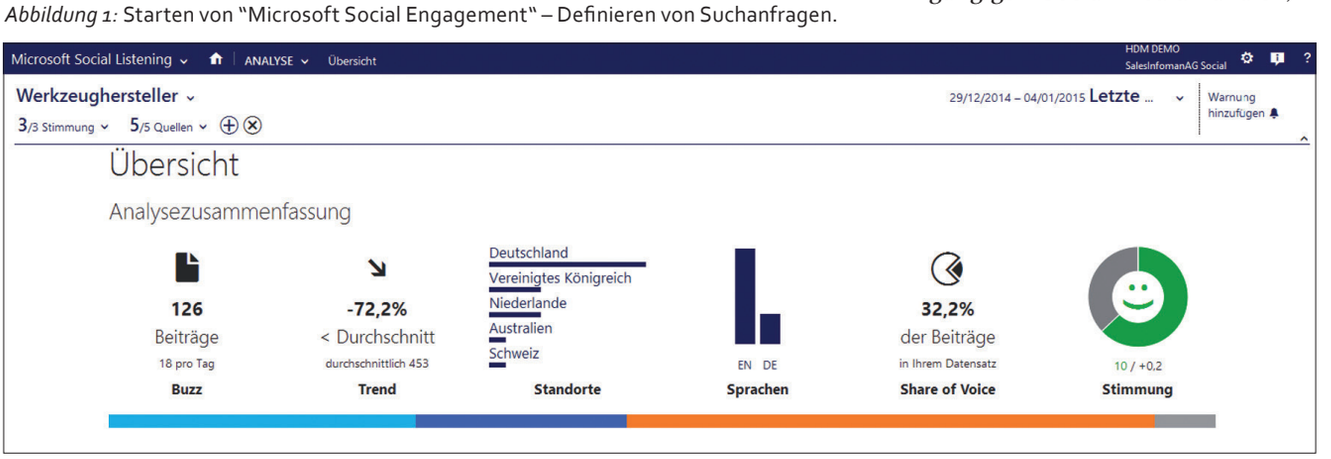


Abbildung 2: Starten von "Microsoft Social Engagement" – Übersicht einer Suchanfrage.

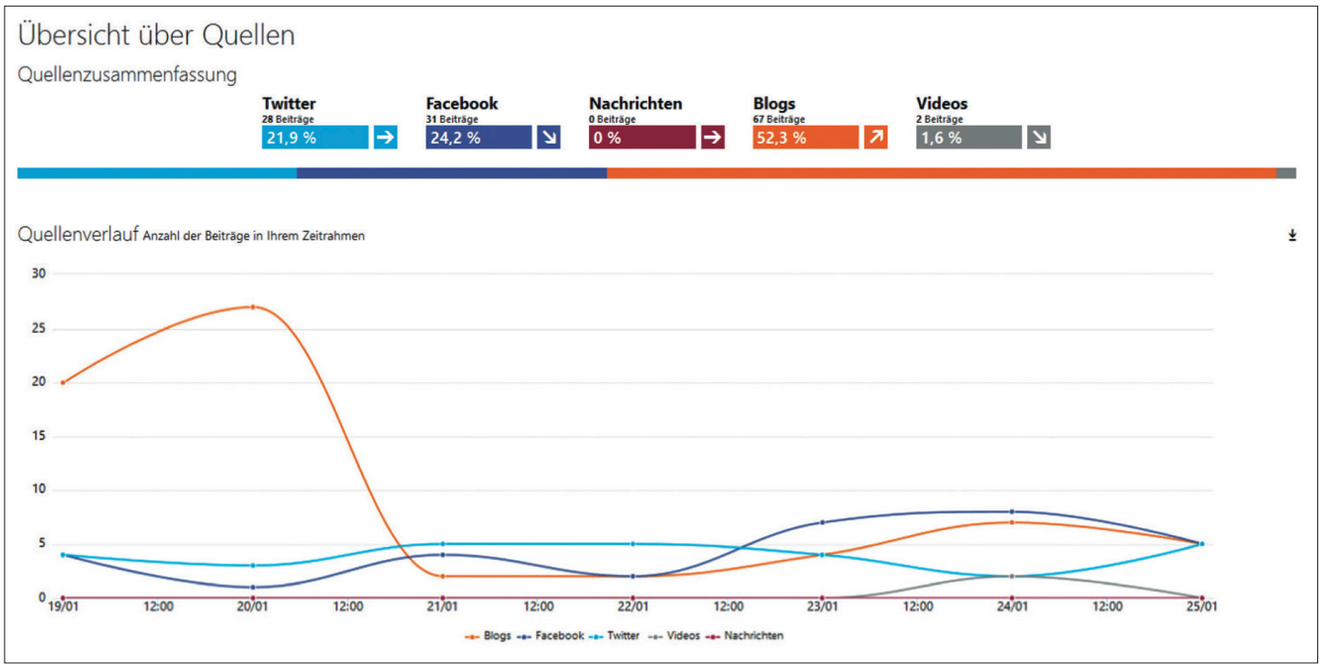


Abbildung 3: Starten von "Microsoft Social Engagement" – Übersicht über Quellen.

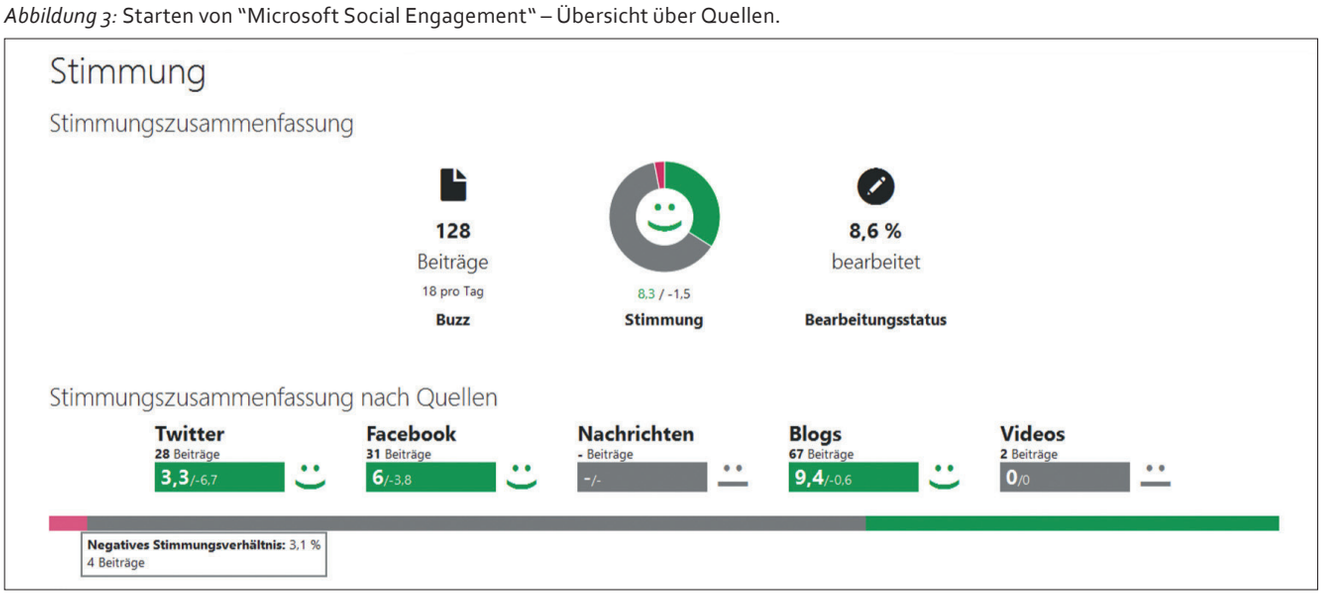


Abbildung 4: Benutzung von "Microsoft Social Engagement" – Sentiment Analyse (Stimmung) Resultat.

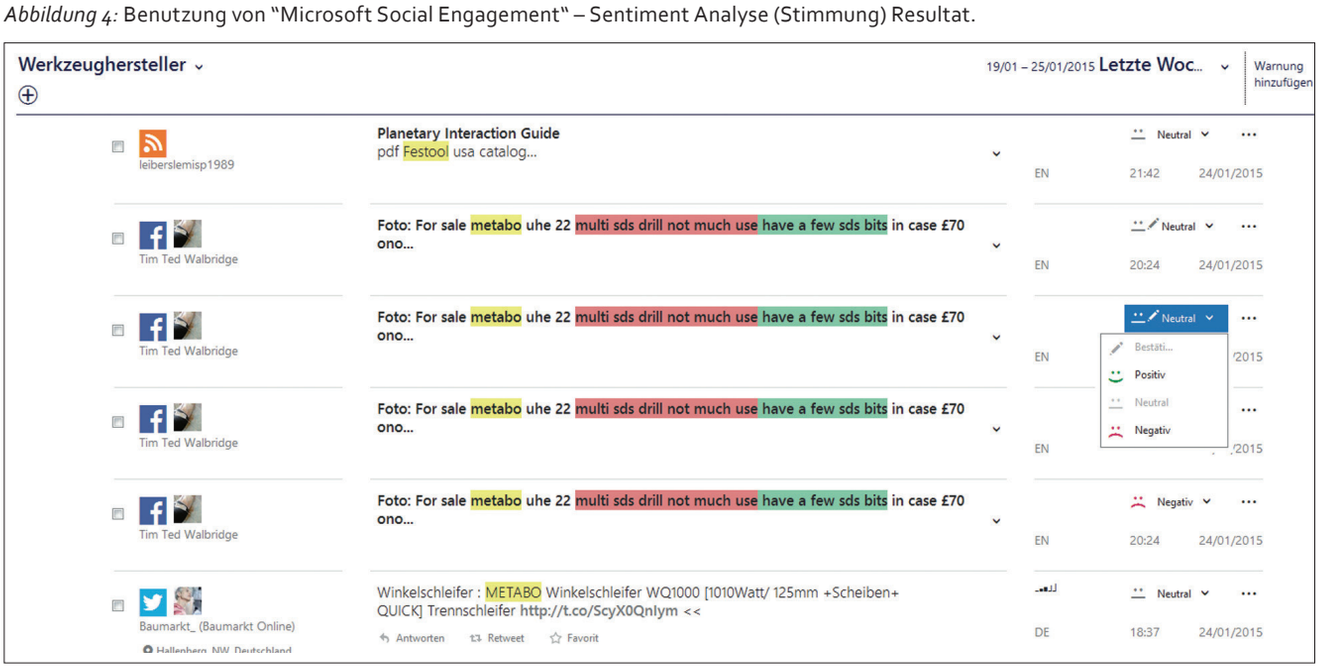


Abbildung 5: Benutzung von "Microsoft Social Engagement" – Kommentar Analyse Resultat.

- Monitoring und Modifikation von Kampagnen,
- Wettbewerber Monitoring und Wettbewerberanalyse.

Beim Start von "Microsoft Social Engagement" erhält der Benutzer eine Übersicht von bereits vorhandenen Suchanfragen, um diese hinsichtlich enthaltener Metadaten modifizieren zu können, siehe Abbildung 1. Für einen Marketingexperten ist es wichtig zu definieren, welche Themen analysiert werden sollen, nach welchen Kategorien die Resultate zu kategorisieren sind (z.B. Sprache, Quelle, Land), welche Kanäle für die Identifizierung in Frage kommen und sicherlich welche *Keywords* beinhaltet und welche *Keywords* ausgeschlossen werden sollen. In unserem Anwendungsfall haben wir beispielsweise die Suchanfrage [Werkzeughersteller] kreiert, welche die Schlüsselwörter [Festool] + [Metabo] + [Fein] innerhalb von englischen und deutschen Beiträgen auf Blogs, Facebook und Twitter absucht.

Als erstes Resultat einer Suchanfrage erhält der Benutzer eine Übersicht der Analyse, siehe Abbildung 2. Diese enthält Informationen bezüglich der Anzahl gefundener Beiträge, einen Durchschnittstrend im Vergleich zu ähnlichen Suchanfragen, in welchen Ländern die Beiträge gefunden wurden, in welcher Sprache die Beiträge verfasst wurden, welche Datenquelle die meisten Beiträge enthält und letztendlich ein durchschnittliches Resultat der Sentiment Analyse. In unserem konkreten Beispiel erhielten wir 126 Beiträge für die Suchanfrage aus Abbildung 1, welche im Zeitraum 29.12.2015 bis 04.01.2016 durchgeführt wurde. Des Weiteren erschließt sich in unserer Anfrage, dass die meisten Beiträge in Englisch verfasst wurden. Die Sentiment Analyse resultiert eine überwiegend positive Stimmung.

Der nächste wichtige Reiter liefert detailliertere Ergebnisse zu den angeschlossenen Suchquellen, siehe Abbildung 3. Hier kann der Benutzer sehen, wie viele Beiträge in den angeschlossenen *Social-Media* Kanälen gefunden wurden, um die Datenquelle mit den meisten Treffern zu identifizieren. In unserem konkreten Beispiel fanden wir die meisten Beiträge auf Blogs, gefolgt von den Datenquellen Facebook und Twitter. Ergänzend sehen wir im Diagramm, dass zu Beginn des angefragten Zeitfensters mehr Kommentare gefunden wurden.

3. BENUTZUNG VON "MICROSOFT SOCIAL ENGAGEMENT"

Die Sentiment Analyse wird genutzt, um die Stimmung für angefragte Schlüsselworte (z.B. [Festool] + [Schrauber]) zu resultieren, siehe Abbildung 4. Genauer, welche Kommentare in den Social-Media Kanälen welcher Stimmung zugeordnet werden können. In "Microsoft Social Engagement" kann hierfür jeder Kommentar ergänzend einzeln begutachtet werden, um z.B. irrelevante Kommentare manuell auszufiltern, siehe Abbildung 5. Hier können wir einen Beispielkommentar sehen, welcher wesentlich durch eine schlechte Stimmung geprägt ist.

Die Spracherkennung von derartigen Tools spielt die prägnanteste Rolle, da diese über die Qualität der Sentiment Analyse entscheidet. "Microsoft Social Engagement" erwies sich hinsichtlich dieses Kriteriums als wettbewerbsfähig aber nicht ausgereift, siehe Abbildung 6. Beispielsweise werden Kommentare mit doppelter Negation (z. B. [nicht schlecht]) als positives Statement erkannt. Jedoch war dies in unserem Beispielszenario nicht immer der Fall. Es muss deshalb zwingend eine manuelle Korrektur für falsch klassifizierte Kommentare vorgenommen werden. Ergänzend zu den bereits erwähnten Reitern, können noch die Autoren der gefunden Beiträge und die Beiträge nach Ländern übersichtlicher dargestellt werden.

4. WETTBEWERBER VON MICROSOFT SOCIAL ENGAGEMENT"

4.1 "BRANDWATCH ANALYTICS"

"Brandwatch Analytics" kann Informationen von über 50 Millionen Datenquellen verarbeiten und 25 Sprachen erkennen. Ähnlich "Microsoft Social Engagement" arbeitet "Brandwatch Analytics" mit dem vom Benutzer definierten Suchanfragen. Diese können automatisiert auch irrelevante Treffer wie Duplikate und Spam (z.B. Vulgarismus) filtern. "Brandwatch Analytics" ist speziell auf Agenturen ausgerichtet, welche große Datenmengen analysieren, da nur derartige Kunden die breite Funktionspalette des Systems wirklich effektiv ausschöpfen können:

Abbildung 6: Benutzung von "Microsoft Social Engagement" – Beiträge Analyse Resultat.

- Übersicht für das *Monitoring* von Web-Gemeinschaften,
- Datenakquise, Datenanalyse und Resultatpräsentation,
- Nachverfolgung von *Influencers* und sensibler Informationen,
- Sentiment-Analyse mit drei Stimmungskategorien (positiv, neutral, negativ),
- *Monitoring* und Modifikation von Kampagnen,
- Wettbewerber *Monitoring* und Wettbewerberanalyse,
- Direkte Schnittstelle zu allen gängigen *Social-Media* Kanälen,
- Individuelle Benachrichtigungsfunktionen für verschiedene Benutzerrollen.

4.2 "BUZZSUMO PRO"

"Buzzsumo PRO" wird als "Content Marketing Intelligence Tool" angeboten, welches ebenfalls für das *Monitoring* von *Social-Media* Kanälen eingesetzt wird. Das System bezieht sich auf *Top-Contents* (Deutsch: wichtigste Inhalte) und *Top-Shares* (Deutsch: am häufigsten verteilte Inhalte), um Sentiment Analysen für Marken, Produkte, Kampagnen und Wettbewerber durchzuführen. Die Datenakquise wird mithilfe von Schlüsselworten durchgeführt. Filtermethoden und der Ausschluss von Schlüsselworten ist nicht möglich. Das System bietet neben einer individuellen graphischen Datenakquise und Echtzeit Benachrichtigungen folgende Funktionen:

- *Top-Content* und Beeinflusser Analyse mithilfe von Schlüsselworten,
- Übersicht von *Top-Shares* für Blogs,
- Vergleich von unterschiedlichen Domains (Datenquellen) und Wettbewerbern.

4.3 "SOCIALSEARCHER"

"Social Searcher" ist eine *Open-Source* Suchmaschine für *Social-Media* Kanäle, welches Sentiment Analysen in Echtzeit durchführen kann. Die Premiumversion unterstützt neben dem Speichern von Suchanfragen, Ausführen von Benachrichtigungen sowie dem Export von Resultaten, die folgenden Funktionen:

- Interpretierung von Kommentaren in Twitter, Facebook und Google+,
- Suchresultate können anhand Inhalt kategorisiert werden (Links, Bilder, Videos),
- Fortgeschrittene Suche und Unterstützung von Mehrsprachigkeit.

5. FAZIT "MICROSOFT SOCIAL ENGAGEMENT"

Zusammenfassend können wir "Microsoft Social Engagement" als umfassendes *Social Listening* System bewerten, welches sich auf Augenhöhe mit den bereits etablierten *Social Listening* Systemen befindet. Es bietet unterschiedliche Funktionen zur Analysierung verschiedenster Datenquellen und Inhaltstypen. Die Struktur des *Dashboards* ist übersichtlich aufgebaut und erlaubt eine rasche Einarbeitung, insofern der Benutzer mit dem *Look and Feel* von Microsoft Produkten vertraut ist. Die Filtermöglichkeiten zur Realisierung benutzerdefinierter Analysen (Zeitspanne, Schlüsselworte, Datenquellen) sind wettbewerbsfähig. Die Sentiment-Analyse musste in unserem konkreten Beispiel oftmals manuell modifiziert werden, um die Sinnhaftigkeit der Resultate zu steigern.

HEIKO ANGERMANN ist Dozent an der Hochschule der Medien und University of the West of Scotland in den Bereichen "Metadaten Management" und "Content Management". Er ist Betreiber des Beratungshauses "HA-ECM" (Internet: *www.ha-ecm.com*; E-Mail: *angermann@ha-ecm.com*), sowie Autor, Sprecher und Trainer im Themenumfeld Microsoft SharePoint, war als SharePoint Administrator für einen Microsoft Gold Partner tätig und leitete internationale SAP-Projekte im Bereich Produkt-Daten-Management.

NAEEM RAMZAN
ist Dozent an der University of the West of Scotland und Autor von "Social Media Retrieval". Er ist in europaweite Industrie- und Forschungsprojekte im Themenfeld "Social Media" involviert und im Bereich "Netzwerke" tätig.

[1] https://www.w3.org/standards/semanticweb/
[2] https://www.w3.org/RDF/
[3] http://www.zdnet.de/88148107/
[4] https://www.brandwatch.com
[5] https://www.brandwatch.com/brandwatch-analytics/
[6] http://buzzsumo.com/
[7] http://www.social-searcher.com/
[8] http://www.festool.com

FEST VERSCHLOSSEN

Anwendungen mit Lizenzbibliotheken absichern

Lizenzen sind ein wichtiges Thema, wenn es um Software geht. Nicht nur aus Sicht der Infrastruktur für den Betrieb von Software, die für den alltäglichen Einsatz notwendig ist. Sondern auch bei der eigenen Anwendung, die durchaus vor dem unbefugten Zugriff Dritter geschützt sein soll. Lösungen dafür gibt es unzählige auf dem Markt. Von Hardware, über Software bin zu Client-Server basierten Strategien. Dieser Artikel gibt einen kleinen Überblick über mögliche Lösungen und deren Vor- und Nachteile.

Anwendungen schreiben kostet immens viel Zeit. Und wo ein hoher Zeitaufwand notwendig ist, sind hohe Kosten in der Regel auch nicht weit. Also kommen irgendwann die Themen Lizenzen, Lizenzsysteme, Lizenzierung und Softwareschutz im Allgemeinen auf den Tisch. Je früher desto besser, da solche Themen auch Implikationen auf die Software an sich haben, wie beispielsweise die Architektur, Implementierungen und allgemein technischer Lösungen.

Im Großen und Ganzen betrifft das nur kommerzielle Softwareprodukte. Beim Open Source Projekten ist die Lage etwas anders. Dort sind Lizenzen zwar auch ein Thema, allerdings in der Hinsicht, dass die Nutzung des Quelltextes reglementiert werden soll. In der Regel deutlich offener, als das bei kommerziellen Systemen der Fall ist. Der Artikel befasst sich mit Lizenzen aus der Sicht kommerzieller Software und zeigt mögliche Lösungen auf, diese Art von Softwareprodukten zu schützen.

BERÜHRUNGSÄNGSTE

Hand aufs Herz: Wer kümmert sich gerne um das Thema Lizenzen? Insbesondere aus Sicht einer möglichen Implementierung in der eigenen Anwendung? Es ist zugegebenermaßen kein sonderlich beliebtes Thema. Allerdings ist es dadurch nicht minder wichtig. Denn wer möchte schon, dass die eigene Anwendung einfach so kopiert wird. Investitionen wollen geschützt werden. Und am besten ohne die Kundenbasis zu vergraulen. Also muss eine kundenfreundliche Lizenzierung, ein gutes Lizenzmodell, sowohl technisch, organisatorisch als auch preislich und eine ausgereifte Implementierung her.

Die in diesem Artikel vorgestellten Techniken und Systeme sind eine Mischung aus frei verfügbaren Komponenten beziehungsweise Implementierungen, gemischt mit kommerziellen Angeboten. In der Hoffnung, jeder findet für seinen Anwendungsfall das richtige. Klar ist aber auch, dass ein einzelner Artikel nur eine kleine Übersicht liefern kann. Dafür sind deutlich zu viele Möglichkeiten und Lösungen vorhanden. Zumal sich Lizenzierungen auch im stetigen Wandel befinden. Wo es früher darum ging, Disketten zu kopieren, bei einem relativ kleinen Markt, lassen sich heutige selbst große Sammlung an Computerspielen ohne Probleme tauschen. Auch darauf müssen Lizenzierungssysteme reagieren.

VERSCHIEDENE SYSTEME

Wie oben schon kurz angedeutet, befindet sich die gesamte IT Branche in einem stetigen Wandel. Diese Änderungen müssen auch von Lizenzierungssystemen mitgemacht werden, ansonsten wird der angebotene Schutz schnell obsolet. Disketten zu schützen ist heute kein Thema mehr.

Es gibt verschiedene Möglichkeiten, eine Lizenz beziehungsweise ein Lizenzsystem als Ganzes zu realisieren:

- Dateien
- Hardware-Dongle
- Hardwarebasiert
- Onlinebasiert
- Cloudbasiert
- Spezielle Zeichenketten als Lizenzschlüssel
- Transparente Verfahren

Daraus resultieren verschiedene Möglichkeiten für konkrete Implementierungen der jeweiligen Lizenzsysteme. Aber auch organisatorisch gibt es vieles zu beachten. Denn Lizenzdateien an Kunden zu verteilen oder sich darum zu bemühen, funktionierende Dongles einzusetzen und auszuliefern, ist etwas anderes, als eine Cloud-Lösung anzubieten und diese vor dem unbefugten Nutzen schützen zu müssen.

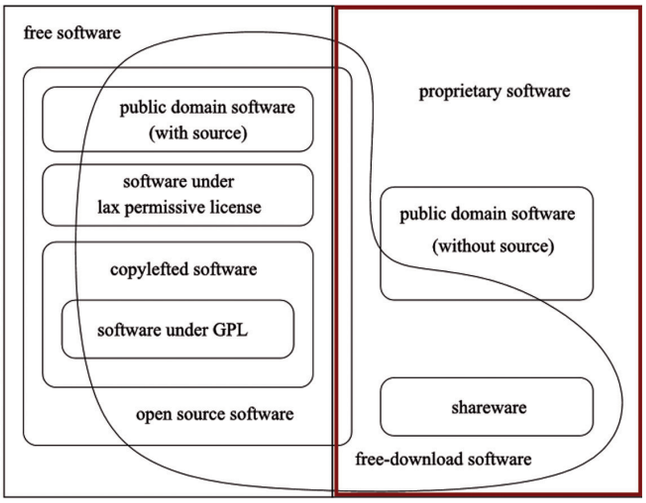


Abbildung 1: Verschiedene Arten von Softwarelizenzen.

Abbildung 1 zeigt eine kurze Übersicht über verschiedene Softwarelizenzen. Diese unterschiedlichen Lizenzen haben jeweils andere Anforderungen an die Lizenzierung an sich, aber auch an das dahinterstehende Lizenzmodell. Wie schon kurz in der Einleitung erwähnt, zielen die hier vorgestellten Lösungen auf den proprietären Bereich, der in der Abbildung durch den rot umrandeten Teil dargestellt wird.

QUELLTEXTBASIERTE LÖSUNGEN

Es gibt eine Reihe von quelltextbasierten Lösungen, die sich nutzen lassen, um eigene Anwendungen abzusichern. Zu diesen Lösungen zählen insbesondere Open-Source-Lösungen, bei denen der Quelltext nicht nur einsehbar ist, sondern auch durchaus angepasst werden kann.

Anderes als fertige Tools, die im nachfolgenden Kapitel beschrieben sind, dienen diese quelltextbasierten Lösungen eher dazu, sich selber mit der Materie Lizenzierung zu befassen. Ein Anbieter beziehungsweise Hersteller befindet sich dabei nicht im Hintergrund der in Tabelle 1 vorgestellten Lösungen. Das kann sowohl ein Vorteil sein, wie auch ein Nachteil. Änderungen sind eigenständig durchführbar, ohne dass allzu viel beachtet werden muss. Allerdings fehlt auch häufig eine helfende Hand oder gar ein Support, wenn nicht gerade die

Lösung	Kurze Beschreibung	Quelle
Component Licensing in .NET	Beispielsanwendung in C#, die .NET-Komponenten mit einem Lizenzschlüssel versieht.	http://www.codeproject.com/Articles/599509/How-To-Component-Licensing-in-NET
License Key Generator	Ein einfacher Generator für Lizenzschlüssel.	http://licensekeygenerator.codeplex.com/
Eindeutiger Fingerabdruck	Lizenzschlüssel an die verwendete Hardware koppeln.	http://www.codeproject.com/Articles/28678/Generating-Unique-Key-Finger-Print-for-a-Computer
SKGL - Serial Key Generating Library	Komplexere Methode, um Lizenzschlüssel zu generieren.	http://skgl.codeplex.com/
Rhino Licensing	Freies Lizenzierungsframework in C#.	https://github.com/ayende/rhino-licensing
Product Activation	Produktaktivierung über RSA Signaturen.	http://www.codeproject.com/Articles/16506/Product-Activation-Based-on-RSA-Signatures
Endjin.Licensing	Freies Lizenzierungsframework in C#, inspiriert von Rhino Licensing.	https://github.com/endjin/Endjin.Licensing

Tabelle 1: Quelltextbasierte Lösungen zur Lizenzierung von .NET Anwendungen.

