

# SE IS **QualityNews**

**Analysis. Development. Test. Management**

**Ausgabe H1/2021**

## **Coding & Development**

### **Deliver & Maintain Real Life Solutions**

**Continuous Deployment  
Pipeline mit Gitlab**

Seite 12

**Entwicklung von  
Responsive Web Apps:  
Welches Framework sollte  
man nutzen?**

Seite 24

**Warum Node.js und  
Full Stack JavaScript  
die etablierten  
Programmiersprachen im  
Enterprise Umfeld ablösen  
wird...**

Seite 28

Titel: Gruppenbild Malgruppe „der verlassene Ort“ – Fließtechnik Fluid Acrylfarben

**Analysis. Development. Test. Management. Better IT Results.**

## Lesen Sie in dieser Ausgabe:

<b>Editorial.....</b>	<b>3</b>	<b>Referenzstory:</b>	
		<b>Peacequare GmbH.....</b>	<b>18</b>
		Dominik Tuchs Schmidt	
<b>Neulich im Netz.....</b>	<b>4</b>	<b>Eine kurze Vorstellung von R.....</b>	<b>20</b>
Hansjörg Münster		Claudia Amon	
<b>Aus der SEQIS Sales Ecke.....</b>	<b>5</b>	<b>Model-Driven Development.....</b>	<b>22</b>
Sandra Benseler		Daniel Kiesenhofer	
<b>Native Mobile Apps mit Native</b>		<b>Entwicklung von Responsive Web</b>	
<b>Script-Vue entwickeln .....</b>	<b>6</b>	<b>Apps: Welches Framework sollte</b>	
Michael Dinhof und Leon Palluch		<b>man nutzen?.....</b>	<b>24</b>
<b>Agile Circle Events 2021/22.....</b>	<b>9</b>	Markus Schwabeneder	
Helena Thurner		<b>ScriptRunner for Jira.....</b>	<b>26</b>
<b>Visual Studio Code, eine kostenlose</b>		Stefan Ladstätter	
<b>IDE revolutioniert die Softwareent-</b>		<b>Warum Node.js und Full Stack</b>	
<b>wicklung.....</b>	<b>10</b>	<b>JavaScript die etablierten Program-</b>	
Alexander Vukovic		<b>miersprachen im Enterprise Umfeld</b>	
<b>Continuous Deployment Pipeline</b>		<b>ablösen wird.....</b>	<b>28</b>
<b>mit Gitlab.....</b>	<b>12</b>	Alexander Vukovic	
Klemens Loschy		<b>5 Fragen an.....</b>	<b>31</b>
<b>SEQIS: Ten more things .....</b>	<b>15</b>	Alexander Weichselberger	
Helena Thurner		<b>SEQIS Kalender.....</b>	<b>16</b>
<b>Alle Termine auf einen Blick</b>			

### Ihre Meinung ist gefragt!

Nach den QualityNews ist bekanntlich vor den QualityNews!  
Schon bald arbeiten wir wieder auf Hochtouren an der nächsten, spannenden Ausgabe. Lesen Sie nur das, was Sie wirklich interessiert! Sagen Sie uns, welche Themen Sie spannend finden.

Kontaktieren Sie uns: [marketing@SEQIS.com](mailto:marketing@SEQIS.com)  
Join us: [twitter.com/SwTestIsCool](https://twitter.com/SwTestIsCool)

Wir freuen uns auf Ihre Vorschläge und Wünsche!

### Über SEQIS QualityNews:

Dieses Magazin richtet sich an Gleichgesinnte aus den Bereichen IT Analyse, Development, Softwaretest und Projektmanagement im IT Umfeld. Die SEQIS Experten berichten über ihre Erfahrungen zu aktuellen Themen in der Branche. Die Leser des Magazins gestalten die Ausgaben mit: Schreiben Sie uns Ihre Meinung im SEQIS Blog ([www.SEQIS.com/de/blog-index](http://www.SEQIS.com/de/blog-index)) oder als Leserbrief.

Wenn Sie dieses Magazin abbestellen möchten, senden Sie bitte ein Mail an [marketing@SEQIS.com](mailto:marketing@SEQIS.com).