Dokumentation der genutzten Lösung besonders gut ist. Diese Tabelle bietet allerdings nur einen kleinen Überblick und einige Stichpunkte, nach denen gesucht werden kann, um weitere Lösungen zu finden.

LIZENZIERUNGSTOOLS

Alle in der vorherigen Tabelle vorgestellten Möglichkeiten gehen eher in die Richtung, nur den Teil abzudecken, der sich um die Generierung einer Lizenz oder eines Lizenzschlüssels kümmert. Organisatorische Aspekte werden dabei außen vorgelassen.

Tabelle 2 bietet einen kleinen Überblick in Richtung Lizenzierungstools. Damit sind Werkzeuge gemeint, die nicht zwingend im Quelltext vorliegen müssen und zudem mehr die organisatorischen Aspekte berücksichtigen. Ganz überschneidungsfrei ist diese Unterteilung, die sich in den Tabellen 1 und 2 widerspiegelt, natürlich nicht. Auch Lizenzierungstools können im Quelltext zur Verfügung stehen und die Erfahrung zeigt, dass gerade diese Art von Lizenzierungsmöglichkeit häufig als Open Source-Projekte vorangetrieben wird.

DIE KLASSIKER

Beide Tabellen beinhaltet lediglich Open Source Lösungen, da der Artikel bewusst den Fokus auf dieses Spektrum von Lizenzierungsmöglichkeiten gelegt hat.

Selbstverständlich gibt zahlreiche kommerzielle Anbieter von Lizenzierungstools im Allgemeinen, die ebenfalls gute Dienste leisten. Die hier vorgestellten Tools und Möglichkeiten sind lediglich als Auswahl zu verstehen und nicht als Wertung gegenüber Closed Source vs. Open Source.

Allerdings ist einzuräumen, dass Klassiker aus dem Open Source-Bereich der Lizenzierungen derzeit sehr beliebt sind. Zu diesen Klassikern zählen unter anderem Rhino Licensing, Endjin.Licensing und Portable.Licensing. Alle drei Lösungen sind einfach zu bedienen, schnell in eigene Anwendungen integriert und decken mit den unterstützten Lizenztypen und Daten, die in Lizenzen eingebettet werden können, viele Anwendungsfälle ab.

Lediglich verteilte Szenarien, in denen viele Instanzen einer Anwendung lizenziert werden müssen, unterstützen die genannten Systeme nicht. Dafür muss händisch nachgebessert oder auf andere Anbieter ausgewichen werden.

FAZIT

Da sich das Angebot an Möglichkeiten und Lösungen im Bereich der Softwarelizenzierung beständig weiterentwickelt, ist ein aktueller Überblick auch recht schnell nicht mehr aktuell. Alle hier vorgestellten Lösungen erheben daher keinen Anspruch auf Vollständigkeit.

Leider macht es eine größer werdende Auswahl auch nicht leichter, den für den eigenen Anwendungsfall perfekten Schutz zu finden. Wenn es diesen perfekten Schutz überhaupt gibt. Denn beim Thema Lizenzierung ist es wie bei anderen Sicherheitsaspekten auch. Es geht immer um eine ausgewogene Lösung zwischen Sicherheit und Angemessenheit.

Persönlich habe ich sehr gute Erfahrungen mit Portable.Licensing gemacht, was mir in vielen Situationen schon enorm geholfen hat, Zeit, Nerven und Geld zu sparen. Allerdings waren das kleinere Lizenzierungs-Szenarien.

FABIAN DEITELHOFF lebt und arbeitet in Dortmund, der Metropole des Ruhrgebiets. Er studiert derzeit den Masterstudiengang Informatik mit dem Schwerpunkt Biomedizinische Informatik an der Hochschule Bonn-Rhein-Sieg in Sankt Augustin. Seine Schwerpunkte liegen in der Entwicklung von Visual Studio Erweiterungen, der Analyse und Beschreibung von Open Source Frameworks sowie im Rapid Prototyping. Beruflich ist er als freier Autor, unter anderem für Pluralsight, Sprecher und Softwareentwickler im .NET Umfeld tätig. Sie erreichen Ihn über seinen Blog *www.fabiandeitelhoff.de*, per E-Mail unter *Fabian@FabianDeitelhoff.de* oder auf Twitter als *@FDeitelhoff*.

Lösung	Kurze Beschreibung	Quelle
Portable.Licensing	Plattformübergreifendes Lizenzierungsframework in C#, das auch unterschiedliche Lizenztypen (Voll, Subscription) unterstützt.	https://github.com/dnauck/Portable.Licensing
Babelfor.NET	Verwaltetes Lizenzierungssystem, das unterschiedliche Lizenztypen beherrscht und ein Obfuscating anbietet.	http://babelfor.net/

Tabelle 2: Lizenzierungstools für die .NET Plattform.

EFFEKTIVES UNIT TESTING MIT FAKES

Unit Tests sind aufwändig. Der gesamte Code muss umgeschrieben werden um die Anwendung automatisiert Testen zu können. Dass dies nicht immer so sein muss zeigt das Fakes-Framework von Microsoft. Code, der anscheinend „untestbar“ ist, kann dennoch getestet werden.

War es früher noch normal, alle zwei Jahre eine neue Version herauszubringen, so muss heutzutage nach maximal drei Monaten bereits das nächste Update verfügbar sein. Auch Mobiltelefone, Tablets oder Notebooks müssen immer schneller ausgetauscht werden. Dieser Trend, alles möglichst schnell zu aktualisieren hat auch Auswirkungen auf die Softwareentwicklung, da die Entwicklungszyklen immer kürzer werden. Die Kunden erwarten in immer kürzeren Abständen neue Versionen und neue Features.

Wenn aber weniger Zeit für die Entwicklung bleibt, muss zwangsläufig irgendwo eingespart werden. Die erste Möglichkeit zur Einsparung besteht im Weglassen der Dokumentation. Wofür sollte man überhaupt eine Dokumentation schreiben, wenn diese in spätestens drei Monaten wieder veraltet ist? Und den Kunden interessiert es nicht, ob die Architektur sauber dokumentiert ist. Wer gar nie damit begonnen hat, seine Entwicklungsarbeit zu dokumentieren, kann aber durch das Weglassen der Dokumentation leider auch keine Zeit sparen.

Der zweite Bereich, der sich zur Einsparung von Zeit eignet, ist das Testen der Software. Allerdings fällt es hier dem Kunden sehr wohl auf, wenn die Software nicht ausreichend getestet wurde. Das sogenannte „Bananen-Prinzip“, die Ware reift beim Kunden, spricht sich mittlerweile dank Social Media schnell herum.

Es müssen also effiziente Möglichkeiten her, um die Software zu testen, aber dennoch nicht zu viel Zeit zu investieren. Leider geht es nicht ganz ohne Investition, aber wenn schon Ressourcen aufgewendet werden sollen, dann sollten sie sich möglichst in Grenzen halten.

Unit Tests können, wenn sie optimal eingesetzt werden, dazu beitragen, den Qualitätsstandard beizubehalten und beanspruchen dennoch wenig zusätzliche Entwicklungszeit. Beim Unit-Testing werden kleine Einheiten der Software getestet. D.h. es wird nicht von der Datenbank bis zur Oberfläche eine einzelne Funktionalität getestet, sondern nur ein Teil dieser Kette. „Button-Click“-Anwendungen, wo es zu einer Vermischung von Oberfläche, dem Click-Event, und der restlichen

Anwendungs-Logik kommt, eigenen sich daher eher weniger für Tests. Auch wird der Anwendungs-Code sehr schnell unübersichtlich und de facto nicht mehr wartbar. Dafür ist die Anwendung schnell erstellt, was dem Anfangsgedanken des Artikels entspricht.

Moderne Architekturen wie Model-View-Controller (MVC) oder Model-View-Viewmodel (MVVM) zielen darauf ab, den Oberflächen-Code von der restlichen Logik zu trennen. Das bedeutet aber, dass die Basis für einfaches Testen schon in der Architektur der Anwendung gelegt wird.

In einem MVC Projekt übernimmt ein Controller die Steuerung zwischen dem Datenmodell und dem View. D.h. die Funktionen des Controllers laden die benötigten Daten aus einem Repository, ermitteln den anzuzeigenden View und geben sie inklusive der eingebetteten Daten zurück. Die Rendering-Engine erstellt danach aus View und Daten den erwarteten HTML-Code für die Anzeige im Browser.

Eine typische Action-Methode eines MVC Controllers wird in Listing 1 vorgestellt. Es werden anhand eines Suchwertes Kunden aus der Datenbank geladen. Sollten keine gefunden werden, wird ein View mit dem Namen „noResults“ geliefert. Sobald Kunden gefunden wurden, wird der View mit dem Namen „Search“ und der Liste der gefundenen Kunden als Datenmodell des Views zurückgegeben.

Um diese Methode zu testen, müssen mehrere Testmethoden geschrieben werden. Jede Verzweigung des if-Befehls muss in einem eigenen Test abgeprüft werden. Das erklärte Ziel ist es, eine Code-Coverage von 100% zu erreichen. Das bedeutet, dass alle Testmethoden zusammen jede einzelne geschriebene Zeile des produktiven Codes durchlaufen. Visual Studio bietet eine Möglichkeit um die Code-Coverage des Testprojektes ermitteln zu lassen. Aus dieser Anforderung heraus ergibt sich die Notwendigkeit, beide Pfade der If-Verzweigung in eigenen Methoden testen zu lassen.

In Listing 2 befindet sich die Test-Methode für den Fall, dass keine Ergebnisse gefunden werden. Eine Testmethode sollte nach dem Arrange-Act-Assert-Prinzip aufgebaut sein. Zunächst wird im Arrange-Bereich die Umgebung für den Test vorbereitet. In diesem Fall wird eine Instanz des Controllers gebildet, die zumeist als Target bezeichnet wird.

Im Act-Bereich wird die Search-Methode mit einem Parameter aufgerufen, bei dem wir davon ausgehen, dass keine Ergebnisse geliefert werden. Der Rückgabewert wird in eine Variable gespeichert. Im dritten Bereich, dem

Listing 1

```
public ActionResult Search(string searchString)
{
    var ctx = new NorthwindEntities();

    var res = from c in ctx.Customers
            where c.CompanyName.Contains(searchString)
            select c;

    if (res.Count() == 0)
        return View(„noResults“);
    else
        return View(res.ToList());
}
```

Listing 2

```
TestMethod()]
public void Search_noResults()
{
    // Arrange
    var target = new CustomersController();

    //Act
    var result = target.Search(„abcdef“);

    //Assert
    Assert.IsInstanceOfType(result,
typeof(ViewResult));

    var res2 = result as ViewResult;
    Assert.AreEqual(„noResults“, res2.ViewName);
}
```

Assert-Bereich, wird der Rückgabewert geprüft. Zunächst prüfen wir, ob der Datentyp richtig ist, und dann, ob der Name des Views richtig ist. Die Prüfung, ob der View den richtigen HTML Code aus diesen Daten erstellt, ist nicht Aufgabe in diesem Unit-Test.

Ein zweiter notwendiger Test besteht darin, eine Abfrage abzusetzen, die Ergebnisse liefert um dann zu prüfen, ob das Ergebnis tatsächlich dem erwarteten Ergebnis entspricht. Da die verwendete Datenbank die Nordwind-Datenbank von Microsoft ist, kann nach dem Begriff „Alfred“ gesucht werden, da wir auf jeden Fall einen Treffer zurückgeliefert bekommen: „Alfreds Futterkiste“. Genau diesen Fall prüft die Testmethode in Listing 3 ab.
Dieser Test wird positiv durchlaufen, solange die Datenbank unverändert ist. Es kann vorkommen, dass die Datenbank verändert wurde und ein weiterer Datensatz eingefügt wird, der mit dem Suchbegriff „Alfred“ gefunden werden könnte. Es gibt auch noch eine Vielzahl anderer Möglichkeiten die Datenbank so zu verändern, sodass die Controller-Methode korrekt arbeitet, aber der Test fehlschlägt. Diesen Fall nennt man auch „False-Negative“. Der Test wird in so einem Fall negativ beendet, obwohl die zu testende Methode eigentlich korrekt arbeitet.

Um diesen Problemen vorzubeugen ist es notwendig, die Datenbank immer vor einen Test in den Ausgangszustand zu versetzen. Das kann auch mit einem Restore der Datenbank vor der eigentlichen Test-Ausführung geschehen. Alternativ kann auch jedes Mal eine neue Datenbank angelegt werden. Da aber jeder Test für sich alleine

Listing 3

```
[TestMethod]
public void Search_1Result()
{
    var target = new CustomersController();

    var result = target.Search(„Alfred“);

    Assert.IsInstanceOfType(result,
    typeof(ViewResult));
    var res2 = result as ViewResult;

    Assert.IsInstanceOfType(res2.Model,
    typeof(List<Customer>));

    List<Customer> ergListe = res2.Model as
    List<Customer>;
    Assert.AreEqual(1, ergListe.Count);
    Assert.AreEqual(„Alfreds Futterkiste“,
    ergListe[0].CompanyName);
}
```

ausführbar und korrekt sein soll, bedeutet das, dass vor jeder Testausführung die Datenbank verändert wird, was je nach Größe der Datenbank ein paar Sekunden dauern könnte. Ein durchschnittliches Projekt kann schon mal mehr als 1000 Testfälle beinhalten. Wenn nun für jeden Test auch nur 5 Sekunden verloren gehen um die Datenbank zu manipulieren, sind das in der Summe fast 1,4 Stunden!

MOCKING

Zugriffe auf Datenbanken, oder die Einbindung von externer Hardware, wie zum Beispiel Sensoren-Messwerte, sind externe Einflussfaktoren, die eine Auswirkung auf den Ausgang der Testmethode haben. Diese werden als externe Abhängigkeiten bezeichnet. Wichtig ist es, keine externen Abhängigkeiten bei einem Unit Test zu haben. Das wird dadurch erreicht, dass die externen Softwaremodule durch sogenannte Mock-Objekte ersetzt werden. Ein Mock-Objekt hat dasselbe Interface wie das Original, liefert aber bei den Methodenaufrufen nur vordefinierte Rückgabewerte. So kann der Zugriff auf Daten der Sensoren einfach durch ein Mock-Objekt simuliert werden. Der Testlauf ist dann nicht mehr vom tatsächlichen Messwerten der Sensoren abhängig.

Mittlerweile haben sich viele Mocking-Frameworks etabliert, wie zum Beispiel „Moq“[1] oder „Rhino Mocks“[2]. Diese Frameworks ermöglichen es, aus einem Interface ein Mock-Objekt zu erstellen, welches die Methoden des Interface implementiert, ohne dass der Entwickler dafür Code schreiben muss.

Letztendlich ist es egal, wie man zu den Mock-Objekten gelangt, sei es durch manuelles Programmieren, oder durch ein Framework. In einem Testlauf muss der zu testende Code das Mock-Objekt verwenden und nicht das Original. D.h. muss das Original-Objekt durch das Mock-Objekt ausgetauscht werden. Prinzipiell gibt es mehrere Varianten, wie diese Injection durchgeführt werden kann. Aber alle existierenden Varianten setzen eine entsprechende Architektur der Anwendung voraus. So könnte beispielsweise der Datenbankzugriff in eigene Repository-Klassen ausgelagert werden. Die Erstellung der Repository-Objekte übernimmt hierbei eine Factory-Klasse. Die Factory-Klasse wird in diesem Fall so implementiert, dass mittels Code festgelegt werden kann, ob das Original, d.h. die Klasse mit den Zugriff auf die Datenbank, oder ein Mock-Objekt erstellt werden soll. Genau auf diese Art und Weise sollte die Anwendung aufgebaut werden!

Was ist aber, wenn zuerst mit der Entwicklung der Anwendung begonnen wurde, ohne bei der Architektur auf die Testbarkeit zu achten? Manchmal steht der Nutzen in keiner Relation zum Aufwand um dafür alles noch einmal abzuändern. Aber ganz auf die Tests zu verzichten geht in vielen Fällen aber auch nicht.

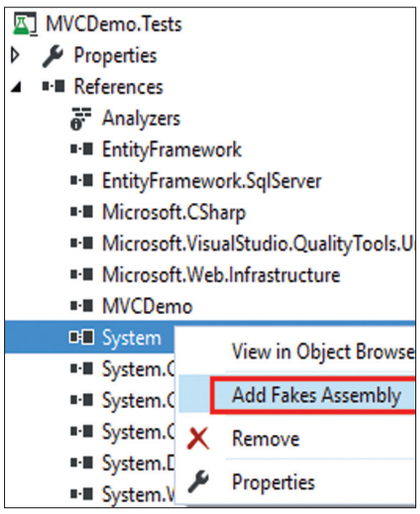
FAKES-FRAMEWORK

Hier hilft das Fakes-Framework, das in Visual Studio ab der Version 2013 enthalten ist, weiter. Ursprünglich wurde das Projekt von Microsoft Research unter dem Namen Moles vorangetrieben. Bereits unter Visual Studio 2010 war die Erweiterung Pex & Moles verfügbar [3]. Nun hat es das Projekt in den offiziellen Support geschafft und ist Bestandteil der höchsten Visual Studio Versionen, also Visual Studio 2013 Ultimate und Visual Studio 2015 Enterprise.

Das Fakes-Framework ist ein Rerouting-Framework. Das bedeutet, dass alle Methodenaufrufe in DotNet-Assemblies auf eigene Funktionen umgeleitet werden können. Diese Assemblies können selbst erstellte Assemblies sein, aber auch die System-DLLs können auf diese Art und Weise manipuliert werden. Der Entwickler kann für jede bereits implementierte Methode eine andere Funktion hinterlegen. Wird dann die Orginal-Funktion aufgerufen leitet das Fakes-Framework den Aufruf zu der, vom Entwickler, hinterlegten Funktion um. Und das alles ist ohne Eingriff in den bestehenden Code möglich, da das Rerouting tief im DotNet-Framework geschieht.

Um den Aufruf DateTime.Now umzuleiten, ist es notwendig, Shim-Klassen anzulegen. Dies geschieht im Visual Studio Projekt über das Kontext-Menü der Referenzen. Dort kann bei einer referenzierten DLL der Punkt „Add Fakes Assembly“ gewählt werden.

Von Visual Studio werden nun die Fakes Assemblies angelegt und dem Projekt hinzugefügt.



Innerhalb der Test-Methode wird nun mit ShimsContext.Create() ein neuer ShimsContext angelegt. Bis zum Dispose dieses Contextes besteht nun die Möglichkeit, jede Funktion durch eine eigene zu ersetzen. Properties werden in Get und Set Methoden abgebildet. Daher gibt es nun mit System.Fakes.ShimDateTime.NowGet die Möglichkeit, dort eine neue Funktion zu hinterlegen (siehe Listing 4). Das Assert wird in diesem Fall erfolgreich sein, da nun innerhalb des Usings-Blocks jeder Aufruf von DateTime.Now, von der hinterlegten NowGet-Methode abgearbeitet wird. Dies gilt auch, wenn der Aufruf von NowGet in einem anderen Assembly, wie zum Beispiel der zu testenden Assembly geschieht.

GEFAKTES ENTITY-FRAMEWORK

Mit diesem Wissen können wir nun den Controller testen, ohne auf die Datenbank zugreifen zu müssen. Das reduziert die Laufzeit der Tests und der Testlauf ist somit unabhängig vom Datenbankzustand. Es geht

Listing 4

```
using (ShimsContext.Create())
{
    System.Fakes.ShimDateTime.
    NowGet =() =>
    { return new DateTime(2000, 1,
1); };

    Assert.AreEqual(new
    DateTime(2000, 1, 1),
            System.DateTime.Now);
}
```

Listing 5

```
[TestMethod]
public void Search_1Result_mitFakes()
{
    //Arrange
    using (ShimsContext.Create())
    {
        ShimNorthwindEntities.AllInstances.CustomersGet = (ctx) =>
        {
            List<Customer> cList = new List<Customer>();
            cList.Add(new Customer { CompanyName="TestFirma1" });
            cList.Add(new Customer { CompanyName="Alfreds Futterkiste"});
            cList.Add(new Customer { CompanyName = „TestFirma2" });

           ShimDbSet<Customer> fakeCustomerDBSet = new ShimDbSet<Customer>();
            fakeCustomerDBSet.Bind(cList.AsQueryable());
            return fakeCustomerDBSet;
        };

        //Act
        var target = new CustomersController();

        var result = target.Search(„Alfred");

        //Assert
        Assert.IsInstanceOfType(result, typeof(ViewResult));
        var res2 = result as ViewResult;

        Assert.IsInstanceOfType(res2.Model, typeof(List<Customer>));

        List<Customer> ergListe = res2.Model as List<Customer>;
        Assert.AreEqual(1, ergListe.Count);
        Assert.AreEqual(„Alfreds Futterkiste", ergListe[0].CompanyName);
    }
}
```

sogar so weit, dass nun der Controller entwickelt werden kann obwohl wir keinen Zugriff auf die Datenbank haben.

Es ist an dieser Stelle keine Änderungen am Controller selbst notwendig. Alle Änderungen geschehen im Testprojekt. Als erstes müssen die Fakes-Assemblies vom EntityFramework und von unserer selbst geschriebenen DLL, dem Output unseres MVC-Projektes erstellt werden. Das Entity-Framework legt in unserem MVC Projekt eine Klasse namens „NorthWindEntities" an. Und von dieser Klasse müssen wir ein Shim-Objekt erzeugen. Daher ist es notwendig, auch von unserer eigenen DLL ein Fake-Assembly erstellen zu lassen.

Nachdem die Assemblies erstellt sind, kann nun die neue Test-Methode erstellt werden. Der Arrange-Teil wird nun deutlich länger. Das Testing-Framework von Visual Studio bietet auch die Möglichkeit, den Arrange Teil in eine separate Methode zu schreiben. Diese Methode ist mit dem Attribute „TestInitialize" zu kennzeichnen. Das Testing-Framework ruft dann vor jeder einzelnen Test-Methode diese Initialize-Methode auf. Nach der Testmethode wird noch eine „TestCleanup"-Methode aufgerufen. Dadurch ist es leicht, die „gefakte" Datenbank einmal zu erstellen und in vielen Tests zu nutzen.

Prinzipiell ist es egal ob die Shim-Objekte im Initialize oder in der Test-Methode selbst vorbereitet werden. Der Weg ist in beiden Fällen gleich. Es muss die Get-Methode der Customer-Liste ausgetauscht werden. Mit ShimNorthwindEntities.AllInstances kann für jede zukünftig gebildete Instanz eine Methode ausgetauscht werden. Innerhalb der Fake-Methode wird eine neue Liste von Customer-Objekten angelegt. Diese Objekte sind jene, die vom Entity-Framework nun als Datenbank-Einträge zurückgeliefert werden. Zuletzt muss noch ein ShimDbSet für die Customer angelegt werden. Die Liste der Customer-Objekte wird mittels Bind in das DBSet aufgenommen. Da die Liste als „AsQueryable" aufgenommen wurde, funktioniert auch die LINQ Abfrage, die im Controller die Kunden mit dem Suchausdruck abfragt.

FAZIT

Automatische Tests einer Anwendung helfen die immer kürzeren Entwicklungszyklen in der Softwareentwicklung zu meistern und dennoch eine hohe Qualität zu liefern. Das technische Applikationsdesign entscheidet, wie gut eine Anwendung automatisiert getestet werden kann. Wenn von Beginn ist klar ist, dass mit Unit Tests gearbeitet wird, kann die Anwendung entsprechend ausgelegt werden. Allerdings sind hierfür Vorkehrungen im Code notwendig, die ohne Unit Tests gar nicht den Weg in die Anwendung finden.

Das FakesFramework von Microsoft ermöglicht das Testen von Source-Code, obwohl keine Wege implementiert wurden, um Mock-Objekte einzuschleusen. Mit den Fakes-Objekten kann jedes DotNet-Assembly nachträglich im Verhalten verändert werden ,ohne das Assembly ändern zu müssen. Diese Möglichkeiten können eingesetzt werden, um die Datenbankzugriffe in den Testmethoden durch eigene Fake-Methode zu ersetzen und damit Tests, die unabhängig von der Datenbank laufen, durchzuführen.

LINKS

[1] http://www.moqthis.com/

[2] https://www.hibernatingrhinos.com/oss/rhino-mocks

[3] http://research.microsoft.com/en-us/projects/pex/Documentation.aspx

MARTIN GROBLSCHEGGs Spezialgebiete sind die .NET-Plattform, Microsoft SQL Server sowie Microsoft SharePoint. Er ist CTO der ppedv, MCT (Microsoft Certified Trainer), MCT Regional Lead für Österreich, sowie MCAD (Microsoft Certified Application Developer), MCTS (Microsoft Certified Technical Specialist) für SharePoint Development, SharePoint Administration, SQL Server Administration, SQL Server Business Intelligence und MCITP (Microsoft Certified IT Professional) für SharePoint Administration. Er gibt sein Wissen im Rahmen von Schulungen der ppedv AG und auf internationalen Konferenzen weiter. Martin Groblschegg ist Sprecher auf der VSone.
E-Mail: MartinG@ppedv.at

UNIVERSAL WINDOWS PLATFORM API

Der ehrgeizig kompromisslose Steven Sinofsky und sein ebenso ehrgeiziges Windows 8 Projekt sind gescheitert. Geblieben ist uns eine Erblast, die bis Windows 10 nachwirkt. Die WinRT API ist bis heute Basis für Windows 10 Universal Apps. Warum das so ist und worauf man als Entwickler achten muss, beleuchtet Hannes Preishuber.

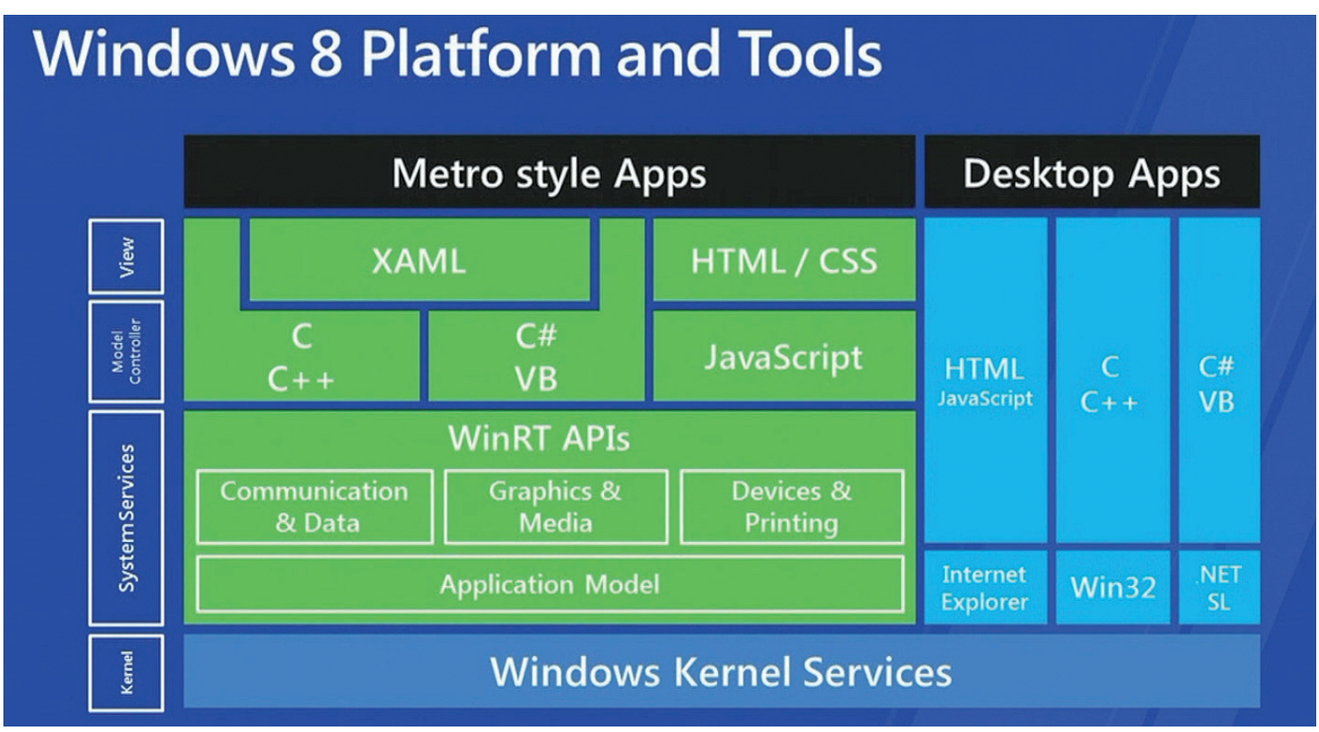
Aus vielerlei Gründen wurde in der Entwurfsphase von Windows 8 versucht, die ausführende Anwendungslogik näher ans Betriebssystem zu bringen. Windows ist nach wie vor COM basiert. .NET benötigt sogenannte Wrapper um mit diesen COM Objekten umzugehen. Es wird behauptet und ist auch nachvollziehbar, das bei diesem Übertritt durch Prozessgrenzen hindurch und der dazu nötigen Typumwandlung, dem Marshalling, Performance verloren geht. Das wirkt sich bei mobilen Geräten wie Smartphones wesentlich spürbarer aus, als bei einem Desktop PC, der am Stromnetz hängt.

Allerdings, so wird jedenfalls in den Insider Kreisen erzählt, gibt es auch zwei weitere Gründe. Einer davon ist zutiefst menschlich. Vor langer Zeit, waren zwei Abteilungen bei Microsoft wenig kooperativ. Die sogenannte WinDiv (Windows Developing Divison) und die DevDiv (Developer Divison). Letztere kümmerte sich um Visual Studio und .NET und erstere um Windows als Betriebssystem. Die Gerüchte berichten von einer Geringschätzung des .NET Stacks in der Abteilung WinDiv. Faktisch hat Windows nie managed Code beinhaltet, sondern basiert meist C++. Im Zuge der Windows 8 Aktivitäten soll es dabei 2010 zum Gemetzel gekommen sein, dem auch Silverlight zum Opfer fiel zugunsten von HTML. Jedenfalls wurde dann kurz vor Release im Juni 2011 der ersten Beta von Windows 8, das XAML Team in die WinDiv integriert und durfte in einer Hauruck Aktion den .NET Stack auf Windows 8 portieren. Ursprünglich sollte wohl ausschließlich JavaScript und HTML5 als App Plattform dienen.

Der zweite Teil dieser Geschichte dreht sich auch um JavaScript. Durchaus kluge Geister haben bei Microsoft vorausgesagt, das die JavaScript Community die nächsten Jahre den Ton angeben wird. Die alte Freund –Feind Betrachtung, schon zu Zeiten von Bill Gates erfunden und von Steve Balmer perfektioniert, zieht bei der nachwachsenden Generation an Entwicklern nicht mehr. Also wird Microsoft geöffnet, setzt auf Standards und wird Teil der Open Source Community. Soweit der Strategiewechsel übrigens deutlich vor Satya Nadella, dem aktuellen CEO von Microsoft.

Nun biedert sich Redmond der Welt als Hüter des JavaScript Grals an und befördert diese Sprache als „first class citizen“ der Windows 8 Developer. Microsoft behauptet gegenüber seinen Bestandskunden was anderes, aber die Tatsachen sprechen eine andere Sprache. Exemplarisch sei genannt das Flyout Element, obwohl ziemlich fundamental, gab es nur bei HTML JavaScript Projekten. Auch die Observable Collection war erst später verfügbar und musste vorerst direkt per API IObservableVector abgebildet werden.

Das Ergebnis dieser Bemühungen kann man übrigens deutlich erkennen. Microsoft freundliche Entwickler setzten, wenn überhaupt, für Windows 8 Apps auf den XAML Stack und die Microsoft unfreundlichen Entwickler fragten nach dem Marktanteil der Plattform und entschieden sich dann für Phonegap bzw. Cordova.



In jedem Fall musste ob der Gräulichkeiten der objektbasierten Sprache JavaScript eine Reihe von Kompromissen beim API Design für Windows 8 eingegangen werden. Und sie nannten es WinRT, wohl in Anlehnung an die Win32 API.

Aus WinRT wird mit Windows 10 die Universal Windows Platform. UWP für Windows 10 ist ein .NET Subset, das auf dem Erbe der WinRT API aufsetzt. Über den Mechanismus der Projektion kümmert sich das Framework um eine effiziente Konvertierung von benötigten Typen. Der nicht unwesentliche Unterschied ist allerdings, dass ein Release kompiliertes App Paket gar kein .NET Framework mehr nutzt, sondern direkt ohne Umweg über Intermediate Language (IL) Binaries bereitstellt. Microsoft nennt dies .NET native.

ZIEL MULTIPLATTFORM, EIN USER INTERFACE!

In der Tat lässt sich eine UWP App auf dem ARM Prozessor eines Raspberry Microcomputer, einem Windows 10 Smartphone oder dem Desktop ausführen.

Um zu erkennen, ob man nun die UWP Api oder .NET verwendet, reicht ein Blick auf die Namensräume. Die .NET Klassen finden sich im Namensraum System und die WinRT API Klassen im Windows Namensraum. Dadurch existieren Zweigleisigkeiten von Klassen, die zwar ident benannt, aber unterschiedliche Funktion aufweisen. Wer beim Einbinden per Using oder Imports Direktive nicht aufpasst, landet so schnell in unerklärlichen Compiler Fehlern.

```
0 Verweise
Private Sub button2_Click(sender As Object, e As RoutedEventArgs
    Dim hostname = NetworkInformation.GetHostNames()

End Sub
d Class
```

Imports System.Net
Imports Windows.Networking.Connectivity

Objekte die aus WinRT direkt erzeugt werden, sind in der Regel COM Objekte. Mit all seinen Vor- und Nachteilen. Im Debug Modus stößt man als Entwickler beim Betrachten der Objekte darauf. Zur Laufzeit erhält der Benutzer im nicht behandelten Fehlerfall typische COM Exception HResult Dialoge.

```
0 Verweise
Private Sub button2_Click(sender As Object, e As Routed
    Dim hostname = NetworkInformation.GetHostNames()
End Sub
```

hostname {System.__ComObject}

Will man nun strategisch so viel .NET Code wie möglich weiter verwenden, muss bei jedem Objekt Zugriff von .NET auf COM die Projektion der Datentypen erfolgen. Als nächstes wird am Beispiel von Dateizugriffen, Sockets und Streams die Problematik erläutert und Lösungsvarianten aufgezeigt.

Mit Hilfe von Streams werden größere Datenmengen sozusagen während des Durchflusses bearbeitet. Wenn man eine Datei lesen und schreiben möchte, kann man diese in ein Byte Array laden und damit viel Speicher belegen. Ein Stream belegt mit Hilfe eines Buffers in der Regel nur einen Bruchteil davon.

Am Beispiel eines TCP Servers kann man die Problematik der API Unterschiede gut beschreiben. Möchte man eine Instanz eines TCPListeners als Socket Server erstellen, wird ein Visual Studio UWP Projekt dies als Fehler erkennen. Es hilft auch nicht den Namensraum System.Net zu referenzieren. Die adäquate UWP API Klasse nennt sich StreamSocketListener und weist unterschiedliche Funktionen auf. Es wird also nichts mit dem kopieren von Silverlight, WPF oder Winforms Code an dieser Stelle.

Eine eingehende Verbindung eines Clients zum UWP Socket Server, lässt sich per Event behandeln. Im Übergabe Parameter vom Typ StreamSocketListenerConnectionReceivedEventArgs ist die Referenz auf den Input und Output Stream enthalten. Allerdings im .NET untypischen IInputStream Interfacetyp. Dieses Interface wiederum setzt nur eine Methode voraus um die eingehenden Daten zu lesen– readasync. Erst über Erweiterungs-Methoden lassen sich die Streams in passende .NET Stream Objekte konvertieren. Die Konverter Methoden kommen aus dem Namensraum System.Io und der Klasse WindowsRuntimeStreamExtensions. Jedenfalls wird per AsStreamForRead ein Stream, der per Streamreader gelesen werden kann.

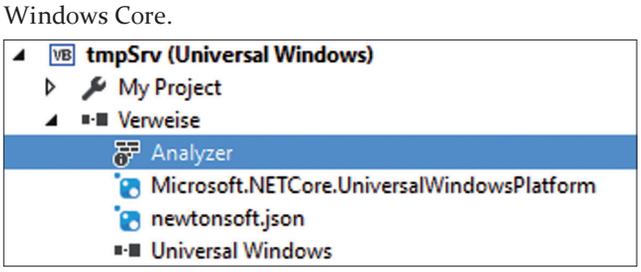
```
Dim rein = args.Socket.InputStream.AsStreamForRead
Dim reader As StreamReader = New StreamReader(rein)
Dim response = Await reader.ReadLineAsync()
```

Zum Vergleich nun der Code mit ähnlicher Funktion allerdings ohne die UWP API zu verlassen. Wie bereits ausgeführt lässt sich die API Funktion ReadAsync nutzen. Benötigte Parameter sind ein Buffer vom Typ IBuffer, die Länge als UInteger und die Options. Auf den ersten Blick gänzlich unverständlich erscheint die folgende Konvertierung eines IBuffers in ein .NET String Objekt. Hierzu wird auf eine API Funktion aus dem Kryptografie Bereich zurückgegriffen.

```
Dim readBuf = New Windows.Storage.Streams.Buffer(100)
Await args.Socket.InputStream.ReadAsync(readBuf, 100,
InputStreamOptions.Partial)
Dim response = CryptographicBuffer.ConvertBinaryToStrin
g(BinaryStringEncoding.Utf8, readBuf)
```

Wenn man sich die Mühe macht die Performance zu messen, sollte Code Schnipsel #2 schneller sein. Ob das in Zusammenhang mit TCP/IP Traffic eine Rolle spielt, mag bezweifelt werden.

Um die Kompatibilität mit existierenden Code aus Silverlight oder Windows Phone Projekten, auch in Hinblick auf .NET Core 1 zu erhöhen, erweitert Microsoft die UWP API laufend. So wurde erst vor kurzem System.Net um die Namensräume Socket und http erweitert. Allerdings kompiliert entsprechend kopierter Code aktuell im Visual Studio Projekt nicht. Was auch schon zum nächsten Teil führt. Anders als bei einer normalen .NET Anwendung finden sich in den Projektreferenzen keine Abhängigkeiten zu .NET Assemblies, sondern lediglich zu Windows Core.



```
Alle Pakete sind mit UAP,Version=v10.0 kompatibel.
Alle Pakete sind mit UAP,Version=v10.0 (win10-arm) kompatibel.
Alle Pakete sind mit UAP,Version=v10.0 (win10-arm-aot) kompatibel.
Alle Pakete sind mit UAP,Version=v10.0 (win10-x64) kompatibel.
Alle Pakete sind mit UAP,Version=v10.0 (win10-x64-aot) kompatibel.
Alle Pakete sind mit UAP,Version=v10.0 (win10-x86) kompatibel.
Alle Pakete sind mit UAP,Version=v10.0 (win10-x86-aot) kompatibel.
Die Paketwiederherstellung für "tmpSrv" wurde erfolgreich abgeschlossen.
"newtonsoft.json 8.0.2" wurde erfolgreich auf "tmpSrv" installiert.
```

Will man Erweiterungen nutzen, wie z.B. hier im Bild JSON.NET für das parsen von JSON Daten, wird dies per NuGet Paket nachinstalliert. Da es keine Intermediate Language mehr gibt, sondern nur .NET native, müssen diese Pakete die DLL's für alle möglichen Plattformen enthalten.

Zurück zur TCP/IP Verbindung per universeller Socket Klasse. Zwar kann man stand heute, die Socket Klasse aus dem Namensraum System.net instanziieren, aber das Code Beispiel nutzt zusätzliche Klassen wie DNS oder IPHostEntry. Um das aufzulösen müsste separat das Paket System.Net.NameResolution installiert werden, obwohl laut MSDN Dokumentation dies bereits im Namensraum System.Net enthalten sein sollte. Es gibt aber UWP typisch keine System.dll die diesen Code mitliefern könnte. Also per Versuch das NuGet Paket nachladen um die Nameresolution Funktionalität zu erhalten. Nur scheint dieses GitHub Paket nicht UWP App Container komplett zu sein und lässt sich vom Autor nicht installieren. Dieser Weg scheint aktuell noch nicht fertig.

Windows 10 führt UWP Apps in einem Container aus, mit stark reduzierten Zugriffsrechten. So kann man nicht direkt Dateien beliebig lesen und schreiben. Folglich unterscheidet sich die UWP API zwangläufig ganz wesentlich von klassischen .NET File Zugriff.

Zwar existieren die Hilfsklassen aus System.IO.File nun auch in UWP, allerdings darf man im UI Thread keine blockierenden Funktionen ausführen, die länger als 50ms dauern könnten. Um das zu umgehen, wird aus einer Zeile Code ein asynchroner Vorgang in einem separaten Task.

Im nächsten Schritt wird die gleiche Aufgabe per UWP API gelöst. Zunächst fällt auf, dass der Temp Ordner sozusagen pro App privat ist. Es ist so nicht möglich auf c:\temp zuzugreifen. Im letzten Schritt wird per Erweiterungsmethode aus dem IBuffer eine Byte Array.

Natürlich ist es wenig optimal eine komplette Datei in den Speicher zu laden, wenn es nicht nötig ist. Es wird nun ein Byte Array mit 200Bytes als Block Buffer verwendet und dann Segment per Segment in ein weiteres Byte Array eingelesen. Wie man sieht hat der dafür benötigte Code nahezu keine Ähnlichkeit zum vorigen Beispiel.

Allerdings wird dem aufmerksamen Leser nicht entgangen sein, dass das Byte Array keine direkte Eigenschaft der UWP API ist. Genau hier beginnt auch die Verwirrung bei vielen Entwicklern. Weil streng genommen muss man den Umweg häufig gar nicht gehen. Speziell wenn man von einer API Funktion die Daten an eine weitere übergibt, lässt sich ein IBuffer direkt als Parameter verwenden. Das Beispiel nimmt eine Datei, die direkt über eine Socket TCP/IP Verbindung weitergereicht werden kann, ohne sie komplett zu laden. Eben über Streams.

Die Herausforderung dabei ist, die mäßig dokumentierte und sich ändernde API von UWP im Überblick zu haben. Die Namensgebung tut ein Übriges dazu, dass man Code quasi nicht erraten kann. Die UWP API Objekte, die meist mit einem I beginnen, lassen sich zwar mit Extension Methoden umwandeln, aber unter Umständen wählt man dabei nicht den Performance optimalen Weg. Es somit ein gutes Stück schwieriger eine UWP App mit C# oder VB.NET zu schreiben als das mit Silverlight, um ein Beispiel zu nennen, möglich war.

```
Private Async Sub button_Click(sender As Object, e As RoutedEventArgs)
        Await Task.Run(Sub()
                    Dim array() As Byte = File.ReadAllBytes(„C:\temp\bild.
                    jpg“)
                      End Sub)
End Sub
```

```
Dim sf As StorageFile = Await ApplicationData.Current.TemporaryFolder.
GetFileAsync(„bild.jpg“)
Dim buffer As IBuffer = Await FileIO.ReadBufferAsync(sf)
Dim array() As Byte = buffer.ToArray
```

```
Dim sf As StorageFile = Await
Windows.Storage.KnownFolders.PicturesLibrary.GetFileAsync(„bild.jpg“)
Dim buffer As IBuffer = Await FileIO.ReadBufferAsync(sf)
Using dr = Windows.Storage.Streams.DataReader.FromBuffer(buffer)
      Dim array(250) As Byte
      While dr.UnconsumedBufferLength > 0
            dr.ReadBytes(array)
      End While
End Using
```

```
Dim fs = Await KnownFolders.PicturesLibrary.GetFileAsync(„bild.jpg“)
Dim b As IBuffer = Await FileIO.ReadBufferAsync(fs)
Dim r = Await client.OutputStream.WriteAsync(b)
Await client.OutputStream.FlushAsync()
client.Dispose()
```

HANNES PREISHUBER
ist CEO der ppedv AG und Microsoft-Experte (MCSD, MCAD, MCT) mit Schwerpunkt auf Web-Technologien. Er ist Sprecher, Trainer und Autor rund um Development-Themen.
E-Mail: hannesp@ppedv.de

Custom Controls in WPF Crashkurs

Bei den Custom-Controls handelt es sich im Vergleich zu den User-Controls um eine weitaus komplexere Variante, um eigene Steuerelemente für unsere Windows-Applikationen zu implementieren. Custom-Controls besitzen das gleiche *Lookless*-Verhalten wie die bereits in der WPF mitgelieferten Steuerelemente. Dementsprechend sind Logik und grafische Repräsentation dieser Steuerelemente strikt voneinander getrennt. Diese Eigenschaft bietet uns den Vorteil, dass bestehende Custom-Controls problemlos erweiterbar sind, indem von bereits bestehenden Custom-Controls abgeleitet werden kann. Darüber hinaus haben wir die Möglichkeit, die grafische Repräsentation unseres Custom-Control durch Anlegen eines Control-Template (Visual Tree) neu zu definieren bzw. problemlos zu erweitern. User-Controls hingegen können diese Fähigkeiten nur bedingt bieten. Hierbei können lediglich die einzelnen Bestandteile aus denen das User-Control zusammengesetzt ist, angepasst werden. Die Definition des Control-Template bleibt hierbei jedoch gleich. Dies mindert die Wiederverwendbarkeit der User-Controls, da die Repräsentation und die Logik stark miteinander gekoppelt sind. Suchen wir also eine erweiterbare Lösung, welche individuell anpassbar sein soll, so ist das Custom-Control zu wählen. Darüber hinaus bieten uns Custom-Controls die Möglichkeit, Templates für spezifische Windows-Themes festzulegen (wie beispielsweise das Aero-Theme in Windows Vista oder das Luna-Theme in Windows XP), welche je nach aktuell gewählten Windows-Theme für die Anzeige des Controls verwendet werden.

Listing 1: **CustomControl1**

```
public class CustomControl1 : Control
{
  static CustomControl1()
  {
     DefaultStyleKeyProperty.OverrideMetadata(typeof(CustomControl1), new
     FrameworkPropertyMetadata(typeof(CustomControl1)));
  }
}
```

Der nachfolgende Artikel befasst sich mit einer beispielhaften Implementierung eines *WPF Custom Controls*. Im Zuge dessen werden wir genauer auf die Besonderheiten von Custom-Controls eingehen, und hierbei besonders auf gängige Regeln bei der Implementierung dieser Control-Typen achten. Das Ziel wird es sein, ein eigenes TextBox-Control zu definieren und dieses um die Eigenschaften und Funktionalitäten eines Platzhalter-Textes zu ergänzen. Die Eigenschaft sowie das Verhalten eines Platzhalter-Textes sind möglicherweise bereits aus dem TextBox-Control der *Universal Windows Platform* Apps bekannt. Eine Vorschau für das im Rahmen dieses Artikels zu implementierende Control ist in Abb. 1 zusehen.

UNSER ERSTES CUSTOM CONTROL

Als erstes schauen wir uns an, wie ein solches Custom-Control in der WPF aufgebaut ist. Hierfür legen wir uns in Visual Studio eine neue Solution an, die ein neues Projekt vom Typ *WPF Custom Control Library* enthält (siehe Abb. 2).

Wie ihr euch vielleicht bereits denken könnt, dient die *WPF Custom Control Library* dazu, verschiedene Custom Controls in einer Assembly auszulagern, um diese später ggf. in anderen Projekten wiederverwenden zu können. Weiterhin besteht die

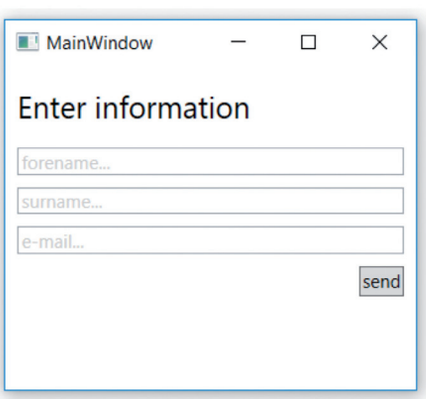


Abbildung 1: TextBox-Controls mit Platzhalter-Text

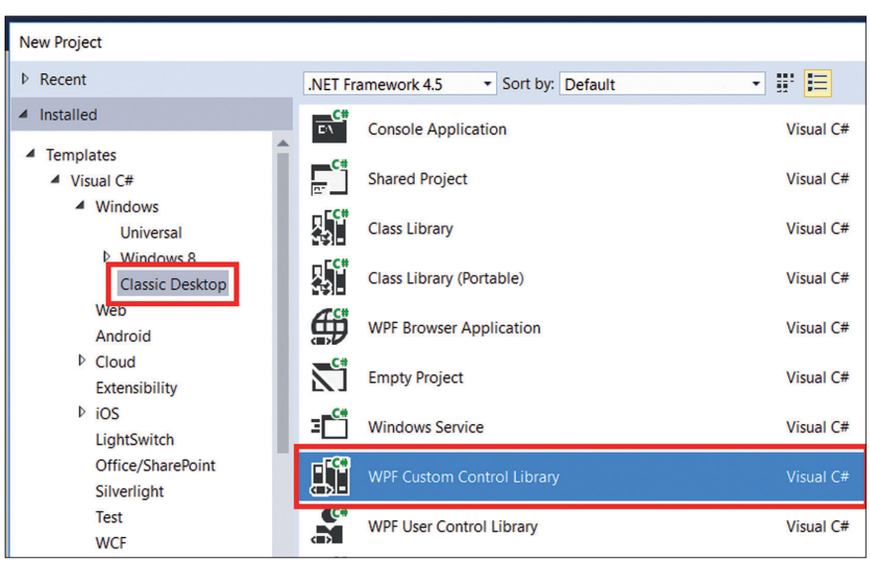


Abbildung 2: Erstellen einer WPF Custom Control Library

Möglichkeit einem bereits bestehenden Projekt ein Custom-Control hinzuzufügen. Nachdem wir uns eine *WPF Custom Control Library* erstellt haben, sieht unsere Projekt-Struktur im *Solution-Explorer* wie in Abb. 3 dargestellt aus.

In unserer Projektübersicht sind vorerst zwei Elemente für uns von besonderer Relevanz: Hierbei handelt es sich um die Dateien *CustomControl1.cs* und *Generic.xaml*. Die Datei *CustomControl.cs* ist eine gewöhnliche C#-Datei, in der wir die Eigenschaften und Methoden unseres Controls definieren können. Wenn wir diese Datei öffnen, sehen wir, dass Visual Studio standardmäßig Code für unser soeben erstelltes Custom-Control generiert hat (Listing 1)

Wenn wir uns die Klassen-Definition einmal genauer anschauen, können wir feststellen, dass unser Custom-Control von der Klasse *Control* erbt. Die Klasse *Control* repräsentiert die Basisklasse für UI-Elemente, welche es erlauben, das Erscheinungsbild mittels Control-Templates zu definieren. Das ist genau die

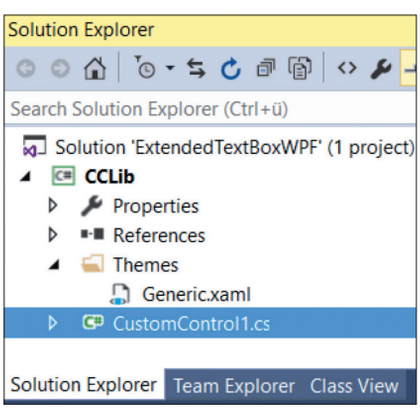


Abbildung 3: Projektübersicht

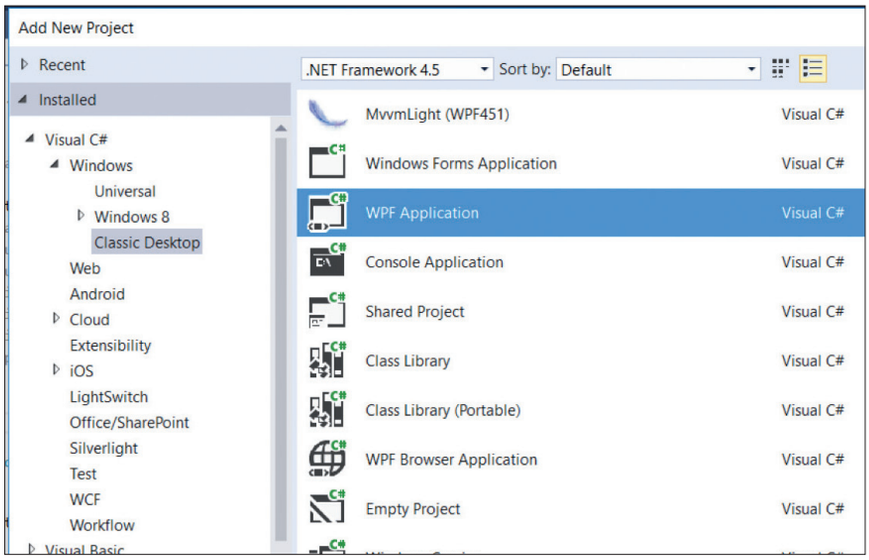


Abbildung 4: Erstellen einer Testanwendung

Eigenschaft, die wir später für unser eigens erstelltes TextBox-Control verwenden möchten. Des Weiteren stellen wir fest, dass uns Visual Studio bereits einen statischen Konstruktor für unser Control angelegt hat. Hierin werden durch die Anweisung *DefaultStyleKeyProperty.OverrideMetadata()* die Metadaten für den Standard-Style der Basisklasse *Control* überschrieben, sodass wir einen eigenen Style für unser Control festlegen können. Möchten wir einen eigenen Style für unser Control anlegen, so ist das Überschreiben der Metadaten notwendig. Als Parameter erwartet die Funktion *OverrideMetadata* zum einen den Typen unseres Controls sowie neue Metadaten vom Typ *FrameworkPropertyMetadata*.