### Impressum:

Information und Offenlegung gem.  
§5 E-Commerce-Gesetz und  
§25 Mediengesetz

Herausgeber: SEQIS GmbH,  
Neusiedler Straße 36, A-2340 Mödling  
Tel: +43 2236 320 320 0  
Fax: +43 2236 320 320 350  
[info@SEQIS.com](mailto:info@SEQIS.com), [www.SEQIS.com](http://www.SEQIS.com)  
Gericht: Bezirksgericht Mödling  
Firmenbuchnummer: 204918a  
Umsatzsteuer-ID: ATU51140607  
Geschäftsführung: Mag. (FH) Alexander  
Vukovic, Mag. (FH) Alexander  
Weichselberger, DI Reinhard Salomon

Druck: druck.at Druck- und Handels-  
gesellschaft mbH, 2544 Leobersdorf  
Erscheinungsweise: 2x pro Jahr  
Für die verwendeten Bilder und Grafiken  
liegen die Rechte für die Nutzung und  
Veröffentlichung in dieser Ausgabe vor.  
Die veröffentlichten Beiträge, Bilder und  
Grafiken sind urheberrechtlich geschützt.  
(Kunstwerke: Lebenshilfe Baden und  
Mödling, Fotos: Shutterstock).

Sämtliche in diesem Magazin zur  
Verfügung gestellten Informationen und  
Erklärungen geben die Meinung des  
jeweiligen Autors wieder und sind  
unverbindlich. Irrtümer oder Druckfehler  
sind vorbehalten. Hinweis im Sinne des  
Gleichbehandlungsgesetzes: Aus  
Gründen der leichteren Lesbarkeit wird die  
geschlechtsspezifische Differenzierung nicht  
durchgehend berücksichtigt. Entsprechende  
Begriffe gelten im Sinne der  
Gleichbehandlung für beide Geschlechter.





DI Reinhard Salomon

Mag. (FH) Alexander Vukovic

Mag. (FH) Alexander Weichselberger

# Editorial

Sehr geehrte Leserin,  
sehr geehrter Leser,

wir freuen uns, Ihnen die Ausgabe für  
das erste Halbjahr 2021 zu präsentieren.

Vielen Dank für das positive Feedback zu unseren letzten Ausgaben zu den Themenbereichen „The Next Big Thing in Software Engineering and Quality“. Wir hoffen, wir konnten Ihnen damit interessanten Content zur Verfügung stellen und Sie stellenweise auch gut unterhalten. Über weitere Anregungen, Themenwünsche und Feedback Ihrerseits freuen wir uns.

Auch in dieser Ausgabe finden Sie neben den branchenbezogenen Artikeln auch nicht-technische Bereiche:

Im Heft finden Sie einige Kunstwerke der Lebenshilfe Niederösterreich der Werkstätten Baden und Mödling. Denn nicht nur unsere Spezialisten, sondern auch die Klienten der Lebenshilfe leben für ihre(n) Beruf(ung).

In dieser Ausgabe dreht sich alles um das Thema „Coding & Development“ und bietet eine Auswahl von Fachartikeln mit diesem Schwerpunkt, die wir aus unserem [blog.seqis.com](http://blog.seqis.com) für Sie zusammengestellt haben.

Auf den folgenden Seiten geben Ihnen unsere Experten einen Einblick in die vielseitigen Aspekte zum Thema.

Wir wünschen Ihnen viel Lesevergnügen mit der aktuellen Ausgabe der SEQIS QualityNews!

Ihre SEQIS Geschäftsleitung

## Neulich im Netz: Biografie einer IT-Pionierin

von Hansjörg Münster

**Schon einmal bin ich in dieser Kolumne auf die Rolle der Frauen in der Frühzeit der IT eingegangen. Aus Anlass zum kürzlich begangenen Frauentag, gibt es hier ein Portrait einer dieser Pionierin: Evelyn Berezin.**

Geboren 1925 in New York hatte sie es schwer ihre Ausbildung voranzutreiben. Als Frau und aus armen Verhältnissen gelang es ihr nicht, an einer staatlichen Universität aufgenommen zu werden. Erst im zweiten Weltkrieg, als die Männer an die Front mussten, gelang es ihr einen Platz in einem Abendkurs zu bekommen und studierte Mathematik, Physik und Chemie. In dieser Zeit arbeitete sie von 8:00 bis 17:00, die Kurse besuchte sie von 18:00 bis 22:00. Die Stunde dazwischen brauchte sie für die Fahrt mit der U-Bahn, in der sie ihre Aufgaben erledigte. 1946 erhielt sie ihren Abschluss in Physik und begann ihre Doktorarbeit mit einem Studium der Teilchenphysik.