Bei der Datei *Generic.xaml* im Ordner *Themes* handelt es sich um ein *ResourceDictionary*, in dem der Standard-Style für unser eigenes Control beschrieben wird. Die Verknüpfung dieses Standard-Style für unser Control findet über den Dateinamen *Generic.xaml* statt. Möchten wir Windows-Theme spezifische Styles anlegen, so müssen wir entsprechende Dateien mit passender Benennung im Ordner *Themes* anlegen. Wenn wir uns den Inhalt des *ResourceDictionary* genauer anschauen, stellen wir fest, dass sich hierin bereits ein vordefinierter Style für unser Custom-Control befindet. (Siehe Listing 2)

Wie wir hier sehen können, wird über einen Style-*Setter* das Standard Control-Template für unser Control gesetzt. Innerhalb des Templates befindet sich bereits ein *Border*-Control, dessen Eigenschaften *Background*, *BorderBrush* und *BorderThickness* auf die jeweiligen *DependencyProperties* der Basisklasse gebunden sind. Dies ermöglicht uns das Schlüsselwort *TemplateBinding*, hinter dem sich eine Kurzschreibweise für ein Binding mit relativer Quellenangabe verbirgt (Thema: *Binding RelativeSource*). Diese relative Quellenangabe ist notwendig, da unser Control das Template erst zur Laufzeit zugewiesen bekommt. Da der „Vater" unseres Templates unser Control selber ist, können wir hiermit aus dem Template heraus auf Eigenschaften unseres Controls binden. Im Grunde ist das, was uns Visual Studio automatisch angelegt hat, bereits ein voll einsatzfähiges Custom-Control, dessen Verwendung wir in einer kleinen WPF-Testapplikation testen können.

Listing 2: **Control Style**

```
<ResourceDictionary
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:CCLib">
    <Style TargetType="{x:Type local:CustomControl1}">
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="{x:Type
                local:CustomControl1}">
                    <Border Background="{TemplateBinding Background}"
                            BorderBrush="{TemplateBinding BorderBrush}"
                            BorderThickness="{TemplateBinding
                            BorderThickness}">

                    </Border>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>
</ResourceDictionary>
```

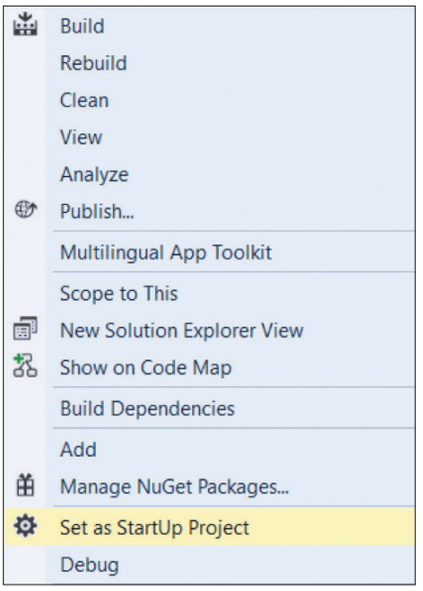


Abbildung 5: TestApp als Startprojekt festlegen

Hierfür legen wir uns in unserer Solution mit dem Namen *ExtendedTextBoxWPF* eine neue Anwendung vom Typ *WPF Application* an. Diese nennen wir in diesem Fall einfach *TestApp* (siehe Abb. 4).

Nachdem wir unsere Testanwendung erstellt haben, legen wir unsere *TestApp* zunächst noch als Startprojekt fest, damit wir diese auch ausführen und testen können. Dies geschieht über einen Rechtsklick auf unser TestApp-Projekt. Anschließend wählen wir aus dem erscheinenden Menü den Punkt Set as *StartUp* Project (siehe Abb. 5).

Damit wir unser zuvor erstelltes Custom-Control nun in unserer *TestApp* verwenden können, müssen wir dieser noch eine Referenz auf unsere *WPF Custom Control Library* mit dem Namen *CCLib* geben. Dafür machen wir einen Rechtsklick auf *References* unter *TestApp* im *Solution-Explorer* und klicken auf *Add Reference...*. Im sich nun öffnenden Dialogfenster ist unter dem Reiter *Solutions* unsere eigens angefertigte *CCLib* zu finden. Wir wählen die Checkbox neben unserer Bibliothek an und bestätigen den Dialog anschließend mit *OK*. Nachdem wir unsere Referenz hinzugefügt haben, fehlt noch ein letzter Schritt damit wir unser Control in unserer *TestApp* verwenden können: Wir müssen in unserer *MainWindow.xaml* den Namensraum für unser Control einbinden. Danach können wir über den soeben festgelegten Namensraum eine oder mehrere Instanzen unseres Control erzeugen. Nachdem wir in unserer *TestApp* den notwendigen Namensraum hinzugefügt sowie eine Instanz unseres Controls erzeugt haben, sieht der *Xaml-Code* des *MainWindow* wie in Listing 3 gezeigt aus.

Soeben haben wir unser eigenes kleines Custom Control erstellt und in einer lauffähigen Applikation verwendet. Führen wir die Anwendung nun aus bzw. betrachten den Designer, so stellen wir fest, dass die von uns gesetzten Eigenschaften *Background*, *BorderBrush* und *BorderThickness* von unserem in der Generic.xaml definierten Template übernommen wurden. Viel kann unser eigenes Control noch nicht, dies werden wir allerdings schon im nächsten Abschnitt ändern.

ERSTELLEN DES TEXTBOX-CONTROL

Nachdem wir den grundlegenden Aufbau eines Custom-Control behandelt haben, können wir uns nun der Implementierung der zu Anfang besprochenen *ExtendedTextBox* widmen, welche wir mit einer Platzhalter-Text Funktionalität und Eigenschaft ausstatten wollen.

Neben der Implementierung eines Controls von Grund auf, gibt es ebenfalls die Möglichkeit von bereits bestehenden Controls der WPF zu erben und diese um die benötigten Eigenschaften und Methoden zu erweitern. In den vordefinierten Controls der WPF handelt es sich ebenfalls um Custom-Controls. Diese Möglichkeit nutzen wir, wenn wir ein Control implementieren wollen, dessen Eigenschaften und Funktionalitäten bereits durch ein bestehendes Control teilweise abgedeckt werden. Schließlich wollen wir keine Arbeit doppelt verrichten. Für unsere Zwecke eignet sich das TextBox-Control der WPF hervorragend. Daher leiten wir unser eigenes ExtendedTextBox-Control anstatt von der Custom-Control Basisklasse *Control* von *TextBox* ab. Zu diesem Zweck öffnen wir unsere Code-Datei *CustomControl1.cs* und ersetzen *Control* durch *TextBox*. Zusätzlich ändern wir den Namen unseres *CustomControl1.cs* in *ExtendedTextBox* um. (Listing 4)

Im Anschluss dürfen wir natürlich nicht vergessen, den Control-Namen in der *MainWindow.xaml* unserer *TestApp* zu ändern, da ansonsten der Compiler einen Fehler meldet.

Nachdem wir alle notwendigen Eigenschaften und Funktionen von der WPF Standard-TextBox geerbt haben, müssen wir ebenfalls den Style und das *ControlTemplate* für unsere *ExtendedTextBox* festlegen. Die visuelle Repräsentation unserer *ExtendedTextBox* beschränkt sich momentan lediglich auf ein *Border*-Element. Da wir das Standard-Template der WPF Standard-TextBox nur um

Listing 3: **Main Window**

```
<Window x:Class="TestApp.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/
        markup-compatibility/2006"
        xmlns:local="clr-namespace:TestApp"
        xmlns:cclib="clr-namespace:CCLib;assembly=CCLib"
        mc:Ignorable="d"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
      <cclib:CustomControl1 Background="Black" BorderBrush="Red"
      BorderThickness="8"/>
    </Grid>
</Window>
```

Listing 4: **ExtendedTextBox**

```
public class ExtendedTextBox : TextBox
{
  static ExtendedTextBox()
  {
     DefaultStyleKeyProperty.OverrideMetadata(typeof(ExtendedTextBox), new
       FrameworkPropertyMetadata(typeof(ExtendedTextBox)));
  }
}
```

Listing 5: **Generic.xaml**

```
<ResourceDictionary
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:CCLib">
    <Style TargetType="{x:Type local:ExtendedTextBox}">
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="{x:Type local:ExtendedTextBox}">
                        <Border x:Name="border" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness=
                        "{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}"
                        SnapsToDevicePixels="True">
                        <ScrollViewer x:Name="PART_ContentHost" Focusable="False" HorizontalScrollBarVisibility=
                        "Hidden" VerticalScrollBarVisibility="Hidden"/>
                    </Border>
                    <ControlTemplate.Triggers>
                        <Trigger Property="IsEnabled" Value="False">
                            <Setter Property="Opacity" TargetName="border"
                            Value="0.56"/>
                        </Trigger>
                        <Trigger Property="IsMouseOver" Value="True">
                            <Setter Property="BorderBrush" TargetName="border"
                            Value="#FF7EB4EA"/>
                        </Trigger>
                        <Trigger Property="IsKeyboardFocused" Value="True">
                            <Setter Property="BorderBrush" TargetName="border"
                            Value="#FF569DE5"/>
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>
</ResourceDictionary>
```

einen Platzhalter-Text erweitern wollen, können wir dieses als Ausgangsbasis für unser eigenes Control nutzen. Um das Template der Standard-TextBox zu kopieren und für unser Control zu verwerten, können wir uns in unserer *TestApp* eine Standard-TextBox anlegen. Anschließend markieren wir diese und suchen in unserem *Properties-Window* im *Visual Studio* (standardmäßig rechts unten) die Überschrift *Miscellaneous*. Hier finden wir den Eintrag *Template*. Neben *Template* können wir nun auf das kleine weiße Rechteck klicken und den Unterpunkt *Convert* to *New Resource* wählen (siehe Abb. 6). Im Anschluss daran öffnet sich ein Dialog, in dem wir angeben können, wo wir unsere Kopie des Standard-Template der TextBox abspeichern möchten. Hier können wir einfach die Standardeinstellungen wählen (Ressourcen von *MainWindow*) und mit *OK* bestätigen (siehe Abb. 7).

Anschließend können wir das Template unseres Controls, welches in der Datei *Generic.xaml* abgelegt ist, durch das gerade erzeugte Template ersetzen. (Siehe Listing 5)

Das Attribut *x:Key* muss aus dem kopierten Template entfernt werden, da sonst der Compiler einen Fehler wirft. Zusätzlich sollte der *TargetType* auf unsere *ExtendedTextBox* gesetzt werden. Die angelegte TextBox sowie der komplette Ressourcen-Bereich in unserem *MainWindow* der *TestApp* können nun wieder entfernt werden. Wenn wir unsere Anwendung nun ausführen, so sehen wir, dass wir eine fast identische Kopie der WPF Standard-TextBox haben. Im Vergleich zur Standard-TextBox fehlen nun nur noch ein paar Style-Definitionen, wie beispielsweise das *Border*-Element. Um unser Control auch vom TextBox Standard-Style erben zu lassen, verwenden wir einfach die Eigenschaft *BasedOn* des *Styles* und setzen diese

Listing 6: **PlaceHolderText-Property**

```
public class ExtendedTextBox : TextBox
{
    public static readonly DependencyProperty PlaceHolderTextProperty;

    public string PlaceHolderText
    {
        get { return (string)GetValue(PlaceHolderTextProperty); }
        set { SetValue(PlaceHolderTextProperty, value); }
    }

    static ExtendedTextBox()
    {
       DefaultStyleKeyProperty.OverrideMetadata(typeof(ExtendedTextBox),
       new FrameworkPropertyMetadata(typeof(ExtendedTextBox)));

       PlaceHolderTextProperty = DependencyProperty.Register(„PlaceHolderText", typeof(string),
       typeof(ExtendedTextBox), new FrameworkPropertyMetadata(string.Empty));
    }
}
```

auf eine statische Ressource. Häufig werden bei Angabe der statischen Ressourcen die Keys der zu verwendeten Ressource angegeben. Da es sich bei unserer Ressource jedoch um den globalen Style der WPF-TextBox handelt, müssen wir diese ohne Key angeben. Zum Erben des globalen TextBox Styles nutzen wir folgende Angabe der statischen Ressource:

```
<Style TargetType="{x:Type local:ExtendedTextBox}"
BasedOn="{StaticResource {x:Type TextBox}}">
```

Nun stimmt auch der Style unserer *ExtendedTextBox* mit dem der Standard-TextBox überein. Im nächsten Abschnitt kümmern wir uns dann darum, unsere *ExtendedTextBox* um den Platzhalter-Text zu erweitern.

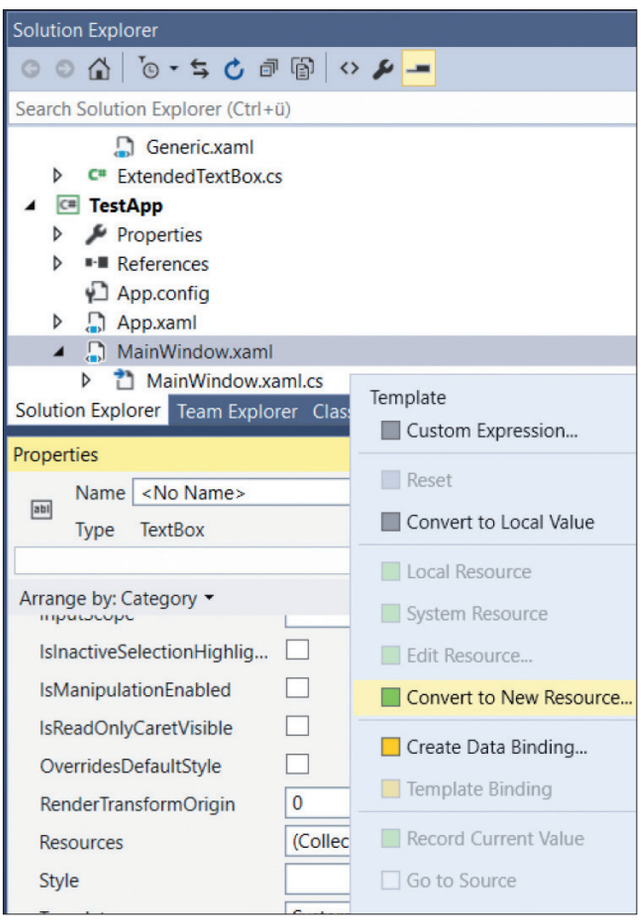


Abbildung 6: Menü um Kopie des TextBox-Template anlegen

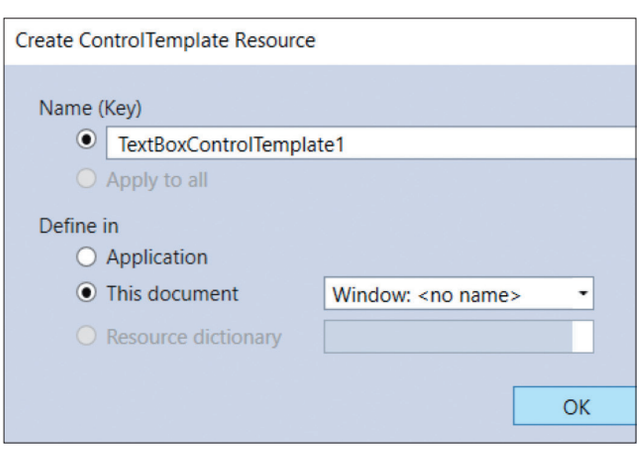


Abbildung 7: Dialog um TextBox-Template Kopie zu speichern

DIE PLATZHALTER-TEXT EIGENSCHAFT

Nachdem wir eine identische Kopie des Standard TextBox-Control erzeugt haben, können wir diese nun um unsere gewünschte Funktionalität erweitern. Was braucht unser Control eigentlich zusätzlich? Neben der gewöhnlichen Texteigenschaft benötigen wir eine zusätzliche Eigenschaft, in der der Platzhalter-Text gespeichert werden kann. Hierfür ergänzen wir unsere *ExtendedTextBox* um eine *DependencyProperty* mit dem Namen *PlaceHolderTextProperty*. (Listing 6)

Nun können wir bereits die Eigenschaft *PlaceHolderText* in unserer *ExtendedTextBox* unserer *TestApp* zuweisen. Angezeigt wird unser Platzhalter allerdings noch nicht. Hierum kümmern wir uns als nächstes.

Damit unser Platzhalter auch angezeigt werden kann, erweitern wir unser Template in der *Gerneric.xaml* um ein *TextBlock*-Element, welches sich um die Anzeige des Textes kümmern soll. Unseren *TextBlock* müssen wir nun auf unsere *PlaceHolderText*-Eigenschaft binden, damit unser zugewiesener Platzhalter-Text auch in der *View* des Controls angezeigt werden kann. Zusätzlich passen wir über den Style unseres *TextBlock* noch den *Foreground* und den *Margin* an. (Listing 7) Führen wir nun unsere *TestApp* aus, so können wir sehen, dass unser Platzhalter-Text bereits angezeigt wird. Im folgenden Abschnitt werden wir die Logik implementieren, damit sich unser Control auch wie gewünscht verhält. Momentan überlagern sich noch der Platzhalter-Text und der durch den Nutzer eingegebene Text.

IMPLEMENTIERUNG DER LOGIK

Nachdem wir im letzten Abschnitt unser *ExtendedTextBox*-Control mit einem Platzhalter ausgestattet haben, werden wir diesen nun mit der gewünschten Logik versehen. Diese soll wie folgt aussehen:

1. Wenn der Nutzer unser *ExtendedTextBox*-Control

Listing 7: **Platzhalter für Textanzeige**

```
<Border x:Name="border" BorderBrush="{TemplateBinding BorderBrush}"
BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}"
SnapsToDevicePixels="True">
    <Grid>
       <ScrollViewer x:Name="PART_ContentHost" Focusable="False" HorizontalScrollBarVisibility="Hidden"
       VerticalScrollBarVisibility="Hidden"/>
       <TextBlock x:Name="PART_PlaceHolderTextBlock" Text="{TemplateBinding PlaceHolderText}">
            <TextBlock.Style>
                <Style TargetType="{x:Type TextBlock}">
                    <Style.Setters>
                        <Setter Property="Foreground" Value="Gray"/>
                        <Setter Property="Margin" Value="2 0 0 0"/>
                    </Style.Setters>
                </Style>
             </TextBlock.Style>
         </TextBlock>
    </Grid>
</Border>
```

fokussiert, dann soll unser TextBlock zur Anzeige des Platzhalters unsichtbar werden

2. Wenn unsere *ExtendedTextBox* den Fokus verliert und der Nutzer keinen Text in die TextBox geschrieben hat, dann soll der Platzhalter-Text wieder sichtbar werden

Um unsere Logik umzusetzen, wollen wir, abhängig vom Fokus und dem enthaltenen Text, unsere Eigenschaft *Visibility* des *TextBlocks* zur Anzeige des Platzhalters neu setzen. Hierfür definieren wir uns im Code ein Feld vom Typ *TextBlock*, welches später eine Referenz auf das in der *Generic.xaml* definierte *TextBlock*-Element zur Modifikation der *Visibility*-Eigenschaft erhalten soll:

```
private TextBlock
placeHolderTextBlock;
```

Durch ein Überschreiben der Methoden *OnGotFocus* und *OnLostFocus* der Basisklasse *TextBox* können wir nun darauf reagieren, wenn der Nutzer unsere *ExtendedTextBox* fokussiert bzw. den Fokus wieder verlässt. Wir überschreiben diese Methoden und fügen unsere zuvor besprochene Logik für die Sichtbarkeit des Platzhalter-*TextBlock* hinzu (Listing 8)

Wenn wir unsere *TestApp* nun ausführen und unserer *ExtendedTextBox* den Fokus geben, so stellen wir fest, dass unsere Methoden *OnGotFocus* und *OnLostFocus* bereits ausgeführt werden und somit auf Nutzerinteraktionen reagieren. Jedoch kann die Sichtbarkeit unseres *TextBlocks* momentan noch nicht modifiziert werden, da unser Feld *placeHolderTextBlock* noch nicht das in der *Generic.xaml* definierte *TextBlock*-Element referenziert, dessen *Visibility*-Eigenschaft wir eigentlich ändern möchten.

Da unser *TextBlock* lediglich im Template definiert ist, können wir nicht direkt über dessen Namen darauf zugreifen. Im Rahmen der Custom Controls wird eine andere Technik verwendet um auf Elemente im Template zuzugreifen. Hierfür überschreiben wir die Methode *OnApplyTemplate* der Basisklasse *FrameworkElement*, von welcher wiederrum das Standard TextBox-Control der WPF erbt. (Listing 9)

Diese Methode wird kurz nachdem der Visual Tree des Custom-Control erzeugt wird aufgerufen. Nachdem der Visual Tree für unsere *ExtendedTextBox* zugewiesen wurde, können wir nun über *this.Template.FindName()* unser Element mit dem Namen *PART_PlaceHolderTextBlock* suchen und über unser Feld *placeHolderTextBlock* referenzieren. Der Präfix PART ist hierbei eine Namenskonvention für alle Elemente, welche für die Logik des Custom Controls benötigt werden und bei einem neuen Template unbedingt wieder vorhanden sein müssen, da diese mit der Logik des Steuerelementes verknüpft sind. Ansonsten würde die Logik des Controls nicht mehr funktionieren. Um dies für Entwickler und Designer kenntlich zu machen, werden alle Template-Parts über Klassen-Attribute des Custom-Controls kenntlich gemacht. In unserem Fall müssen wir

Listing 8: **Focus-Handling**

```
protected override void OnGotFocus(RoutedEventArgs e)
{
     base.OnGotFocus(e);
     this.placeHolderTextBlock.Visibility = Visibility.Collapsed;
}

protected override void OnLostFocus(RoutedEventArgs e)
{
     base.OnLostFocus(e);
     if (string.IsNullOrEmpty(this.Text) && placeHolderTextBlock != null)
     {
          this.placeHolderTextBlock.Visibility = Visibility.Visible;
     }
}
```

Listing 9: **Methode OnApplyTemplate**

```
public override void OnApplyTemplate()
{
     base.OnApplyTemplate();
     if (this.Template != null)
     {
          this.placeHolderTextBlock =
          this.Template.FindName(„PART_PlaceHolderTextBlock", this) as
          TextBlock;
     }
}
```

unserer Klasse *ExtendedTextBox* noch ein Attribut für unseren *PART_PlaceHolderTextBlock* hinzufügen. Im Attribut *TemplatePart* wird der Name des Teils sowie dessen Typ angegeben. Die finale Klasse des *ExtendedTextBox*-Control ist in Listing 10 zu finden.

Somit ist unser *ExtendeddTextBox*-Control voll funktionsfähig. Darüber hinaus können wir in den *TestApp*-Ressourcen das Standard *ControlTemplate* unseres Controls testweise neu definieren. Hierbei wird die Flexibilität eines Custom-Control ersichtlich. Des Weiteren ließe sich problemlos von unserer *ExtendedTextBox* ableiten, um diese um weitere Funktionalität und Eigenschaften zu ergänzen. Der komplette Quellcode für das *ExtendedTextBox*-Control ist unter *https://github.com/d-wolf/ExtendedTextBoxWPF* verfügbar.

DANIEL WOLF hat Medieninformatik an der HTW Dresden studiert, ist leidenschaftlicher .NET-Entwickler und als Development-Trainer bei der ppedv tätig. Seine Schwerpunkte liegen verstärkt in den Bereichen Softwareentwicklung und Architektur, mit dem technologischen Fokus auf Universal Windows Platform Apps sowie Anwendungen auf Basis der WPF und XAML.

Listing 10: **die gesamte Klasse**

```
[TemplatePart(Name = „PART_PlaceHolderTextBlock“, Type = typeof(TextBlock))]
public class ExtendedTextBox : TextBox
{
    public static readonly DependencyProperty PlaceHolderTextProperty;

    private TextBlock placeHolderTextBlock;

    public string PlaceHolderText
    {
        get { return (string)GetValue(PlaceHolderTextProperty); }
        set { SetValue(PlaceHolderTextProperty, value); }
    }

    static ExtendedTextBox()
    {
       DefaultStyleKeyProperty.OverrideMetadata(typeof(ExtendedTextBox), new FrameworkPropertyMetadata
       (typeof(ExtendedTextBox)));

       PlaceHolderTextProperty = DependencyProperty.Register(„PlaceHolderText“, typeof(string),
       typeof(ExtendedTextBox),
       new FrameworkPropertyMetadata(string.Empty));
    }

    public override void OnApplyTemplate()
    {
        base.OnApplyTemplate();

        if (this.Template != null)
        {
            this.placeHolderTextBlock = this.Template.FindName(„PART_PlaceHolderTextBlock“, this) as TextBlock;
        }
    }

    protected override void OnGotFocus(RoutedEventArgs e)
    {
        base.OnGotFocus(e);
        this.placeHolderTextBlock.Visibility = Visibility.Collapsed;
    }

    protected override void OnLostFocus(RoutedEventArgs e)
    {
        base.OnLostFocus(e);

        if (this.Text.Equals(string.Empty))
        {
            this.placeHolderTextBlock.Visibility = Visibility.Visible;
        }
        else
        {
            this.placeHolderTextBlock.Visibility = Visibility.Collapsed;
        }
    }
}
```

Microsoft und R

neue Möglichkeiten zur Datenanalyse

Als ich vom VisualStudio1 Team gefragt wurde, ob ich einen Artikel über R zur Verfügung stellen könnte, war ich sofort begeistert. Es ist mir eine Ehre an dieser Stelle mein neuestes Steckenpferd etwas vorzustellen. Ich hoffe ich kann euch heute auch dafür begeistern, einmal etwas abseits von C#, oder JavaScript zu wildern und möglicherweise eine – erst mal - kleine Lösung mit R in eurem Business-Umfeld umzusetzen. Halten wir uns nicht lange auf, Microsoft hat in den letzten Monaten ordentlich Fahrt aufgenommen und uns – der Entwicklergemeinde - schon einiges vorgelegt. Es ist also an der Zeit diese Möglichkeiten genauer zu betrachten und diese auch zu nutzen.

WAS IST R UND BRAUCHE ICH DAS?

R ist ei ne Programmiersprache mit freier (und kostenloser) Umgebung (inklusive Entwicklungsumgebung) zur statistischen Datenverarbeitung. R ist eine Skriptsprache und integriert eine Vielzahl von Befehlen, bzw. Funktionen, um Daten zu organisieren, zu transformieren, auszuwerten und visualisieren zu können. Natürlich können wir das als Entwickler auch ohne R Lösungen für komplexe Probleme finden, aber uns fehlt normalerweise das nötige Wissen, wie Daten zu analysieren sind. Wir haben kein Problem, die Daten zu finden, zu transformieren, aber wirklich zu analysieren? Beispielsweise wusste ich vor meiner Zeit mit R nicht, was Benford's Law ist, wie ich Naive Bayes zur Klassifizierung, oder K-Means zum Clustern von Daten einsetze. Der große Vorteil bei R ist, dass durch die Community eine Vielzahl von Paketen – frei verfügbar – vorhanden sind, die sofort in die eigene Analyse eingebunden werden können. Eine weitere große Stärke von R ist die recht einfache grafische Darstellung der Daten, bzw. der Ergebnisse von Datenanalysen. Zusammenfassend lässt sich sagen, dass R sich hervorragend zur statistischen Analyse von Daten eignet und man mit wenigen Zeilen R beeindruckende Ergebnisse erzielen kann.

Dazu aber gerne einmal in einem anderen Artikel mehr, dieses Mal konzentriere ich mich auf den Einstieg mit R und werde daher nur ein recht einfaches verwenden.

R OHNE MICROSOFT, GEHT DAS AUCH?

Ja, natürlich. Problemlos - und das macht zu Beginn auch sehr viel Spaß. Microsoft hat's nun einmal nicht erfunden, ist aber sehr gut mit dabei, wie ihr gleich etwas später sehen werdet. Um R auf einem Windows Rechner zum Laufen zu bekommen sind zwei Schritte notwendig:

- Installation von R von *https://cran.r-project.org/*
  - 32- und 64-bit Versionen sind hier vorhanden
- Installation von R-Studio von *https://www.rstudio.com/products/rstudio/download/*
  - Open Source (AGPL v3 Lizenz), bzw. kommerzielle Lizenzen ($995/Jahr) verfügbar

Nach der Installation sind zwei neue Icons verfügbar (siehe Abbildung 1: Icons). Ersteres ist das R-Studio, letzteres ist das Command-Line Tool.
Mit RStudio kann schon entwickelt werden, aber ist das auch das Richtige für uns? Ich glaube, ich habe da noch etwas Besseres für euch.

SCHNELLEINSTIEG IN R

Da ihr ja die aktuelle VisualStudio1 in Händen haltet, bietet es sich natürlich an, R mit Visual Studio zu verwenden, dazu gibt es R Tools für Visual Studio, kurz RTVS – zur Installation in Visual Studio 2015 mit (zumindest) Update1 auf eurem 64bit Rechner, schaut doch gleich einmal hier vorbei: *http://microsoft.github.io/RTVS-docs/installation.html.*

Nach der Installation steht das „R Project"-Template zur Verfügung. (Abbildung 2). Es kann also sofort mit dem eigenen R Projekt losgehen und da bietet es sich doch gleich einmal an, in die Sprache R ein klein wenig hinein zu schnuppern. Nehmen wir ein recht simples, abgeschlossenes Beispiel zur Hand. Dieses zeigt nicht mal ansatzmäßig wie mächtig R ist, aber als Einstieg – denke ich zumindest – recht leicht nachvollziehbar. Das Szenario sieht wie folgt aus: Es gibt 5 Tage an denen Roulette gespielt wurde, mal mit Gewinn (Mittwoch, Freitag), mal mit Verlust (Montag, Dienstag, Donnerstag). Ich würde nun gerne von R folgendes berechnen lassen:

- Anzahl der Tage mit Gewinn (wo wir im Plus waren)
- Die Summe der Tage, wo Gewinn gemacht wurde
- Die Summe über die ganze Woche.

Die Umsetzung mit einem R-Skript

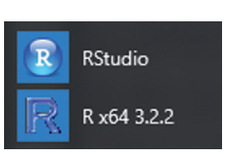


Abbildung 1: Icons

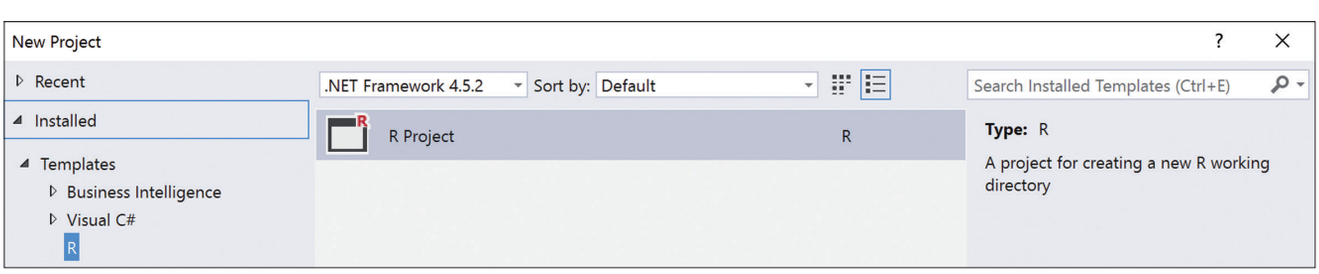


Abbildung 2: Template

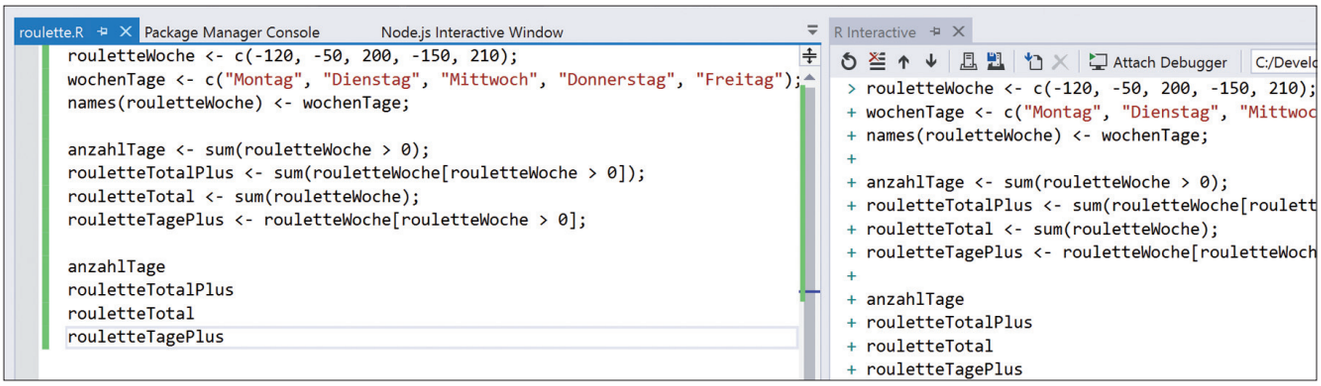


Abbildung 3: RTVS mit R Interactive Console

wird in Listing 1 gezeigt.
Also nichts wie hinein mit diesem Code in ein .R-File in Visual Studio, alles markieren und mit CTRL+Return mit der R Interactive Console ausprobieren. Das sieht dann genau so aus, wie in Abbildung 3: RTVS mit R Interactive Console.

In diesem Script wurden 2 Vektoren definiert, „rouletteWoche“ und „wochenTage“. Diese beiden Vektoren wurden verbunden, um der „rouletteWoche“ die Tage zuzuordnen. Danach wurde die Anzahl der Tage mit Gewinn berechnet („anzahlTage“), sowie die Summe der Gewinne („rouletteTotalPlus“), die Summe des Wochentotals („rouletteTotal“) und die Tage mit Gewinn („rouletteTagePlus“). Somit ist bekannt, wann es wohl am besten ist, Roulette zu spielen (Mittwoch und Freitag).

Das Tolle an RTVS ist die komplette Integration in Visual Studio. Das beginnt bei den Templates, geht über die Intellisense und die Ausführung von R Skripten in der R Interactive Console. Es können übrigens auch R Charts dargestellt werden, dafür gibt es wiederum ein eigenes Fenster (R Plot). Damit ist es auch möglich R-Grafiken als JPG oder PDF aus Visual Studio heraus zu exportieren. Das klappt mit dem obengenannten Beispiel mit einer weiteren Zeile Code:

```
barplot(rouletteWoche, col = ifelse(rouletteWoche < 0,
„red“, „green“))
```

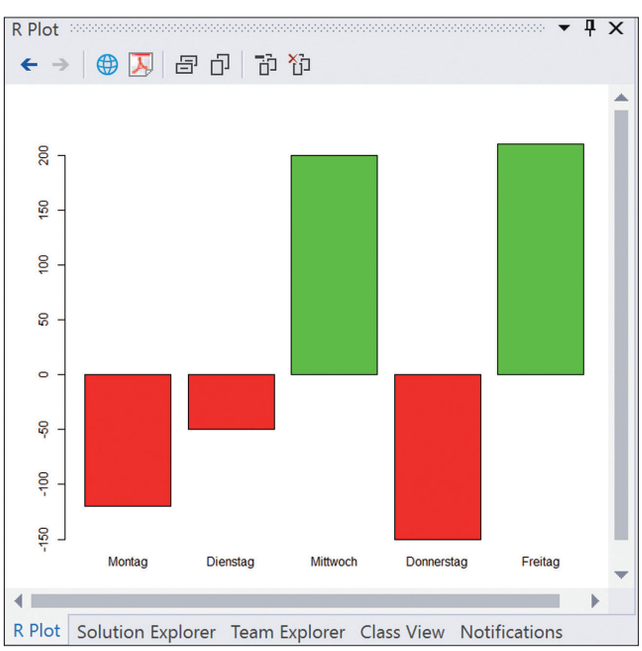


Abbildung 4: R Plot in Visual Studio

Das Ergebnis ist als Säulenchart in Abbildung 4: R Plot in Visual Studio

Diese Kombination der Tools reicht eigentlich auch schon, um gleich einmal ein wenig mit R herumzuspielen und erste kleine und etwas später auch größere Schritte zu gehen, bzw. größere Analysen zu definieren. Für komplexere Lösungen ist dieser Weg natürlich auch produktiv einsetzbar, aber warum sollte man nicht gleich ein größeres Bild definieren. Ja warum eigentlich?

R IM OPERATIONALEN UMFELD

Wie erwähnt, zum Ausprobieren reicht das natürlich schon einmal, möchte man R nun mit größeren Datenmengen und produktiv verwenden, steht man relativ schnell vor einigen - eigentlich üblichen - Problemen, sobald mehr und größere Datenmengen im Spiel sind. Nicht immer stehen die Daten genauso zur Verfügung, um wie gewünscht analysiert, bzw. bearbeitet werden zu können. Wächst auch noch die Datenmenge wird das Ganze schon immer unhandlicher und, nicht zu vergessen, der Sicherheitsaspekt, da die Daten zum Client transferiert werden müssen, um mit dem R-Client verarbeitet werden zu können.

R-Clients sind auch nicht gerade die Schnellsten, wenn es um die Datenverarbeitung geht, da diese bisher alle Single-Threaded arbeiten und das komplette Dataset im Speicher vorgehalten wird. Es stellt sich also eine Reihe von Fragen, einige davon sind in Tabelle 1: R in Großprojekten noch einmal etwas übersichtlicher dargestellt.

Gerade deshalb könnte eine Lösung mit Microsoft Tools uns in zukünftigen R Projekten hilfreich sein und uns einige der Probleme direkt, oder indirekt abnehmen. Was meine ich mit indirekt? Durch eine Unterstützung in

Listing 1: **R-Skript**

```
rouletteWoche <- c(-120, -50, 200, -150, 210);
wochenTage <- c(„Montag“, „Dienstag“, „Mittwoch“, „Donnerstag“,
„Freitag“);
names(rouletteWoche) <- wochenTage;

anzahlTage <- sum(rouletteWoche > 0);
rouletteTotalPlus <- sum(rouletteWoche[rouletteWoche > 0]);
rouletteTotal <- sum(rouletteWoche);
rouletteTagePlus <- rouletteWoche[rouletteWoche > 0];

anzahlTage
rouletteTotalPlus
rouletteTotal
rouletteTagePlus
```

Listing 2: **Konfiguration SQL Server**

```
Exec sp_configure ‚external
scripts enabled‘, 1;
reconfigure;

Exec sp_configure
```

Listing 3: **Konfigurationsskript**

```
“%programfiles%\RRO\RRO-3.2.2-
for-RRE-7.5.0\R-3.2.2\library\
RevoScaleR\rxLibs\x64\
RegisterRExt” /install
/instance:<INSTANCENAME>
```

Listing 4: **Anlegen der Tabelle und Dateneingabe**

```
CREATE TABLE [dbo].[Roulette](
       [Won] [int] NOT NULL,
) ON [PRIMARY]
Go

insert into Roulette
values (-120), (-50), (200), (-150),
(210)
go
```

die bestehende Tools-Palette finden wir natürlich sofort einen Benefit, da Microsoft schon eine sehr lange bestehende, breite Tool-Landschaft hat. Als Beispiele sind hier nur kurz Integration Services für die Transformation, SQL Server für das Vorhalten der Daten, PowerBI für das Reporting, Azure zur Verbesserung der Skalierbarkeit aufgeführt.

R IM MICROSOFT UMFELD

Microsoft hat im April 2015 die Firma Revolution R übernommen und einen Teil, der aus dieser Übernahme entstanden ist haben wir schon gesehen, nämlich die R Tools für Visual Studio (RTVS). Da gibt es aber in der Zwischenzeit mehr und diese neuen Tools, bzw. Erweiterungen an den bestehenden Tools und Lösungen im Microsoft Umfeld helfen auch die Probleme für größere und komplexere Lösungen zu lösen (Siehe Abbildung 5). Generell kann man sagen, dass die Community Lösung für kleinere Projekte durchaus ausreichend ist, für Enterprise taugliche Lösungen sind die kommerziellen Tools vorhanden.
Sehen wir uns einen Ausschnitt im Schnelldurchgang nun nachfolgend an.

MICROSOFT R OPEN

Microsoft R Open ist eine erweiterte R Distribution. Da R unter der GPL

Bewegen größerer Datenmengen	• Verschiedene Datenquellen • Daten müssen aufbereitet werden (ETL) • Sicherheit der Daten • Wachstum der Daten mit der Zeit
Operativer Einsatz	• Wie bringe im meine Lösung in Produktion? • Wie starte ich wiederkehrende Analysen automatisch? • Wo lege ich Ergebnisse ab? • Wie sehe ich die Ergebnisse?
Operativer Einsatz	• R Tools Single Threaded • DataSet wird im Speicher abgearbeitet • Komplexe Analysen dauern dementsprechend lange.

Tabelle 1: R in Großprojekten

Open Source ist, gilt das genauso für Microsoft R Open. Dies bedeutet auch, dass alle Packages, die verfügbar sind, damit laufen, es ist also zu 100 kompatibel. Zusätzlich wurde es mit eigenen Paketen angereichert, wie z.B. einer Mathematikbibliothek, die von Haus aus multithreaded ist. Des Weiteren verwendet es ein eigenes Repository der frei verfügbaren Pakete, um die Versionssicherheit ebendieser zu gewährleisten. Das bedeutet, dass in einem Team Skripte, die auf 3rd Party Packages aufbauen problemlos getauscht werden können, es also keine Probleme bezüglich verschiedener Versionen dieser geben kann.

Den Download von Microsoft R Open findet man hier: *https://mran.revolutionanalytics.com/download*

R UNTERSTÜTZUNG IM SQL SERVER

Um R im SQL Server 2016 nutzen zu können, benötigt man neben der relationalen Engine noch weitere Softwarepakte auf der Maschine, auf der der SQL Server läuft:

- Während der Installation von SQL Server 2016, das Feature „Advanced Analytics Extension“ anwählen – bzw. eine nachträgliche Installation mittels dem SQL Server Setup.
- Installation von Microsoft R Open 3.2.2 for Revolution R Enterprise.
- Installation des Revolution R Enterprise 7.5 Revolution R Enterprise Packages für den SQL Server.

Beides demnächst in der Version 8.0 und als Microsoft R Server verfügbar, in der derzeitigen Version des SQL Server 2016 allerdings noch mit den Vorgängerversionen (was sich spätestens mit der Release des SQL Servers ändern wird).

Die Integration funktioniert über Libraries, die aus dem SQL Server heraus in die .NET Runtime springen und Resultate wieder an den SQL Server zurückreichen. Außerdem gibt es einen neuen Windows Dienst, der gestartet sein muss, das SQL Server Launchpad (MSSQLLaunchpad) Service. Der Ablauf ist in Abbildung 6 dargestellt.

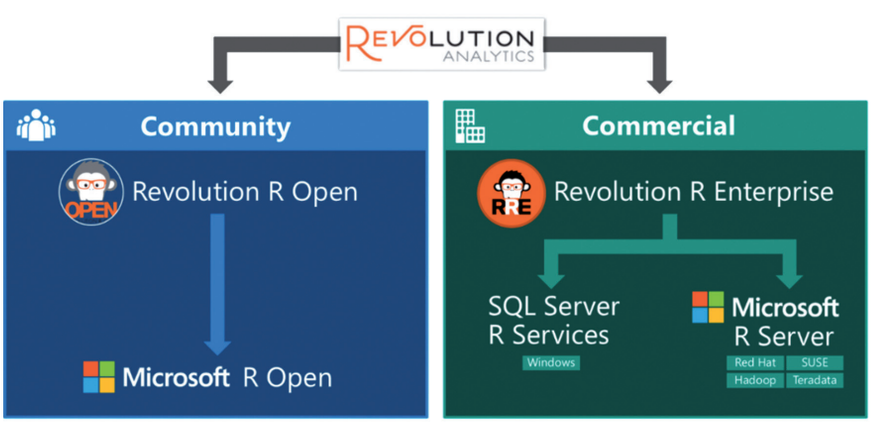


Abbildung 5: Revolution R wurde zu Microsoft R

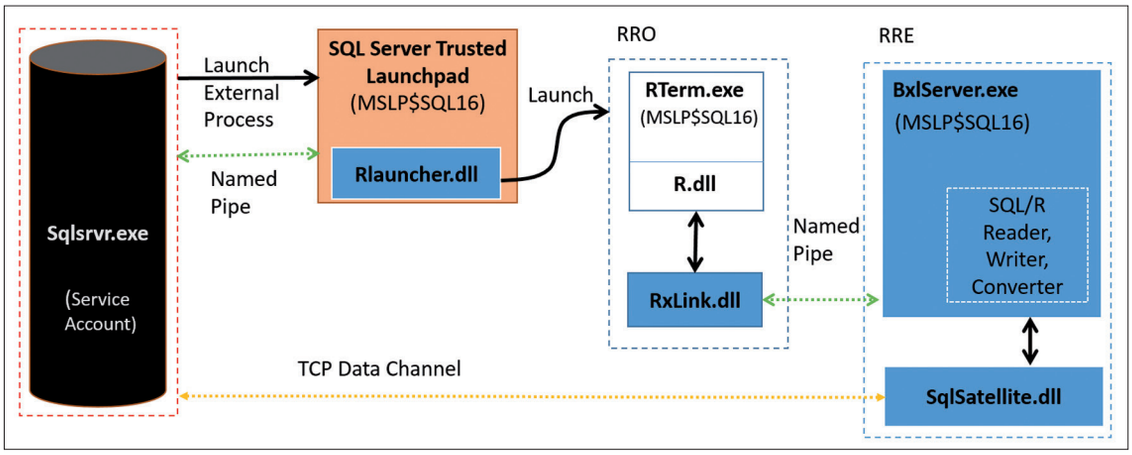


Abbildung 6: Architektur der R Integration im SQL 2016

Nach der Installation müssen abschließend noch zwei weitere Schritte erledigt werden. Der SQL Server muss konfiguriert werden, damit er externe Scripte überhaupt einmal zulässt, dies erledigt man im SSMS mit den Kommandos in Listing 2.

Abschließend ist ein Konfigurationsskript auszuführen, welches mit der Installation des Revolution R Packages mitgeliefert wurde. Dieses muss aus einer CMD-Shell mit Administratoren-Berechtigungen aufgerufen werden. (Listing 3)

Nach dem Neustart des SQL Servers wurden damit alle notwendigen Schritte erledigt und die R-Integration im SQL Server 2016 kann getestet werden.

R SKRIPTE IM SQL SERVER

Jetzt sollte allerdings alles bereit sein und ich möchte nun versuchen, die neue Stored Procedure sp_execute_external_script mit einem einfachen R Skript aus dem SQL Server Management Studio (SSMS) heraus auszuführen. Der erste Parameter @language gibt an, in welcher Sprache das Skript ist, welches im zweiten Parameter (@script) als Unicode String mitgegeben wird. Der dritte Parameter dient dazu, Daten in das R Skript zu übergeben, was wir auch etwas später testen wollen.

Results | Messages

	anzahlTage	rouletteTotalPlus	rouletteTotal
1	2	410	90

Abbildung 7: Ergebnis

Results | Messages

	day	rouletteTagePlus
1	Mittwoch	200
2	Freitag	210

Abbildung 8: Ergebnis Gewinne

Nehmen wir doch gleich einmal das zuvor verwendete Roulettebeispiel her und bauen das gleich einmal in den SQL Server ein, die Daten sollen allerdings aus einer Tabelle am SQL Server kommen.

Die Daten sind einfach die 5 Werte von zuvor, das Beispiel kann auch aufgebohrt werden, um den Tag und die Woche als eigene Spalten zu halten, dies würde aber nur zu einem komplexeren SELECT-Statement führen, was ich aus Gründen der leichten Nachvollziehbarkeit nicht getan habe. (Listing 4)

Jetzt sind die Daten also im SQL Server vorhanden, das R-Skript muss nun insofern erweitert werden, als dass ein Select-Statement (Listing 5) für die Datenübergabe verwendet wird. Die Eingangsdaten wurden in eine Matrix umgewandelt, danach kann mit den Daten wie zuvor gerechnet werden, auch das Resultat ist wie zuletzt und in Abbildung 7.

Abschließend würde ich gerne die beiden Tage sehen, wo Gewinn gemacht wurde und zwar den Wochentag und den Gewinn. (Listing 6) Das trickreiche hier ist das Binden an die Variable „rouletteTagePlus“, um die Spaltennamen ausgeben zu können. Das Ergebnis ist wie gewünscht, es ist der Name des Wochentags zu

Listing 5: **SQL Abfrage**

```
execute sp_execute_external_script
  @language = N'R'
, @script = N'
               rouletteWoche <- as.matrix(InputDataSet);

               anzahlTage <- sum(rouletteWoche > 0);
               rouletteTotalPlus <- sum(rouletteWoche[rouletteWoche > 0]);
               rouletteTotal <- sum(rouletteWoche);

        OutputDataSet <- data.frame(anzahlTage, rouletteTotalPlus,
        rouletteTotal);'
, @input_data_1 = N' Select Won from Roulette'
WITH RESULT SETS (([anzahlTage] int, [rouletteTotalPlus] int,
[rouletteTotal] int));
```

Listing 6: **SQL Abfrage Gewinne**

```
execute sp_execute_external_script
  @language = N'R'
, @script = N'
               rouletteWoche <- as.matrix(InputDataSet);
               wochenTage <- c(„Montag“, „Dienstag“, „Mittwoch“,
               „Donnerstag“, „Freitag“);
               names(rouletteWoche) <- wochenTage;

               rouletteTagePlus <- rouletteWoche[rouletteWoche > 0];

        colNames <- cbind(rouletteTagePlus);
        #colNames <- data.frame(rouletteTagePlus, check.names = FALSE);

        OutputDataSet <- data.frame(rownames(colNames),
        rouletteTagePlus);'
, @input_data_1 = N' Select Won from Roulette'
WITH RESULT SETS (([day] nvarchar(20), [rouletteTagePlus] int));
```

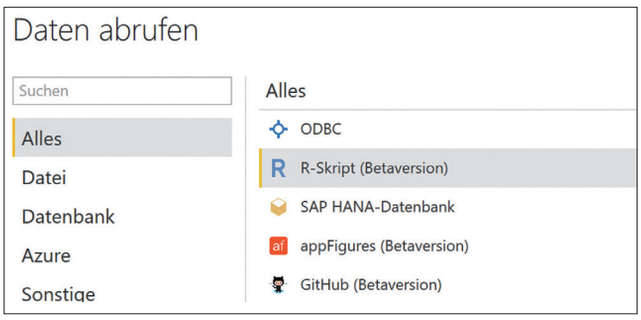


Abbildung 9: R Skript als Datenquelle

sehen und die zugehörigen Werte (Siehe Abbildung 8) Die Integration in den SQL Server ist schon eine sehr schöne Möglichkeit mit R zu arbeiten. Wie sieht es aber nun mit dem Reporting aus? Hier kommt jetzt PowerBI ins Spiel. Auch PowerBI kann mit R umgehen.

POWERBI UND R

Um R in PowerBI zu verwenden, gibt es zwei Möglichkeiten. Einmal liefert ein R Skript die Daten, dann dient ebendieses als Datenquelle, wie in Abbildung 9: R Skript als DatenquelleFehler! Verweisquelle konnte nicht gefunden werden. zu sehen.

Die andere Variante ist, dass R für die Grafik selbst verantwortlich ist. R-Grafiken sind in PowerBI-Desktop derzeit noch ein Vorschaufeature, das heißt, um es zu verwenden, muss dieses derzeit noch extra in den Optionen aktiviert werden (Abbildung 10)

Danach kann es aber schon verwendet werden und ich habe gleich einmal passende Roulette-Daten in Power BI eingetragen (siehe Abbildung 11)

Nun ist es noch ein kleiner Schritt zum R-Plot in Power BI. Es muss noch die Grafik angewählt werden, in diesem Fall ist es das R

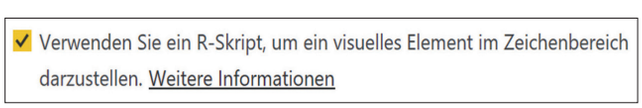


Abbildung 10: Optionen

Icon in der letzten Zeile, die gewünschten Felder können nun in den Chartbereich gezogen werden (Gewinn und Tag) daraufhin erscheint ein R-Skript-Editor (Abbildung 12) und mit einer weiteren Zeile Code (plot(dataset$Tag, dataset$Gewinn)) wird der Plot auch schon dargestellt.

R als Datenlieferant, oder als Werkzeug, um die Daten zu visualisieren. Beides in PowerBI von Microsoft bereits realisiert. Schön langsam lässt sich hier ein umfangreiches Bild erstellen, wo und wie R eingesetzt werden kann. Vielleicht möchte man nun auch Partnern solche Lösungen zur Verfügung stellen. Damit diese eine erstellte Analyse mit eigenen Daten durchführen können? Dann ist möglicherweise der nächste Punkt von Interesse.