Anfangs hatte sie bei ihrer Jobsuche als Physikerin kein Glück, allerdings kam sie mit der jungen Firma „Electronic Computer Corporation“ in Kontakt. Mit den ersten Schaltkreisen und Röhren kannte sie sich aus und wurde als „Logik-Designerin“ eingestellt. 1957 wurde die „Electronic Computer Corporation“, die inzwischen verkauft worden war, geschlossen und alle Mitarbeiter wurden gekündigt.

Bei der Firma „Teleregister“ fand Evelyn einen neuen Job und entwickelte dort einen Computer, der nur mehr Transistoren und keine Röhren mehr verwendete. Als Speicher dienten aber weiterhin Trommeln. Dieser Computer wurde bereits bei Banken, bei der Börse und bei Reservierungssystemen für Airlines verwendet.

Dieses System war für „United Airlines“ 11 Jahre lang im Einsatz und 60 Städte mit dem Server in Denver.

Als nächste wechselte sie zur Firma „Digitronics“, welche von ihren ehemaligen Kollegen bei der „Electronic Computer Corporation“ gegründet wurde. Dort arbeitete sie sich bis zur Vizepräsidentin hoch. Da ihr weitere Karriereschritt nicht möglich waren, machte sie sich selbständig und gründete 1969 ihre eigene Firma: die „Redactron Corporation“.

Berühmt wurde die erste Textverarbeitung – eine Kombination aus Hardware und Software – die diese Firma 1971 unter dem Namen „Data Secretary“ auf den Markt brachte. Es war ein Schrank, in dem der Computer mit seinen 13 Mikrochips verbaut war. Auf dem Schrank war eine elektrische Schreibmaschine (von IBM) verbaut. Die Kapazität lag bei 12 Seiten je Dokument. Das Gerät kostete bis zu 8.000\$ und wurde tausende Male verkauft.

Evelyn verkaufte ihre Firma in der Wirtschaftskrise der 70er Jahre um

25 Million Dollar an die Burroughs Corporation, aus der dann später Unisys hervorging.

Evelyn war 53 Jahre mit ihrem Mann, dem Chemiker Israel Wilenitz verheiratet, den sie während ihrer Studienzeit kennenlernte. Im Jahr 2018 starb Evelyn im Alter von 93 Jahren.■

### Quellen und weiterführende Informationen:

<https://www.golem.de/news/portraet-zum-frauentag-ohne-evelyn-berezin-gaebe-es-keine-textverarbeitung-2103-154732-5.html>

<https://www.computerhistory.org/collections/catalog/102728582>  
[https://de.wikipedia.org/wiki/Evelyn\\_Berezin](https://de.wikipedia.org/wiki/Evelyn_Berezin)

<https://computerhistory.org/profile/evelyn-berezin/>

Ein Ausführliches Interview aus dem Jahre 2014 findet sich Youtube: <https://www.youtube.com/watch?v=3wOWHkX4ilA>



**Hansjörg Münster** ist Principal Consultant und Teamlead bei SEQIS.

Als Allrounder deckt er ein breites Spektrum an Aufgaben ab. Die Schwerpunkte seiner Tätigkeit liegen in den Bereichen Test Management, Testautomation und Lasttest.

Ganz oben auf der Prioritätenliste des IT Profis steht, einen Nutzen und Mehrwert in der Qualitätssicherung seiner IT Projekte zu generieren.



## Aus der SEQIS Sales Ecke

von Sandra Benseler

Liebe Leserin, lieber Leser, ich versuche mit Ihnen ins Gespräch zu kommen, da Sie sich für die Themen interessieren, die wir erfolgreich umsetzen: **IT Analyse, Development, Softwaretest und Projektmanagement.**

... z.B. im Development: Wann immer in Ihrem Unternehmen eine Tätigkeit im Bereich der Software-Entwicklung professionell zu erfüllen ist, Sie kurz- oder auch langfristig skalieren müssen: Bei uns finden Sie die passende Unterstützung!

Konkret kann Ihr Bedarf so aussehen:



Sie sind auf der Suche nach einem internen Mitarbeiter, der Ihr Entwicklungsteam ergänzt. Bis dieser gefunden ist gilt es diese Lücke zu füllen. Jemand muss sich um die laufend anfallenden Tätigkeiten kümmern, damit Ihre Entwicklungen vorankommen.



Sie möchten Ihr Entwicklungsteam fit machen für den agilen Prozess. Am besten gelingt dies, wenn jemand mit Erfahrung das Team begleitet. Doch diese Person mit Erfahrung fehlt Ihnen aktuell.