AZURE ML UND R

Mit Azure ML hat Microsoft eine faszinierende Lösung für maschinelles Lernen zur Verfügung gestellt. Zuerst wird eine Azure ML Lösung gebaut, d.h. Eingangsdaten werden verwendet, diese passend transformiert – hierfür stehen eine Menge von Tasks zur Verfügung – und danach (auch) mittels R analysiert. (Abbildung 13)

Dabei wird üblicherweise erst nur ein Teil der Eingangsdaten definiert, um das Modell zu trainieren, danach wird der Rest der Daten als Test-Dataset verwendet, um die Resultate der Analyse überprüfen zu können. Ist eine zufriedenstellende Lösung gefunden, dann kann aus dem erstellten Modell ein WebService zur Verfügung gestellt werden, mit dem andere, bzw. neue Eingangsdaten hochgeladen werden können. Das Modell verarbeitet die Daten und liefert entsprechende Resultate der Datenanalyse zurück.

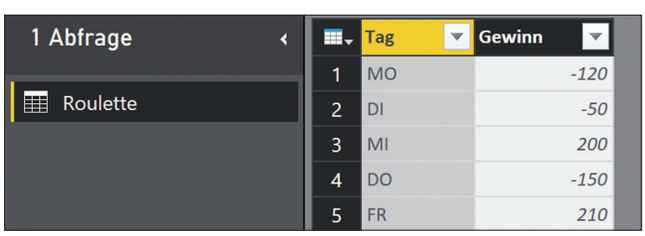


Abbildung 11: Daten für den späteren R-Plot

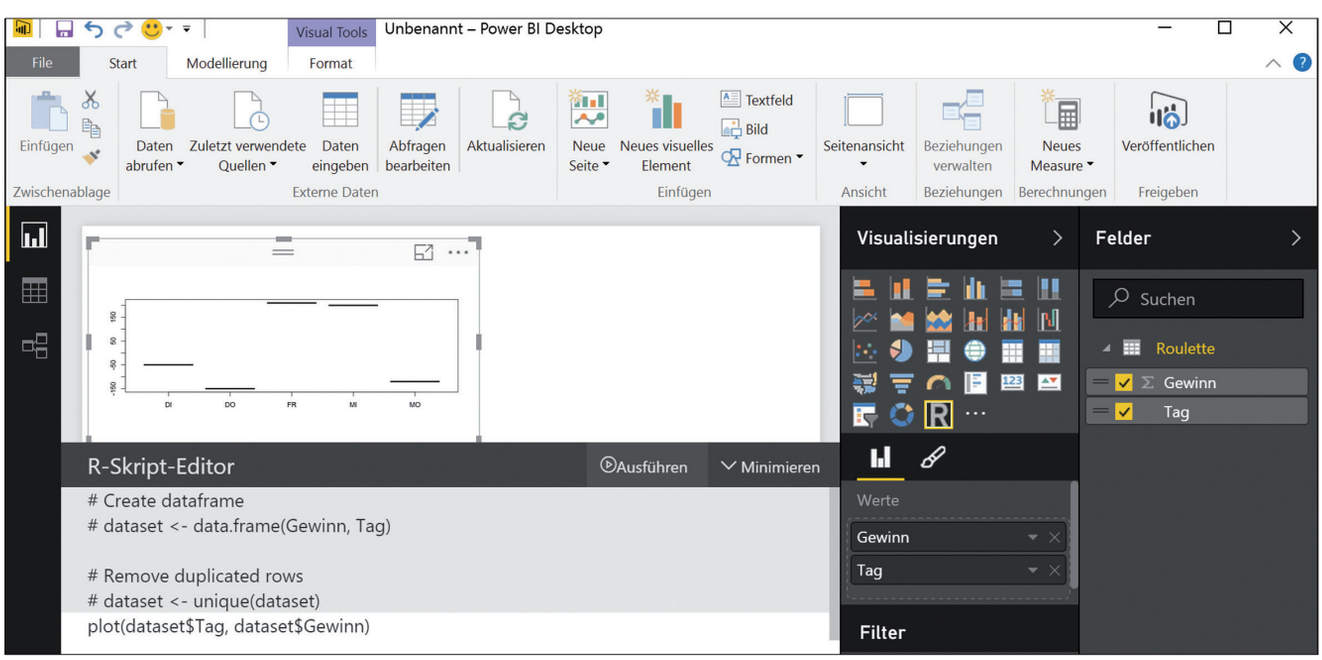


Abbildung 12: R Plot in PowerBI

Ein komplettes Beispiel - auch mit R zur Analyse der Daten - ist hier zu finden: *https://azure.microsoft.com/en-us/documentation/articles-machine-learning-r-quickstart/*

Sollen die Datenanalysen vielleicht doch lieber On-Premise laufen, dann empfiehlt sich möglicherweise eher nachfolgende Lösung.

MIROSOFT R SERVER

Im Jänner 2016 wurden die zugehörige Software Revolution R Enterprise in Microsoft R Server umbenannt und als Microsoft R Server 2016 (Version 8.0) released. Als MSDN Abonnent findet man die Software im MSDN-Portal zum Download. Falls man nur schnell einmal den Microsoft R Server ausprobieren möchte, stehen in Azure fix und fertige VMs zur Verfügung. Einfach eine neue VM erstellen und nach „R Server“ suchen, es finden sich dann die beiden Microsoft R Server VMs und zwar unter Linux (Microsoft R Server 8.0 on OpenLogic 6.5), als auch mit Windows (RRE für Windows 8.0 on Win2012 DC R2) – hier wurde das Rebranding von Revolution R Enterprise noch nicht vollzogen. (Siehe Abbildung 14)

Der Vorteil des Microsoft R Server liegt in der Nutzung von mehreren Cores während der Ausführung von R Skripten. Des Weiteren werden Microsoft-eigene R-Packages zur Verfügung gestellt, die ebenfalls parallelisiert arbeiten. Klarerweise gibt es auch verschiedene APIs, um unterschiedlichste Clients, aber auch Development Tools zu unterstützen. Also wenn es um eine skalierbare R-Lösung zur Datenanalyse geht, dann ist der Microsoft R Server definitiv ein Stück Software, das man sich genauer ansehen sollte.

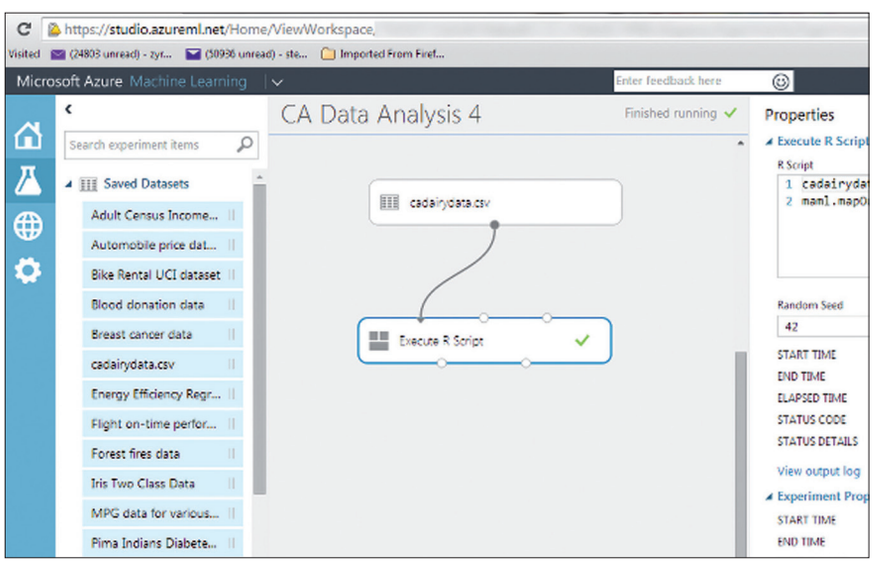


Abbildung 13: Azure ML Modell mit R

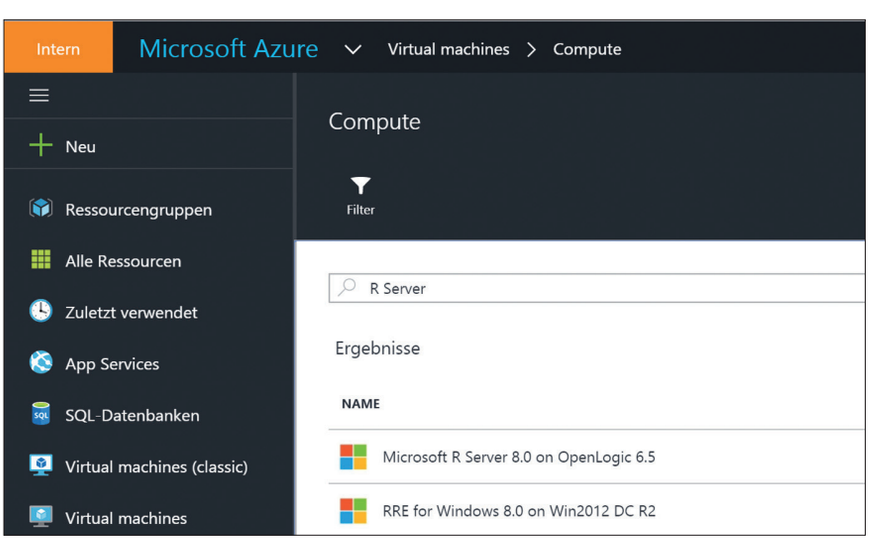


Abbildung 14: Microsoft R Server unter Azure

ZUSAMMENFASSUNG

R ist eine Skriptsprache mit Unterstützung einer sehr großen Community. Microsoft hat R in kurzer Zeit in eine Vielzahl der eigenen Produkte integriert und kann sofort eingesetzt werden:

- Unter Windows, quasi „standalone“, mit Microsoft R Open als IDE.
- In Visual Studio 2015 mit RTVS im Zusammenspiel mit einer beliebigen R-IDE, oder Microsoft R Open.
- Mit SQL Server 2016 (seit der CTP3) zur Verarbeitung und Analyse der Daten.
- Berichte mit PowerBI und R, wobei R entweder als Datenquelle, oder als Grafiklieferant dient.
- Unter Azure mit Azure Machine Learning und zugehörigem WebService.
- Mit dem Microsoft R Server für hoch performante Lösungen mit großen Datenmengen.

Ich denke damit sollte für jeden Geschmack etwas dabei sein. Falls man nur einmal schnuppern möchte, dann spielt man ein wenig mit Microsoft R Open, als Entwickler schätzt man die Integration in Visual Studio. Durch die Unterstützung in Azure ML und mit dem Microsoft R Server sind auch Enterprise Lösungen mit hohen Datenmengen mit hoher Performance zu verwirklichen.

BERNDT HAMBÖCK
beschäftigt sich seit Jahren mit Microsoft Technologien. Seine Schwerpunkt liegen im Development-Bereich, bei Azure und Windows IoT, sowie der mobilen Entwicklung mit Xamarin. Er ist seit 2007 MCT, führte drei Entwicklungsteams und berät Kunden bei der Implementierung von Softwarelösungen. Des Weiteren ist er fasziniert von den SQL Server Business Intelligence Server und Tools, und dem Bereich Big Data mit Machine Learning. Die Spiele-Entwicklung mit Unity und Visual Studio ist da ein spannendes Freizeitvergnügen, für das Ihm allerdings selten Zeit bleibt.

Workshop – Mobile Applikationen mit Android Studio entwickeln

GESCHICHTE

In unserer heutigen digitalen Welt gibt es viele kleine praktische Helfer, die uns allen das Leben ungemein erleichtern. Die Rede ist von den Smartphones, die in den letzten fünf bis zehn Jahren den gesamten Mobilfunkmarkt umgekrempelt haben. Anstelle von kleinen klobigen Mobiltelefonen, die bis auf die Sprachübertragung und das Schreiben von Kurznachrichten recht wenige zusätzliche Funktionen hatten, werden heutzutage vermehrt große und vor allem leistungsstarke Smartphones verwendet.

Anno dazumal war es noch im Trend, sehr kleine und kompakte Mobiltelefone zu besitzen, die einem möglichst wenig Platz in der Hosentasche wegnehmen. Doch mit den Smartphones erleben wir nun eine gegensätzliche Entwicklung: Bildschirme, die von Jahr zu Jahr größer werden.

Dies liegt insbesondere an der grundlegenden Änderung der Benutzereingabe: Anstelle von analogen Tasten werden bei modernen Smartphones berührungsempfindliche Bildschirme, sogenannte Touchscreens, verwendet. Auch die Menge an Informationen, die auf den nun weitaus größeren Bildschirmen dargestellt werden kann, wächst stetig.

Ein Detail, welches vor 10 Jahren größtenteils irrelevant war, aber in letzter Zeit massiv an Bedeutung gewonnen hat, ist das Betriebssystem des Mobiltelefons. Der Markt wird heutzutage von 3 Betriebssystemen beherrscht: Android, iOS und Windows Phone. In diesem Artikel werden wir uns ein wenig mit dem Android-Betriebssystem beschäftigen.

ANDROID

Android ist ein quelloffenes Betriebssystem, welches von der von Google gegründeten Open Handset Alliance entwickelt wird. Android selbst basiert auf einem monolithischen Linux-Kernel, der alle Treiber und Grundfunktionalitäten des Betriebssystems zusammenfasst. Obwohl Android ursprünglich für Digitalkameras entwickelt wurde, gewann das Betriebssystem recht rasch an Popularität und besitzt heute einen nicht unerheblichen Marktanteil.

Android verwendet eine eigens entwickelte Laufzeitumgebung, die den Bytecode einer in Java geschriebenen Android-Anwendung in den nativen Befehlssatz umwandelt. Die aktuell seit Android 5.0 unterstützte Laufzeitumgebung „Android Runtime“ ersetzt hierbei die in vorherigen Versionen eingesetzte virtuelle Maschine „Dalvik“. Die neue Laufzeitumgebung verwendet anstelle eines Just-in-Time-Compilers nun einen Ahead-of-Time-Compiler, der den Bytecode bei der Installation in den nativen Befehlssatz umwandelt.

Der Google-Playstore hat an sich eine Vielzahl an Applikationen im Sortiment, die die Funktionalität eines Android-Smartphones beliebig erweitern können. Eine beliebte Entwicklungsumgebung für das Erstellen von Android-Applikationen ist beispielswiese Android Studio von Google. Die Entwicklerumgebung basiert auf der Entwicklungsumgebung „IntelliJ IDEA“ des Softwareunternehmens JetBrains und ist insbesondere für Einsteiger sehr gut geeignet. Die Software gibt es für Windows, Linux und Mac-OS X und benötigt nur das Java SE Development Kit.

GRUNDLEGENDER AUFBAU EINER ANDROID-APPLIKATION

Eine Android-Applikation besteht aus vielen verschiedenen Komponenten, die hier nun genauer erläutert werden:

Eine *Activity* ist eine Applikationskomponente, mit der ein Benutzer interagieren kann. Üblicherweise besteht die *Activity* aus einem Fenster, welches Steuerelemente für die Interaktion mit dem Benutzer beinhaltet und oftmals den gesamten Bildschirm füllt. Eine Android-Applikation besteht meistens aus mehreren *Activities*, die miteinander verbunden sind.

```
public class MainActivity extends Activity
```

Ein Service ist eine Komponente, die im Hintergrund verschiedenste Aufgaben ausführt. Diese Hintergrundaufgaben können beispielsweise Musik abspielen, wenn der Benutzer gerade eine andere Applikation verwendet.

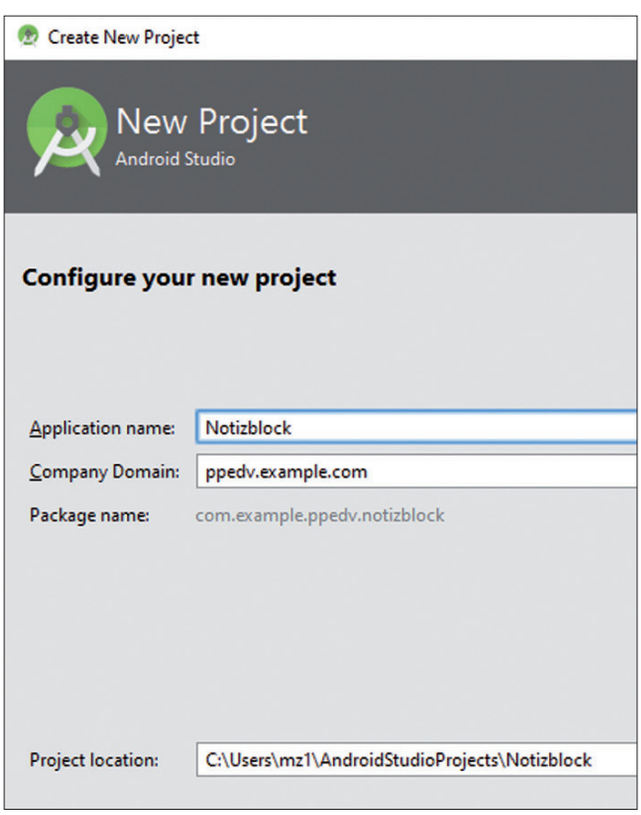


Abbildung 1

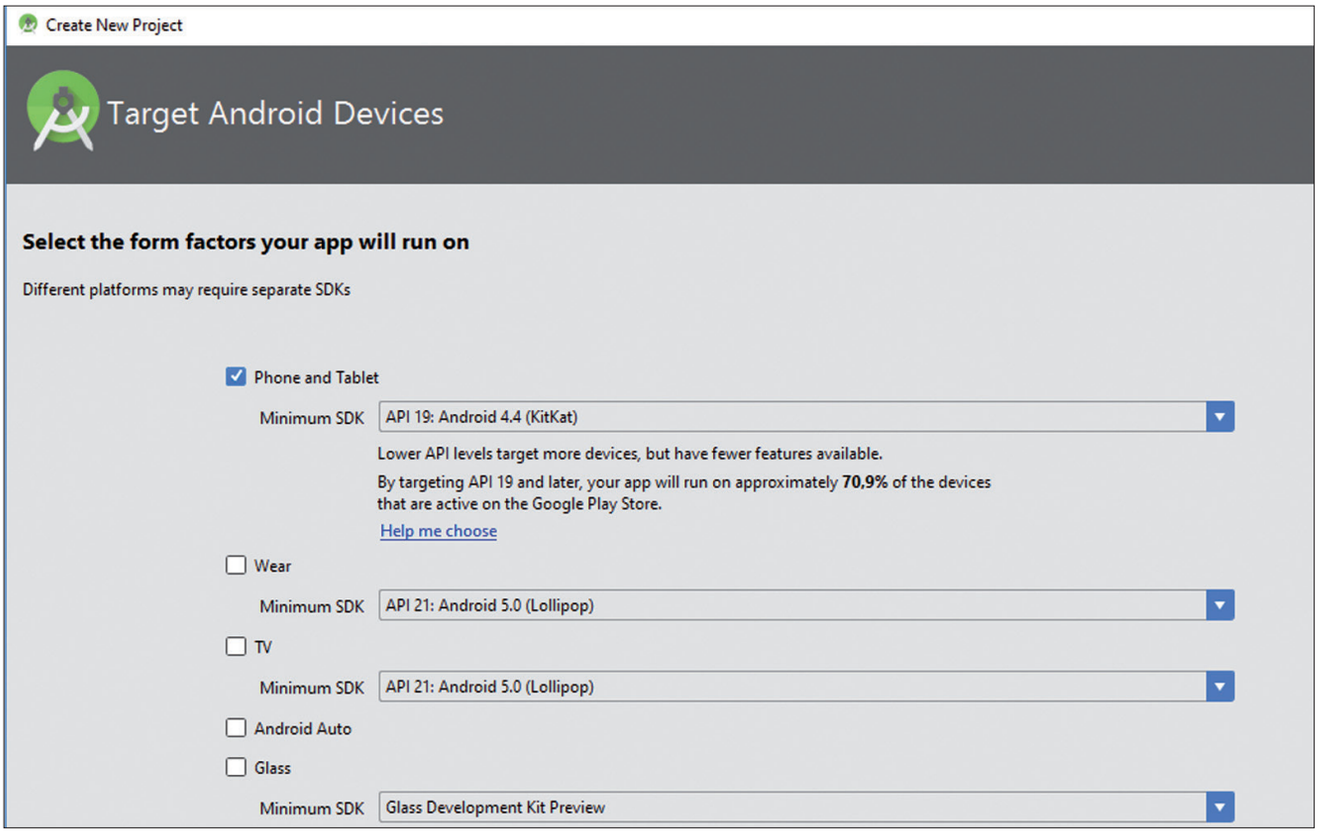


Abbildung 2

```
public class MyService extends Service
```

Broadcast Receiver sind Komponenten, die auf Broadcast-Nachrichten von anderen Systemkomponenten horchen. Applikationen können zum Beispiel einen Broadcast aussenden, wenn der Download einer Datei erfolgreich beendet wurde. Der Broadcast Receiver kann dann folglich entsprechende Aktionen ausführen.

```
public class MyReceiver extends BroadcastReceiver
```

Content Provider können Daten zwischen den einzelnen Applikationen auf dem System verteilen. Die einzelnen Anfragen werden zumeist über Methoden einer ContentResolver-Klasse abgearbeitet.

```
public class MyContentProvider
extends ContentProvider
```

All diese Komponenten werden in der *AndroidManifest.xml* – Datei miteinander verknüpft.

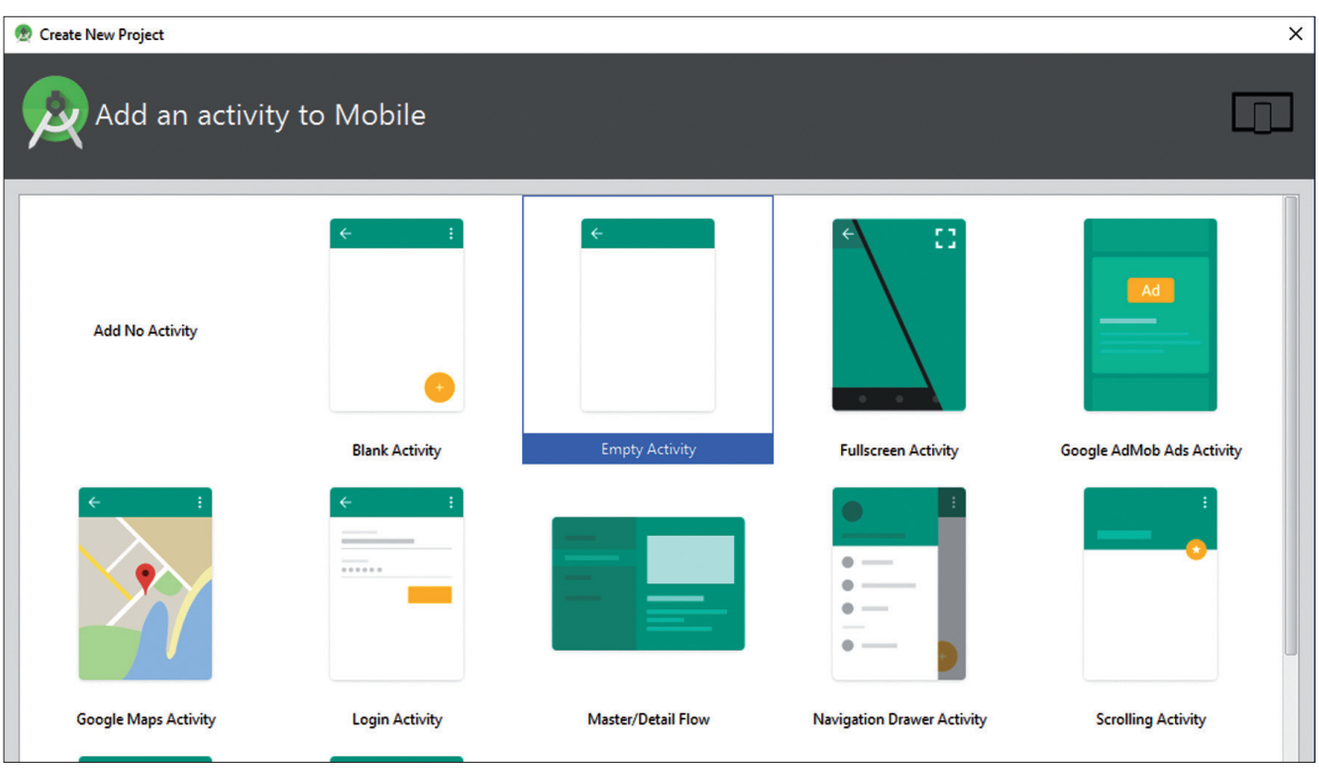


Abbildung 3

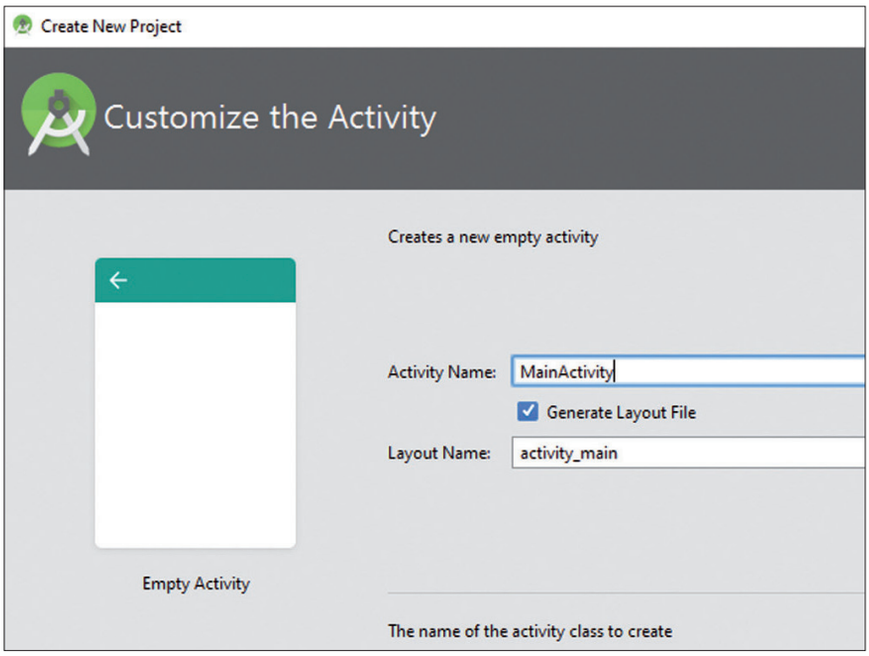


Abbildung 4

DIE BEISPIEL-APP: DER NOTIZBLOCK

In diesem Tutorial werden wir eine einfache Android-Applikation erstellen, mit der wir uns eine Notiz auf unserem Smartphone speichern können. Der Inhalt der Notiz wird in eine Textdatei im lokalen Speicher abgelegt.

Zu diesem Zweck müssen wir einmal ein neues Projekt mit dem Namen „Notizblock" in Android Studio anlegen. Das erste Dialogfenster können wir in Bild 1 sehen.

Im nächsten Dialogfenster (Bild 2) können wir das Gerät wählen, auf dem die Applikation laufen soll. Die Wahl des SDKs (Software Development Kits) ist die erste weitreichende Entscheidung, die wir beim Erstellen der Applikation treffen müssen. Da für jede große Android Version neue bzw. angepasste APIs („Application Programming Interface") herausgegeben werden, können wir mit unserer Wahl steuern, welche Android-Versionen unsere Applikation unterstützten soll. In diesem Beispiel wähle ich die API 19 für Android 4.4 (KitKat) aus, um möglichst viele aktuelle Geräte abzudecken.

Nachdem wir unsere Auswahl bestätigen, können wir uns im nächsten Dialogfenster eine Activity aussuchen. Im diesem Fenster (Bild3) können wir aus mehreren Vorlagen eine Activity auswählen. In unserem Fall reicht eine leere Activity vollkommen aus. Im nächsten Dialogfenster (Bild 4) kann man optional noch den Namen der Activity und den Namen der Layout-Datei ändern.

Letztendlich landen wir nun in der Entwicklerumgebung, die uns gleich einmal die gesamte Projektstruktur unserer Applikation anzeigt. Folgende Dateien sind hierbei von besonderer Bedeutung für uns:

AndroidManifest.xml

Die AndroidManifest.xml – Datei ist eine der wichtigsten Dateien in einer Android-Anwendung. Alle Komponenten einer Android-Anwendung müssen in dieser Datei beschrieben werden, damit die Anwendung auf die angeforderten Ressourcen zugreifen kann. Des Weiteren werden im Manifest auch die benötigten Zugriffsrechte und die erforderliche API-Version angegeben.

In der Manifest-Datei, die in Listing 1 abgebildet wird, werden alle wichtigen Informationen über unsere Applikation abgelegt. Grundsätzlich ist das Activity-Element eines der wichtigsten Bestandteile des Manifests, da im Manifest alle Activities einer Applikation angegeben werden müssen. Hier können wir sehen, dass unsere Applikation nur eine Activity, folglich nur ein einzelnes „Fenster", haben wird. Der Intent-Filter gibt an, welche Aufgaben oder Rollen eine Activity erfüllt. Die Aktion `„android.intent.action.MAIN"` zeigt an, das sich in der Activity der

Listing 1

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ppedv.notizblock">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Listing 2

```
package com.example.ppedv.notizblock;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Listing 3

```
package com.example.ppedv.notizblock;

public final class R {
    public static final class anim {
        public static final int abc_fade_in=0x7f050000;
        public static final int abc_fade_out=0x7f050001;
        public static final int abc_grow_fade_in_from_bottom=0x7f050002;
        public static final int abc_popup_enter=0x7f050003;
        public static final int abc_popup_exit=0x7f050004;
        public static final int abc_shrink_fade_out_from_
bottom=0x7f050005;
        public static final int abc_slide_in_bottom=0x7f050006;
        public static final int abc_slide_in_top=0x7f050007;
        public static final int abc_slide_out_bottom=0x7f050008;
        public static final int abc_slide_out_top=0x7f050009;
    }
```

Listing 4

```
<resources>
    <string name="app_name">Notizblock</string>
    <string name="str_textViewHeader">Bitte geben Sie Ihre Notiz ein:</
string>
    <string name="str_buttonSpeichern">Speichern</string>
    <string name="str_NotizGespeichert">Ihre Notiz wurde soeben
gespeichert</string>
</resources>
```

Einstiegspunkt in die Applikation befindet. Die Kategorie „`android.intent.category.LAUNCHER`" gibt an, dass das Icon der Activity im Menü des Gerätes erscheinen soll.

MainActivity.java

Diese Datei beinhaltet den Quellcode für unsere Activity. Eine universell einsetzbare Activity-Klasse wird automatisch beim Erstellen des Projektes angelegt und könnte wie in Listing 2 aussehen.

Die `onCreate` Methode ist eine der vielen Methoden, die beim Start der Applikation ausgeführt wird. `R.layout.activity_main` verweist hierbei auf die activity_main.xml Datei, die das Layout der Activity definiert. (siehe Listing 5)

R-Datei

Die R-Datei ist eine automatisch generierte Datei, die in der Klasse „R" die IDs für alle Ressourcen zusammenfasst. Diese Datei sollte auf keinen Fall verändert werden. Ein kleiner Auszug aus der R-Datei könnte wie in Listing 3 aussehen.

Strings.xml

Die strings.xml Datei beinhaltet alle Beschriftungen und Texte, die in der Applikation verwendet werden. Darunter fallen zum Beispiel die Beschriftungen von Steuerelementen (Buttons, Labels, ...) oder auch der Text in einer Toast-Nachricht.

Da alle Beschriftungen, die in der Applikation vorkommen könnten, in dieser Datei zusammengefasst werden, ist das Erstellen einer lokalisierten Version der Applikation sehr simpel, da man im Applikationscode sowie in der Layout-Datei mithilfe einer eindeutigen ID auf die Beschriftung verweisen kann.

Unsere strings.xml Datei wird wie in Listing 4 aussehen.

In unserer Applikation werden wir nur 4 Strings verwenden: Den Namen der Applikation selbst, die Beschriftung von zwei Steuerelementen und den Text einer Toast-Benachrichtigung. In der finalen MainActivity.java Datei, die in Listing 6 genauer definiert wird, werden wir dann sehen, wie wir auf die Strings zugreifen können.

activity_main.xml

Diese Datei beinhaltet die Definition des Layouts. Die Benutzeroberfläche wird basierend auf dem Inhalt der Datei automatisch generiert. Für unsere Notizblock-Applikation benötigen wir die nun eine Layout-Definition, die in Listing 5 genauer definiert ist.

Für das Layout unserer Applikation wird ein Container mit dem Namen LinearLayout verwendet. Mithilfe des Containers können wir die darin enthaltenen Steuerelemente sehr leicht positionieren, weil der Container die Anordnung der Steuerelemente für uns übernimmt. In diesem Container werden alle Steuerelemente

Listing 5

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin
"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

    tools:context="com.example.ppedv.notizblock.MainActivity"
    android:weightSum="1">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/str_textViewHeader"
        android:id="@+id/textViewHeader" />

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:inputType="textMultiLine"
        android:ems="10"
        android:id="@+id/editTextNotiz"
        android:layout_marginTop="23dp"
        android:layout_weight="1.00" />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/str_buttonSpeichern"
        android:textAllCaps="true"
        android:id="@+id/buttonSpeichern" />
</LinearLayout>
```

nacheinander aufgereiht, sodass man mit relativ wenig Code eine listenartige Ansicht generieren kann. Für komplexere Oberflächen stehen selbstverständlich auch noch andere Container zur Verfügung.

Zunächst muss mit der Eigenschaft android:orientation die Orientierungsrichtung angeben werden. Mit den Eigenschaften layout_width und layout_height geben wir an, dass der Container den gesamten Bildschirm füllen soll.

Unsere Steuerelemente orientieren sich allesamt an der Breite des Containers. Das Steuerelement für die Texteingabe (editText) soll den gesamten verfügbaren Platz zwischen der Aufschrift und dem Button einnehmen. Um dies zu erreichen wird hier die layout_height auf odp und die Eigenschaft layout_weight auf 1.00 gesetzt. Mit diesem Trick können wir die gesamte freie Fläche in der Mitte des Bildschirms füllen. Letztendlich verweisen sowohl der Button als auch die Beschriftung die Text-Eigenschaft auf einen String in unserer String.xml – Datei (siehe Listing 4). Die nun fertige Oberfläche schaut dann letztendlich wie in Bild 5 aus.

Applikationslogik

Letztendlich kommen wir nun zu unserer MainActivity.java-Datei, die unsere gesamte Applikationslogik beinhaltet. Folgende Zeilen Code, die ich in Listing 6 zusammengefast habe, benötigen wir für unser Programm.
Zu allererst müssen wir ein paar Variablen für den Zugriff auf die Steuerelemente der Activity deklarieren: In der Methode `onCreate`, die beim Start der Applikation ausgeführt wird, holen wir uns die einzelnen Steuerelemente aus der Activity. Diese Steuerelemente finden wir, indem wir in der Activity nach einer ID, die in der automatisch generierten R-Datei definiert ist, suchen. Unserem Button fügen wir bei der Gelegenheit gleich noch einen OnClickListener hinzu. An dieser Stelle hinterlegen wir auch eine anonyme Methode, die immer dann ausgeführt wird, wenn wir auf den Button klicken.

Das Auslesen und Speichern der Datei habe ich zur Wahrung der Übersicht in 2 Methoden unterteilt: In der Methode „NotizSpeichern“ nutzen wir einen OutputStreamWriter um eine Zeichenkette in eine Datei mit dem Namen „Notiz.txt“ zu schreiben. Die Datei befindet sich in folgendem Verzeichnis: `/data/data/com.example.ppedv.Notizblock/files/“`. Der Zugriff auf dieses Applikationsverzeichnis ist jederzeit möglich. Für den Fall, dass Daten auf der SD-Karte oder an anderen Orten abgelegt werden sollen, werden entsprechende Berechtigungen benötigt.

Beim Laden der Notiz wird vorher erst einmal überprüft, ob die Datei überhaupt existiert. Für den Fall, dass die Datei tatsächlich existiert, lesen wir sie zeilenweise aus und zeigen dann den String in der mehrzeiligen TextBox an.

Falls beim Speichern oder beim Laden eine Exception auftreten sollte, bekommt der Benutzer eine Toast-Notification, die einige Sekunden lang auf seinem Bildschirm angezeigt wird. Wenn die Datei erfolgreich gespeichert wurde, wird ebenfalls eine Toast-Notification mit einer Standardnachricht aus der String.xml-Datei angezeigt (siehe Listing 4)

Die fertige Applikation kann man sowohl in einem Emulator als auch auf einem Android-Gerät testen. Zum

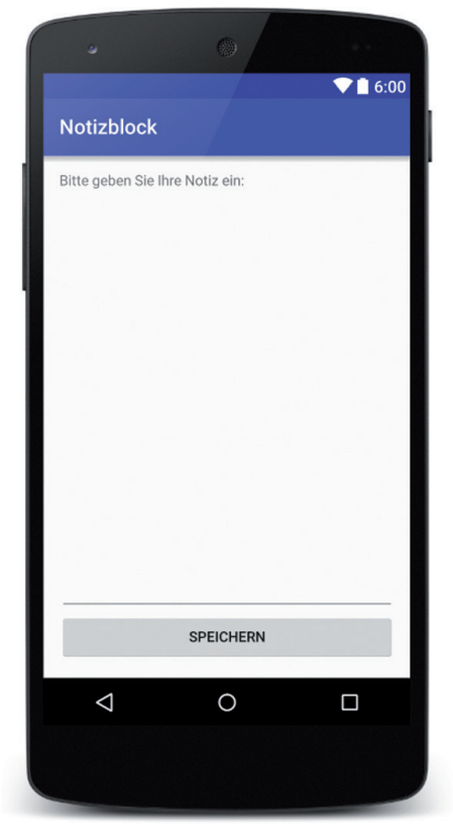


Abbildung 5

Installieren wird eine .apk-Datei benötigt, die in folgendem Verzeichnis liegt: `„Notizblock\app\build\outputs\apk“`.

Eine Alternative zur Entwicklung von Android-Applikationen mit Android Studio ist Xamarin. Mithilfe von Xamarin und dem Visual Studio ist es möglich, native Cross-Plattform-Applikationen mithilfe des .NET-Frameworks und C# zu programmieren. Diese Applikationen können sowohl in Android als auch unter iOS und Windows Phone 8 genutzt werden.

HILFREICHE LINKS FÜR DEN EINSTIEG

http://developer.android.com/tools/studio/index.html
http://developer.android.com/guide/index.html
https://developers.google.com/android/
http://android-developers.blogspot.de/

MICHAEL ZÖHLING
ist als Developer im Microsoft-Umfeld tätig. Er hat in seiner HTL-Laufbahn seine Leidenschaft für das Programmieren entdeckt und befasst sich leidenschaftlich mit dem .NET Framework.

Listing 6

```
package com.example.ppedv.notizblock;

import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;

public class MainActivity extends AppCompatActivity {

    // Variablen für den Zugriff auf die Steuerelemente
    private static Button btnSpeichern = null;
    private static EditText txtNotiz = null;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnSpeichern = (Button) findViewById(R.id.buttonSpeichern);
        btnSpeichern.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                NotizSpeichern();
                Toast.makeText(MainActivity.this, R.string.str_NotizGespeichert, Toast.LENGTH_SHORT).show();
            }
        } );

        txtNotiz = (EditText) findViewById(R.id.editTextNotiz);
        NotizLaden();
    }

    public void NotizSpeichern()
    {
        try {
            OutputStreamWriter out = new OutputStreamWriter(openFileOutput(„Notiz.txt“, Context.MODE_PRIVATE));
            out.write(txtNotiz.getText().toString());
            out.close();
        }
        catch (IOException ex) {
            Toast.makeText(MainActivity.this, „Fehler: „+ex.toString(), Toast.LENGTH_LONG).show();
        }
    }

    public void NotizLaden()
    {
        try {
            File file = getFileStreamPath(„Notiz.txt“);

            if (file.exists()) {
                InputStream inputS = openFileInput(„Notiz.txt“);

                if (inputS != null) {
                    InputStreamReader inptStrmRdr = new InputStreamReader(inputS);
                    BufferedReader br = new BufferedReader(inptStrmRdr);
                    String zeile = „“;
                    StringBuilder stringBuilder = new StringBuilder();

                    while ((zeile = br.readLine()) != null) {
                        stringBuilder.append(zeile+“\n“);
                    }
                    inputS.close();
                    txtNotiz.setText(stringBuilder.toString());
                }
            }
        }
        catch (Exception ex) {
            Toast.makeText(MainActivity.this, „Fehler: „+ex.toString(), Toast.LENGTH_LONG).show();
        }
    }
}
```

NEUERUNGEN IN SHAREPOINT 2016

Die Welt der SharePoint Interessenten hat schon lange auf die neue Version gewartet. Seit vergangenen Mai wissen wir nach der Ignite in Chicago konkreter, um welche Version es sich handeln würde. Der thematische Schwerpunkt liegt auf den Themen Infrastructure, Hybrid-Cloud, Performance und Compliance. Es war lange Zeit nicht klar, ob ein Nachfolger von SharePoint 2013 veröffentlicht werden würde, da die Strategie vom Microsoft vorneweg „Cloud first" in aller Munde war. Es kursieren immer noch Gerüchte, dass SharePoint 2016 das letzte Release sein wird. Aufbauend auf der Codebase von SharePoint Online wurde der neue SharePoint Server entwickelt. Microsoft setzt den Fokus auf Stabilität und Performance nach der Devise „Evolution statt Revolution". Keine umfangreichen Funktionserweiterungen, stattdessen viele kleine detailbehaftete Optimierungen. Die Veröffentlichung erfolgte im März zunächst als Testversion und im Mai sollen die Product-Keys auch zur Verfügung stehen.

Betrachtet werden die Neuerungen nachfolgend aus der Perspektive der IT-Pros und der Administratoren.

MINROLE

Es gibt bei der Installation von SharePoint für Farmadministratoren die Möglichkeit den einzelnen Server mit einer Rolle in der Farmtopologie auszustatten. Die Server, die in eine vorhandene Farm hinzugefügt werden oder wenn eine neue Farm erstellt wird erhalten per definierter Rolle automatisch Dienste konfiguriert und die Leistung wird optimiert. Der Health Analyzer prüft jeden Tag, ob alle Dienste, die zu der jeweiligen Rolle gehören, aktiviert sind und laufen. Weicht das Ergebnis vom Soll-Zustand ab, werden Korrekturen per Rolle automatisch vorgenommen, d.h. es werden sowohl Dienste die fehlen hinzugefügt als auch Dienste entfernt.

ZERO DOWNTIME

Die Anzahl der benötigten Patches werden drastisch reduziert und es wird kein Neustart mehr benötigt, so dass SharePoint im Betrieb das Aktualisieren und Einspielen von Produktverbesserungen ohne Ausfall ermöglicht.

COMPLIANCE RICHTLINIEN

Neue Compliance-Features erlauben das dedizierte Löschen nach einem bestimmten Zeitraum oder auch das Verbot zu löschen für Benutzerdokumente, E-Mails und weitere Dokumente. Die In-Situ-Speicherungsrichtlinie ermöglicht Administratoren, Dokumente, E-Mails und andere Dateien beizubehalten.

HYBRIDWEBSITES

Der Einsatz des neuen Hybrid Szenarios vereint die on-Premise Umgebung mit der Cloud (Office 365) und ermöglicht somit einerseits wichtige Daten weiterhin im Unternehmen zu behalten, jedoch auch Daten in der

Serverrolle	Beschreibung
Front-End	Dienstanwendungen, Dienste und Komponenten, die Benutzeranforderungen auf Front-End-Webservern verarbeiten. Diese Server sind für eine schnelle Leistung optimiert
Anwendung	Dienstanwendungen, Dienste und Komponenten, die Back-End-Anforderungen auf Anwendungsservern verarbeiten (z. B. Hintergrundaufträge oder Suchdurchforstungsanforderungen). Diese Server sind für hohen Durchsatz optimiert
Verteilter Cache	Dienstanwendungen, Dienste und Komponenten, die für einen verteilten Cache auf verteilten Cacheservern erforderlich sind. Optional können Sie diese Server auch für den Lastenausgleich des Farmdatenverkehrs mithilfe des SharePoint-Anforderungs-Managers konfigurieren
Suche	Dienstanwendungen, Dienste und Komponenten für die Suche müssen auf Suchservern gespeichert sein.
Benutzerdefiniert	Benutzerdefinierte Dienstanwendungen, Dienste und Komponenten, die nicht in MinRole integriert sind, müssen auf benutzerdefinierten Servern konfiguriert sein. Der Farmadministrator hat die vollständige Kontrolle darüber, welche Dienstinstanzen auf Servern ausgeführt werden können, denen die benutzerdefinierte Rolle zugeordnet ist. Mithilfe von MinRole wird nicht gesteuert, welche Dienstinstanzen für diese Rolle bereitgestellt werden
Farm mit einem Server	Dienstanwendungen, Dienste und Komponenten, die für eine Farm mit nur einem Computer erforderlich sind, müssen in einer Einzelserverfarm konfiguriert sein. Eine Farm mit nur einem Server ist für die Entwicklung, zu Testzwecken und eine sehr begrenzte Produktion vorgesehen. Eine SharePoint-Farm mit der Rolle „Farm mit einem Server" kann nicht mehr als einen SharePoint-Server in der Farm enthalten.

Neue Limits: Mithin die größten Vorteile ergeben sich beim Blick auf die neuen Limits, die sich wirklich sehen lassen können, hier hat Microsoft *ordentliche Verbesserungen eingebracht.*

Cloud zu speichern, um andererseits diese anderen Personen zugänglich zu machen. Benutzer erhalten aufgrund der neuen Features bei der Verwendung von SharePoint Server und SharePoint Online-Websites eine integrierte Benutzerumgebung:

- Anzeige gefolgter SharePoint Server- und SharePoint Online-Websites in einer Liste
- Verwaltung aller Profilinformationen über ein in Office 365 vorhandenes Profil

Die Benutzer haben durch die Hybriden Sites ein einheitliches Profil. Daten die on-Premise gespeichert werden, werden auch im Cloud Profil angezeigt. Der Benutzer erhält zudem eine Übersicht aller Seiten denen der folgt.

ONEDRIVE FOR BUSINESS-HYBRIDBEREITSTELLUNG

Der Einsatz von Hybrid OneDrive kann der Anwender einzelne Dokumente direkt im Office 365 zur Bearbeitung für andere Teammitglieder bereitstellen.
Der Zugriff auf die Daten erfolgt von jedem Endgerät, wenn OneDrive for Business-Hybridbereitstellungen verwendet werden:

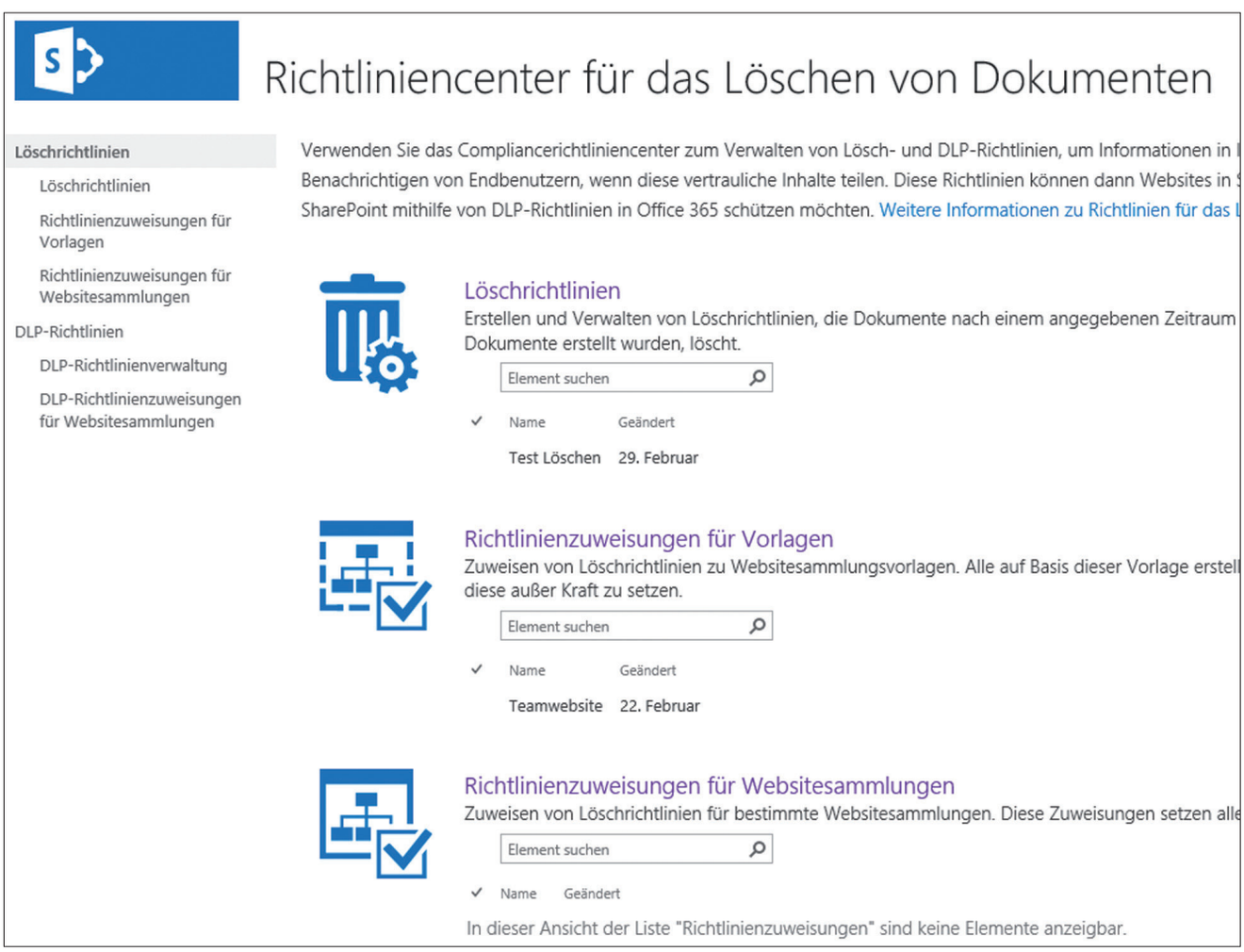
- Benutzer können Dateien mit Office 365 synchronisieren und für andere Personen freigeben
- Benutzer können direkt über Office 365 von jedem Gerät aus auf ihre Dateien zugreifen

CLOUD-HYBRIDSUCHE

Da die Unternehmenssuche ein wichtiger Bestandteil im Unternehmen ist, werden alle Informationen mit einer Suchanfrage angesprochen und der Benutzer erhält alle Inhalte zusammengefasst.
Die Cloud-Hybridsuche bietet folgende neue alternative Möglichkeiten:

- Die gesamten Inhalte, einschließlich der lokalen Inhalte, werden im Suchindex in Office 365. durchforstet.
- Abfragen, die im Suchindex in Office 365 durchgeführt werden, führen zu einheitlichen Suchergebnissen sowohl für lokale als auch Office 365-Inhalte in einem Ergebnisblock.

Inhalts-Datenbank Größe	Anzahl Websitesammlungen pro Inhaltsdatenbank	Listenbeschränkung	Maximale Dateigröße	Verbesserter Index
das genaue limit steht noch nicht fest (vorher 200 GB und unter bestimmten Voraussetzungen 4 TB)	100.000 (vorher max. 10.000, normale Websitesammlungen 2.500)	mehr als 5.000 objekte (vorher exakt 5.000)	10 GB (vorher 2 GB)	500 Millionen (vorher 100 Millionen)



Hybrid-Umgebungen in SharePoint 2016

VORSCHAU VON BILD- UND VIDEODATEIEN

In SharePoint Server 2016 gibt es nun auch eine Vorschau von Bildern und Videos in einer Dokumentbibliothek, wenn der Mauszeiger über das Bild oder Video bewegt wird.

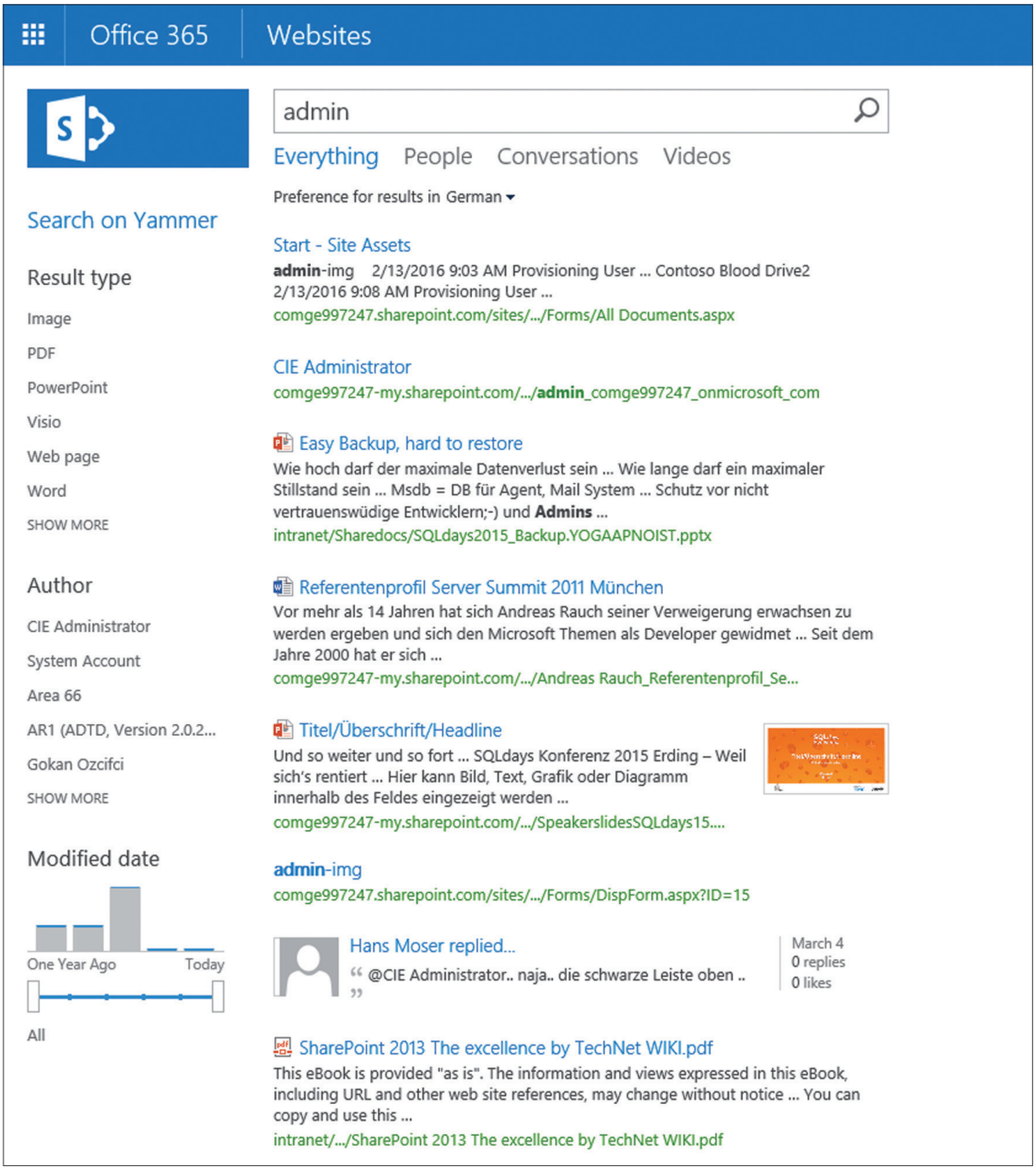
DURABLE LINKS

Dauerhafte Links ermöglichen es, Dateien aufzufinden, ganz egal wo sie liegen. Es handelt sich hierbei um eine Art link-redirect, wobei Änderungen des Dateinamens oder des Speicherortes nicht zum Bruch und Verlust des Hyperlinks führen, und das auch noch innerhalb der gesamten Websitesammlung. Die Dokumente sind mittels einer ID davor geschützt.