Sie möchten eine Software-Entwicklung umsetzen. Doch das Entwicklungsteam fehlt Ihnen.

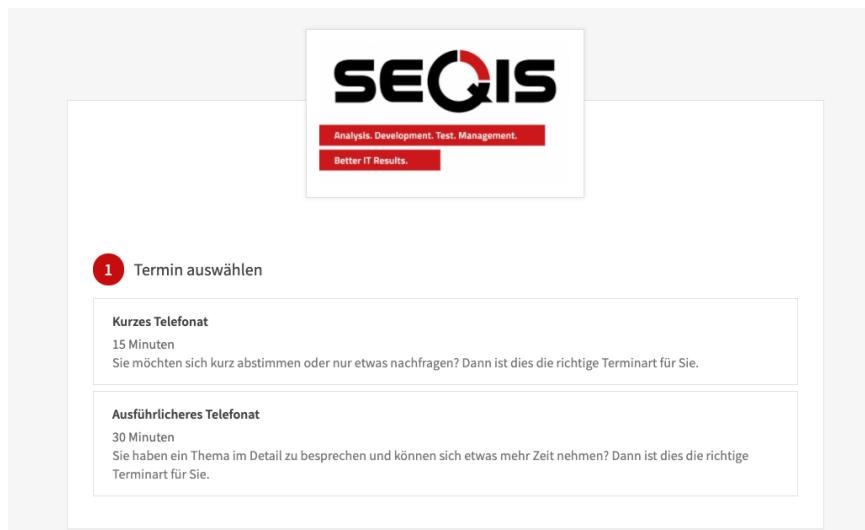


Sie haben veraltete Softwareprodukte im Einsatz. Die Produkte müssen in naher Zukunft neu entwickelt werden, doch Ihnen fehlen die Ressourcen, die sich neben den täglich anfallenden Entwicklungsarbeiten zusätzlich darum kümmern können.

All diese Bedarfe und viele mehr können wir professionell für Sie und mit Ihnen abdecken. **Sie werden sehen, ein Gespräch macht Sinn.**

Rufen Sie mich bitte gerne an +43 664 8378712 oder vereinbaren Sie einen Termin mit mir unter [www.SEQIS.as.me](http://www.SEQIS.as.me) ■

**Wenn Sie mich erreichen möchten, dann scannen Sie einfach den QR Code.**

The screenshot shows the SEQIS website interface. At the top is the SEQIS logo with the tagline 'Analysis. Development. Test. Management. Better IT Results.' Below this is a section titled '1 Termin auswählen' (1. Select appointment). It contains two options: 'Kurzes Telefonat' (15 Minuten) and 'Ausführlicheres Telefonat' (30 Minuten). Each option has a brief description of the service.

Abbildung: SEQIS GmbH



**Sandra Benseler** ist Sales Managerin.

Sie steht Kunden von der Anfrage bis über die Projektlaufzeit hinaus mit Rat und Tat zur Seite.

Ob es sich um Testunterstützung, Beratung, Testtools oder Schulungen handelt – sie hat immer die passende Lösung parat. Im weiteren Verlauf der Geschäftsbeziehung ist sie jederzeit für Kundenanliegen erreichbar. Für Anfragen zu unseren Dienstleistungen, Kundenverträge, Angebotserstellung, Terminkoordination, Lizenzanfragen sowie bei Fragen zum gesamten SEQIS Service Portfolio sind Sie bei ihr genau richtig.

## Native Mobile Apps mit NativeScript-Vue entwickeln

von Michael Dinhof und Leon Palluch

**2022 werden voraussichtlich jeden Tag über 700 Millionen Apps heruntergeladen werden. Mobile Apps punkten gegenüber smartphontauglichen Webseiten unter anderem mit besserer Zugänglichkeit und einem übersichtlicheren Design. Anfang 2020 wurden bereits 52% des Web-Traffics ausschließlich von mobilen Geräten verursacht. Mobile Ansätze zu verfolgen wird für Entwickler also ständig wichtiger.**

Möchte man als Webentwickler ein mobiles Gegenstück für eine bestehende Webapplikation zur Verfügung stellen, kann das mit viel Aufwand verbunden sein. Native Lösungen müssen oft mit unterschiedlichen Technologien für Android und iOS umgesetzt werden. Am Ende hat man einen Webclient und zwei Apps, die einzeln gewartet und erweitert werden müssen, obwohl sie sich kaum voneinander unterscheiden. Es geht aber auch anders.

Mit NativeScript ist plattformunabhängiges Mobile Development auf JavaScript Basis möglich. Bestehende Business Logik kann somit wiederverwendet werden. TypeScript und gewohnte Frontend Frameworks wie Vue.js, Angular und React werden ebenfalls unterstützt.

Zum Umsetzen unserer Apps haben wir uns für die Variante mit Vue und TypeScript entschieden, da wir mit Vue.js bereits vertraut sind. Außerdem ist, verglichen mit dem Implementierungsoverhead anderer Frameworks, Vues übersichtliche Syntax ein Vorteil. Für besonders große und komplexe Applikationen sind unter Umständen andere Frameworks, wie zum Beispiel Angular, besser geeignet.

### Projekt Setup

Zum Aufsetzen eines neuen NativeScript-Vue Projekts ist das Vue CLI zu empfehlen. Sie stellt ein Template bereit, bei dem einige nützliche Voreinstellungen bereits inkludiert sind. So kann man schneller mit der eigentlichen Implementierung beginnen.

Sobald die Projektstruktur vorhanden ist kann man sich die automatisch erstellte „Hello World“ Seite anzeigen lassen. Dazu gibt es die Möglichkeit, mit dem Command „tns preview“ einen QR-Code zu generieren und mit dem eigenen Smartphone einzuscannen (nachdem man 2 Apps installiert hat). Schon hat man die „Vorschau“ der App am Handy.

Diese Funktionalität kommt vom NativeScript Playground, einer

webgehosteten NativeScript Entwicklungsumgebung. Der Playground bietet die Möglichkeit, sich mit der Technologie grundsätzlich vertraut zu machen. Selbst für nur leicht komplexere Projekte ist dieses Tool jedoch zu langsam und unzuverlässig. Für den Entwicklungsprozess ist es ungeeignet, da Änderungen des Sourcecodes teilweise erst nach mehrmaligem Neustart in Kraft treten.

Stattdessen kann die Applikation auf verbundenen Smartphones oder verfügbaren nativen Emulatoren als virtuelles Gerät ausgeführt werden. Dass verbundene Geräte allerdings richtig erkannt werden, ist nicht immer sicher.

Um einen Emulator zu erstellen, bieten sich Programme wie Android Studio an. Wenn das Virtual Device richtig erstellt wurde, sollte es nun dem NativeScript CLI zur Verfügung stehen.



Abbildung: SEQIS „Hello World“ auf Android Virtual Device

Virtuelle Geräte können in unterschiedlichen Auflösungen und mit diversen Android oder iOS Betriebssystemversionen erstellt werden. iOS darf allerdings nur auf Mac OS emuliert werden. Alternativ gibt es kostenpflichtige Cloudlösungen, die Mac OS anbieten. Auf Windows



Abbildung: <https://nativescript.org/community/>

Entwicklungsumgebungen ist nur das Emulieren von virtuellen Geräten für Android erlaubt.

### NativeScript-Vue Komponenten

Nicht nur bei Weboberflächen ist eine komponentenbasierte Architektur ein sinnvoller Ansatz zur Oberflächen-gestaltung. NativeScript-Vue bietet die gewohnte Vue Syntax für Komponenten an. Zwangsweise jedoch mit einigen kleinen Unterschieden.

Im Templateteil der Komponente ist zu beachten, dass statt der bekannten HTML Tags Mobile-spezifische Alternativen eingesetzt werden müssen. Statt `<div>` und `<p>` arbeitet man beispielsweise mit `<StackLayout>`, `<GridLayout>` und `<Label>`. Insbesondere die Layout Tags erlauben ein rascheres Festlegen der groben Seitenstruktur. Nun heißen die Tags zwar anders, aber verwendet werden sie wie gehabt. Attribute lassen sich zuweisen, Property Binding und andere Vue Funktionalitäten wie `v-if` und `v-for` werden ebenfalls unterstützt.

Im Skriptabschnitt hat man die Möglichkeit, bei der Standard Vue Syntax zu bleiben. Mit TypeScript gibt es eine zusätzliche Möglichkeit: Class Components. Hierfür sind vorab Module zu installieren, und neue Decorators zu lernen. Dafür ist die Syntax inklusive Type Declarations übersichtlicher und das Schachteln von Komponenten wird besser unterstützt.

CSS bzw. SCSS ist auch bei NativeScript-Vue für das Styling zuständig, wobei bestimmte Properties wegen der nativen Umgebung nicht verfügbar sind.