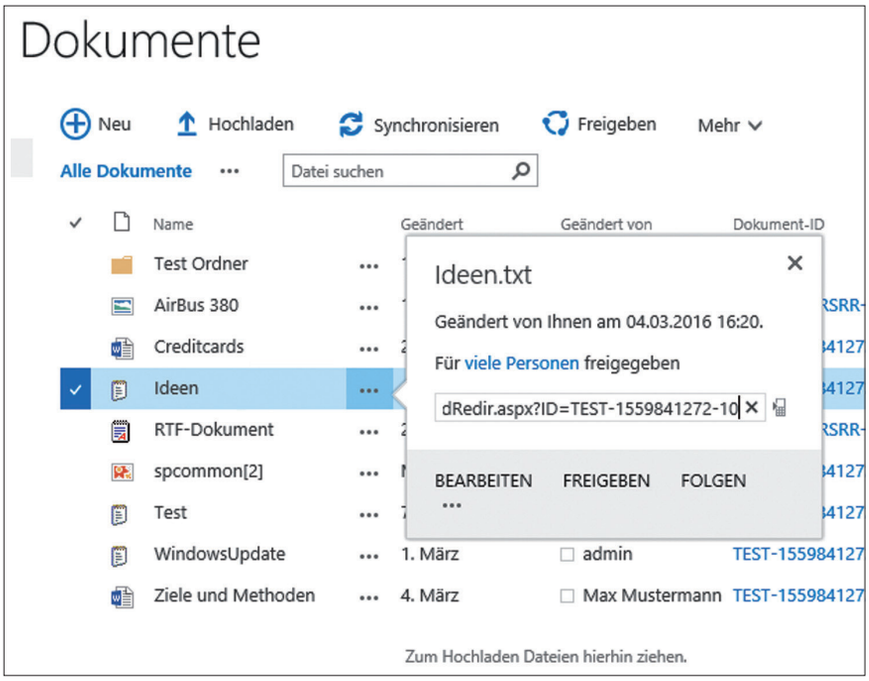
DOKUMENTBIBLIOTHEK-BARRIEREFREIHEIT

Für die Arbeit in SharePoint Server 2016 Dokumentbibliotheken stehen nun die folgenden Bedienungen per Shortcuts zur Verfügung:

Alt+N: Neu
Alt+E: Bearbeiten
Alt+U: Upload
Alt+M: Verwalten
Alt+S: Freigeben
Alt+Y: Synchronisieren



Vorschau von Bild- und Videodateien: In SharePoint Server 2016 gibt es nun auch eine Vorschau von Bildern und Videos in einer Dokumentbibliothek, wenn der Mauszeiger über das Bild oder Video bewegt wird.



FAZIT: SPANNEND UND VIELVERSPRECHEND

Alle Optimierungen sind durchaus begrüßenswert und interessant. SharePoint 2016 profitiert bereits und wird sicher weiterhin von den Features aus der Cloud profitieren. Noch wissen wir nicht wie die endgültige Version von SharePoint 2016 erscheinen wird. Es bleibt spannend!

REMIGIUSZ SUSKIEWICZ
ist seit 2001 als Trainer für die ppedv tätig Seine Themenschwerpunkte sind SharePoint, Projektmanagement, Office-Anwendungen sowie die Analyse von Daten. Insbesondere konzentriert er sich auf die Datenanalyse mit Excel und BI in SharePoint.
RemigiuszS@ppedv.de

.NET DAY FRANKEN

Die fränkische Community-Konferenz

11. Juni 2016 Nürnberg NH-Hotel City

Find us on: facebook

www.dotnet-day-franken.de

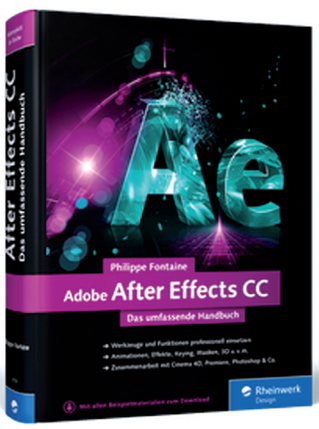
.NET DAY F

NEUE BÜCHER

ADOBE AFTER EFFECTS CC

DAS UMFASSENDE HANDBUCH

von Philippe Fontaine



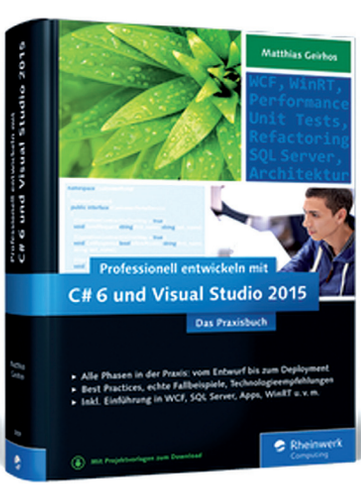
Buch € 79,90
E-Book € 69,90
814 Seiten, 2015, gebunden, in Farbe
ISBN 978-3-8362-3709-3

Atemberaubende Compositings und kinoreife visuelle Effekte – das ist es, was Sie mit After Effects CC machen wollen. Dieses umfassende Handbuch gibt Ihnen das nötige Handwerkszeug und erklärt die Software von A bis Z: Animationen, Motion-Tracking, Keying-Effekte und Bluescreen-Technik, Masken, 3D-Compositings u. v. m. Sie lernen das Programm Schritt für Schritt kennen und setzen die Werkzeuge und Funktionen in zahlreichen Praxis-Workshops kreativ und professionell ein.

PROFESSIONELL ENTWICKELN MIT C# 6 UND VISUAL STUDIO 2015

DAS PRAXISBUCH

von Matthias Geirhos



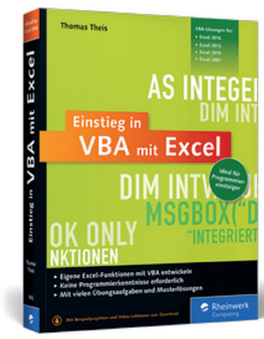
Buch € 49,90
E-Book € 44,90
1209 Seiten, 3., aktualisierte und überarbeitete Auflage 2015, gebunden
ISBN 978-3-8362-3929-5

Sie beherrschen C#, möchten aber gerne noch effizienter entwickeln? In diesem Buch finden Sie eine Vielzahl an Dos & Don'ts, mit denen Sie alle Phasen Ihres Projekts sicher meistern: OOA & OOD, GUIs, TPL und Multithreading, Code Smells, WCF, ADO.NET, Workflow Foundation, Unit Tests, Softwarepflege Deployment u.v.m.

EINSTIEG IN VBA MIT EXCEL

VBA-LÖSUNGEN FÜR EXCEL 2007 BIS 2016

von Thomas Theiss



Buch € 19,90
E-Book € 16,90
455 Seiten, 4., aktualisierte Auflage 2015, broschiert
ISBN 978-3-8362-3962-2

Sie möchten eigene Funktionen für Excel entwickeln? In diesem Buch erfahren Sie, wie es geht. Die Grundlagen der VBA-Programmierung werden leicht verständlich an typischen Anwendungsbeispielen vermittelt. Auch wenn Sie bisher noch nie programmiert haben, werden Sie schnell Ihre ersten Makros schreiben.

CSS DAS UMFASSENDE HANDBUCH

von Kai Laborenz

Buch € 39,90
E-Book € 34,90
825 Seiten, 3., aktualisierte Auflage 2015, gebunden
ISBN 978-3-8362-3876-2

Das vollständige Wissen zu CSS und Co. in einem Band! Einsteiger erhalten eine fundierte Einführung, professionelle Webentwickler einen Überblick über alle CSS-Technologien und Praxislösungen für CSS-Layouts sowie zahlreiche Tipps für die tägliche Arbeit. Inkl. CSS-Animationen und -Transformationen, HTML5, CSS3, Responsive Webdesign, Sass u.v.m.

VBA MIT EXCEL

DAS UMFASSENDE HANDBUCH

von Bernd Held

Buch € 49,90 · E-Book € 44,90
950 Seiten, 2., aktualisierte und erweiterte Auflage 2015, gebunden
ISBN 978-3-8362-3821-2

Werden Sie unter Anleitung des Erfolgsautors Bernd Held zum Profi in VBA mit Excel. Von den Grundlagen (Datentypen, Variablen) und der Programmierung von Objekten (z. B. Zellen, Mappen, Diagramme) über Funktionen und Ereignisse bis hin zu UserForms oder Symbolleisten: Alles drin und mit zahlreichen Beispielen illustriert!

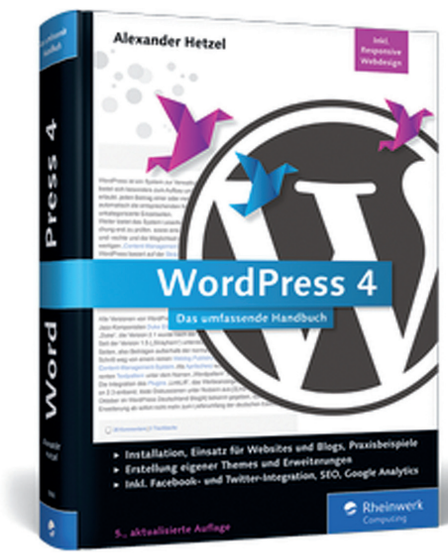
WORDPRESS 4

DAS UMFASSENDE HANDBUCH

von Alexander Hetzel

Buch € 39,90 · E-Book € 34,90
935 Seiten, 5., aktualisierte und erweiterte Auflage
2015, gebunden, ISBN 978-3-8362-3943-1

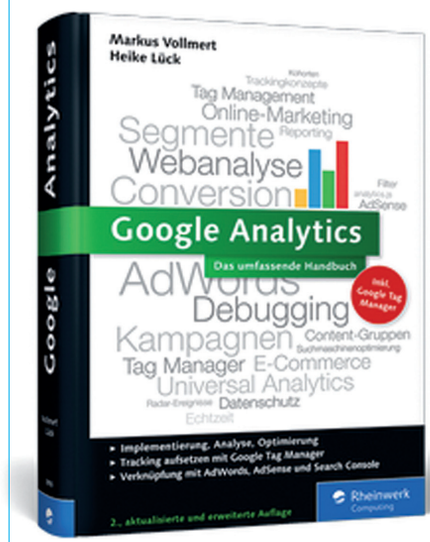
Mit WordPress können Sie weit mehr als nur ein Blog betreiben – es ist auch ein weitverbreitetes Content-Management-System. In diesem Buch finden Sie das umfassende Rundumpaket für jedes Anwendungsszenario: alle Installationen, Entwicklung und Administration eigener Themes und Plugins sowie einen Einstieg in HTML und CSS.



GOOGLE ANALYTICS

DAS UMFASSENDE HANDBUCH

von Markus Vollmert, Heike Lück



Buch € 39,90
E-Book € 34,90
853 Seiten, 2., aktualisierte und erweiterte Auflage
2015, gebunden
ISBN
978-3-8362-3955-4

Lernen Sie mit diesem Buch, wie Sie die vielfältigen Funktionen von Google Analytics nutzen und sie professionell einsetzen können. So erhalten Sie z.B. Hilfestellung dabei, wie Sie Ihr Webanalyse-System konzipieren und strukturieren sollten. Verständliche Beispiele für eine optimale Implementierung und ein erfolgreiches Monitoring all Ihrer Online-Aktivitäten ermöglichen den direkten Einsatz. Zudem können Sie aussagekräftige Berichte generieren, um Ihre Website und Ihre Online-Marketing-Aktivitäten zu optimieren.

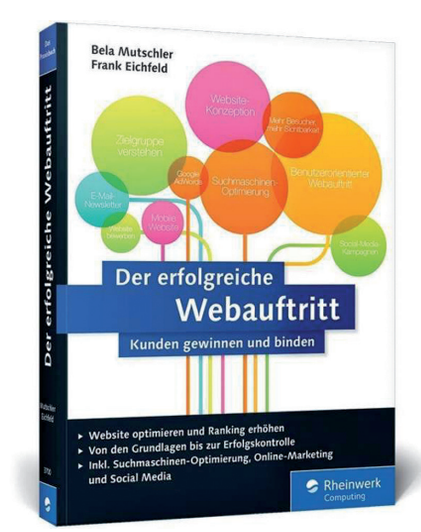
DER ERFOLGREICHE WEB-AUFTRITT

KUNDEN GEWINNEN UND BINDEN

von Bela Mutschler, Frank Eichfeld

Buch € 29,90 · E-Book € 24,90
359 Seiten, 2015, broschiert, ISBN 978-3-8362-3700-0

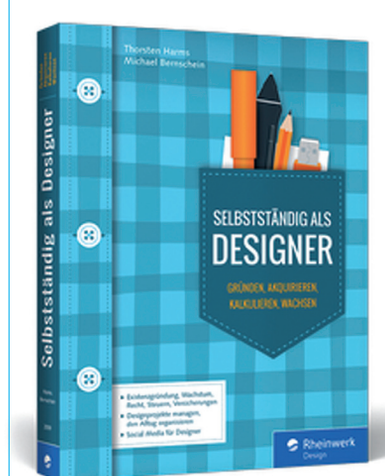
Warum sind manche Websites erfolgreich und helfen sehr gut bei der Kundengewinnung während andere kaum zur Kenntnis genommen werden? Warum haben einige Webauftritte viele Besucher, während andere um jeden Besucher kämpfen müssen und im Internet kaum sichtbar sind? Für erfolgreiche Websites gibt es kein einfaches, immer gleiches Patentrezept. Dennoch gibt es Regeln, die Ihnen dabei helfen, auch Ihre Website nach vorne zu bringen.



SELBSTSTÄNDIG ALS DESIGNER

GRÜNDEN, AKQUIRIEREN, KALKULIEREN, WACHSEN

von Thorsten Harms, Michael Bernschein



Buch € 34,90
E-Book € 29,90
400 Seiten, 2015, broschiert, in Farbe
ISBN:
978-3-8362-2559-5

Sie wollen sich in der Designbranche alleine oder im Team selbstständig machen? Dieses Buch hilft Ihnen bei allen wichtigen wirtschaftlichen Fragen zu Existenzgründung, Businessplan, Versicherungen, Steuern & Co. Aber auch organisatorisch kommt viel auf Sie zu: Wie komme ich an Aufträge, wie hoch kann mein Honorar sein, wie organisiere ich meinen Arbeitsalltag? Hier erfahren Sie, wie Sie die Klippen der Selbstständigkeit sicher umschiffen!

Es waren nur 11 Zeilen Code...

Es war der 20. März dieses Jahr, an dem die JavaScript Entwicklergemeinde in ein kleines schwarzes Loch fiel. Der bisher einwandfrei funktionierende Code lief auf einmal nicht mehr, obwohl niemand etwas Wichtiges verändert hat. Oder besser gesagt, es war nur einer: Ein 28-jähriger Entwickler aus Kalifornien. Seine Aktion hatte Auswirkungen auf viele Software Projekte.

Was ist passiert? Um das zu klären, müssen wir in der noch recht jungen Geschichte der Softwareentwicklung etwas zurückreisen:

Früher hatten wir ein Produkt, um Software zu schreiben. Ich habe zum Beispiel mit Borland C++ gearbeitet. Das war perfekt, denn entweder war die Funktion, die benötigt wurde, bereits in der mitgelieferten Bibliothek enthalten oder man hat sie sich einfach selbst geschrieben. Selbst simple Text-Funktionen, wie zum Beispiel LeftPad, schrieb man damals selbst!

Die Wiederverwendbarkeit von Sourcecode war damals ebenfalls ein großes Thema. Man war stets bemüht, seinen einmal geschriebenen Code in mehreren Projekten wiederzuverwenden um Zeit und damit auch Kosten zu sparen. Wenn wir das Rad der Zeit nun wieder vordrehen, sehen wir im Schnelldurchlauf die sogenannte DLL-Hell bei uns vorbeiziehen. Viele nützlichen Funktionen wurden oftmals in DLLs ausgelagert und dann als Bibliothek im Projekt referenziert. Mit dieser Technik, die bezüglich der Wiederverwendbarkeit sehr praktisch war, konnte man sehr gut Kosten sparen. Auch das Erwerben von bereits existierenden Bibliotheken war oftmals günstiger als eine saubere Eigenentwicklung. Das Konzept der „Software-Komponenten“ war somit geboren. Mit OCX, ActiveX oder COM hat wohl jeder Microsoft-Entwickler zumindest einmal zu tun gehabt.

Software hat als Produkt den Vorteil und gleichzeitig den Nachteil, dass sie beliebig kopiert und weiterverwendet werden kann, da für die Vervielfältigung keine weiteren Kosten anfallen. Zurzeit gibt es kein anderes Gut auf dem Markt, welches diese wichtige Eigenschaft hat. Wenn es keine Angebotseinschränkung gibt, spielt der Preis keine Rolle. Es ist egal, ob ich ein Stück oder tausende davon „kaufe“, da man nur das erste Stück erwerben muss und jede weitere Vervielfältigung „kostenlos“ ist.

Aus der Idee, dass die Software von allen verwendet werden soll, ist die Open Source Bewegung entstanden. Sourcecode kann und soll in Softwareprojekten beliebig eingesetzt werden. Die Bibliotheken werden oftmals von Enthusiasten in ihrer Freizeit geschrieben. Wie bei vielen Hobbies fehlt aber irgendwann der Antrieb, es weiter zu verfolgen. Sollte nun irgendwann der Antrieb verloren gegangen sein, kann man das Projekt entweder stilllegen oder jemanden anderen finden, der daran weiterarbeitet.

Mit dem Internet ist die Verteilung dieser Hobby-Software-Projekte ganz leicht. Meine geschriebene Bibliothek lade ich auf eine der unzähligen Plattformen hoch und freue mich, wenn andere sie verwenden oder weiterentwickeln. Und hier beginnt die Problematik.

Jene, die Software schreiben um davon zu leben, nutzen diese „freien“ Bibliotheken für ihre kommerziellen Projekte. Das macht aus der Überlegung der Wiederverwendbarkeit durchaus Sinn. Und damit sind wir in der Zeitreise in der Gegenwart angekommen:

Wenn in einem Projekt die eine oder andere Methode fehlt, kann man einfach einmal in Google nachsehen. Nur, Sourcecode mit Copy & Paste in das Projekt übertragen ist uncool. Wir nutzen lieber Bower, NuGet, Npm oder andere „Package Management Systeme“. Schnell mal das Paket hinzufügen, es kostet ja nichts. Früher gab es zumindest noch die Hürde, dass einige Bibliotheken kostenpflichtig waren und man die Ausgaben rechtfertigen musste. Das Handling war aufwendig und man überlegte lieber zweimal, ob diese Bibliothek wirklich notwendig ist.

Nun zurück zum 20. März. Unser Programmierer aus Kalifornien nannte sein Paket kik. Das hatte zwar nichts mit der gleichnamigen Handelskette zu tun, aber eine Handvoll an Rechtsanwälten pochte auf die Einhaltung der Namensrechte. Nach einigem Hin- und Her kam die Reaktion des Programmierers:

Seiner Meinung nach hat Npm zu schnell aufgegeben und ist vor den Anwälten eingeknickt.

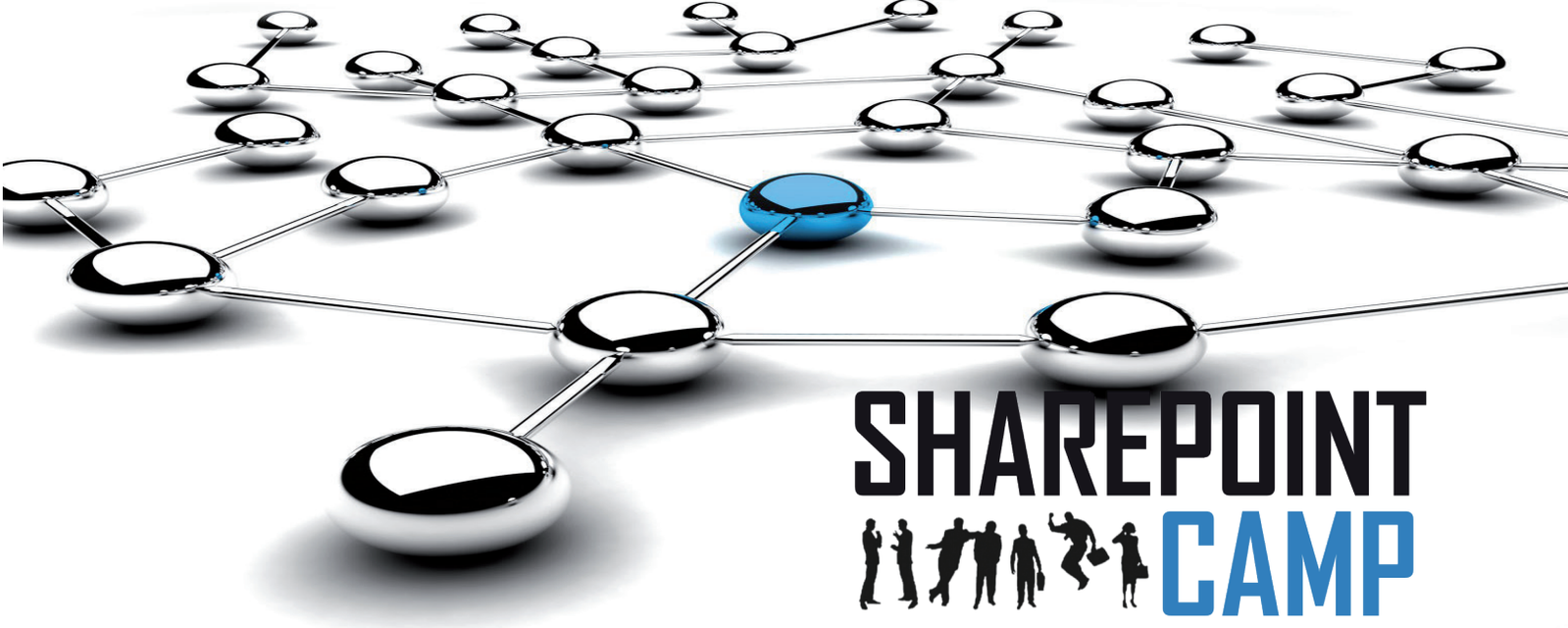
Er löschte das Paket und gleichzeitig damit alle weiteren von ihm auf Npm veröffentlichen Pakete.

Das Kik-Projekt selbst war aber nicht der Grund für den Super-GAU in der JavaScript-Community. Er hatte aber ebenfalls ein weit verbreitetes Paket mit dem Namen „leftpad“ angeboten, welches im letzten Monat 2.486.696-mal heruntergeladen wurde. Und das Fehlen des Paketes verursachte nun Probleme in vielen anderen Softwareprojekten.

```
module.exports = leftpad;
function leftpad (str, len, ch) {
  str = String(str);
  var i = -1;
  if (!ch && ch !== 0) ch = , ,;
  len = len - str.length;
  while (++i < len) {
    str = ch + str;
  }
  return str;
}
```

Das Fehlen von diesen 11 Zeilen Code, die jeder Programmierer selbst schreiben könnte, führte zu massiven Problemen in namhaften JavaScript-Projekten, wie zum Beispiel Node.js oder Babel. Aber bevor etwas selbst geschrieben wird, importieren wir es lieber von einer „coolen“ Plattform.

Liebe Programmier-Kollegen, nehmt das als Anlass genauer darüber nachzudenken, ob das blinde Importieren von Bibliotheken wirklich der richtige Weg ist, Software zu schreiben!



ERGREIFEN SIE DIE CHANCE UND LASSEN SIE SICH VON UNSEREN SHAREPOINT-PROFIS WEITERBILDEN

Ulrich Boddenberg

Martin Groblschegg

Andreas Rauch

Jörg Krause

Termine für das Jahr 2016

6.-10. JUNI
MÜNCHEN

04.-08. JULI
DÜSSELDORF

19.-23. SEPT.
BERLIN

Das SharePoint Camp richtet sich an Software-Entwickler, Systemverantwortliche und Projektleiter, die in kurzer Zeit in Microsoft SharePoint 2013 - mit Ausblick auf 2016 einsteigen möchten.

Dank unseres intensiven Trainingskonzeptes erzielen Sie innerhalb 5 Tagen von 9.00 - 18.00 Uhr individuelle Lernerfolge und erhalten einen Überblick in SharePoint-Technologien wie: Architektur, Tools, Administration, Social Networking, Business Intelligence, Workflows, WebParts, Backup, Restore, SharePoint Designer uvm.

Weitere Informationen finden Sie auf:

www.SharePointCamp.de

ppedv

ppedv

THE COACH IN YOU.

- PROJEKT-MANAGEMENT
- IT-TEAMS FÜHREN
- SCRUM & KANBAN

SOFT SKILL TRAININGS
FÜR TEAM LEADER

PPEDV.DE/COACH

© Beatrice Preve | Dreamstime.com

ppedv AG, Marktlerstr. 15b, 84489 Burghausen, Handelsregister Traunstein, HRB 12703,St.Nr. 131-45412, Vorstand: Hannes Preishuber. Bildquelle: © Brett Critchley | Dreamstime.com