### Weitere Ähnlichkeiten und Unterschiede zum Web

Einige nützliche Vue Erweiterungen, wie der Vuex Store sind auch in NativeScript-Vue verfügbar. Mit Vuex können interne Daten und Zustände während der Laufzeit an einer zentralen Stelle abgebildet werden. Bei

Zustandsänderungen besteht dann die Möglichkeit, Aktionen zu setzen. Wird die Projektstruktur über das Vue CLI aufgesetzt, kann beim Einrichten des Projekts angegeben werden, dass Vuex verwendet werden soll.

Nun fehlt noch eine Lösung für das langfristige lokale Speichern von Applikationsdaten. Der gewohnte LocalStorage ist nämlich außerhalb des Browsers nicht verfügbar. Hierfür gibt es aber glücklicherweise eine einfache Lösung. Das NativeScript Core Modul bietet mit „application-settings“ eine Alternative, mit der Strings, Zahlen und Booleans wie mit LocalStorage gesetzt und ausgelesen werden können.

Allgemein fehlen browserspezifische Funktionalitäten, da diese für eine App nicht vorgesehen sind. Dadurch sind beispielsweise Node Packages, die auf dem DOM basieren nicht verwendbar.

Bei der Navigation gibt es mehrere Möglichkeiten. Der Standardrouter von Vue wird zwar an sich nicht unterstützt, es gibt aber einen NativeScript-Vue Router, der in der Verwendung sehr ähnlich ist, wenn man so nah wie möglich an der Webapplikation bleiben möchte. Der Nachteil besteht darin, dass Komponenten bei jedem Seitenwechsel neu geladen werden. Mit der bei NativeScript üblichen Variante werden alle Komponenten sofort geladen. Anzeigt wird dann immer nur die aktive Komponente, während die anderen im Hintergrund weiter bestehen.

Die Verwendung von UI Plugins ist allerdings nicht so angenehm wie beim gewöhnlichen Vue.js. Die Auswahl ist geringer und die Wahrscheinlichkeit hoch, dass das gewählte Plugin nur teilweise oder überhaupt nicht funktioniert. Sogar das Plugin, das bei der offiziellen Anleitung zum Einbinden von Plugins verwendet wird, ist mit der aktuellen Version von NativeScript-Vue nicht kompatibel. Außer-

dem gibt es 3 leicht unterschiedliche Wege Plugins zu registrieren. Es ist also eventuell notwendig durch Trial-and-Error den richtigen zu finden, um überhaupt feststellen zu können, ob das Plugin funktioniert. Wir haben zwar auch vollständig funktionsfähige, regelmäßig upgedatete Plugins gefunden, sie sind aber selten genug, dass es anfangs so ausgesehen hat, als würde die Einbindung generell nicht funktionieren.

### Unit Testing

Das Testen von klassischen Vue Komponenten ist dank der offiziell zur Verfügung gestellten Test Utils kein schwieriges Unterfangen. Besonders viel Dokumentation für Unit Testing mit NativeScript-Vue scheint es zwar nicht zu geben, aber glücklicherweise sind die Test Utils auch für NativeScript Komponenten verwendbar. Hierfür behandelt man diese als wären sie klassische Vue Komponenten (inklusive Importieren von „vue“ statt „nativescript-vue“). Dann gilt es nur noch Vue die nativen XML Tags von NativeScript beizubringen. Hierfür eignet sich die „stub“ Option beim Mounten von Komponenten. Danach lassen sie sich wie gewohnt testen.

### Deployment

Das Deployment einer App kann einerseits über den Google Play Store für Android oder den Apple App Store für iOS Applikationen erfolgen. Zum Testen kann die fertige Applikation auch händisch auf das jeweilige Smartphone übermittelt werden. Dafür wird eine .apk-Datei benötigt, die einfach und intuitiv mit dem CLI generiert werden kann. Diese Datei kann anschließend mit geringem Aufwand auf ein Smartphone übertragen und ausgeführt werden.

### Erfahrungen beim Entwickeln

NativeScript hat den klaren Vorteil mit unterstützen Webframeworks eine echt native App zu liefern. Somit kann bereits bestehender Sourcecode für NativeScript wiederverwendet werden. Das kann für den Entwickler



eine enorme Zeitersparnis bedeuten.

Probleme oder aufwändige Workarounds können jedoch schnell auftreten, wenn man Packages verwendet, die in NativeScript nicht oder anders unterstützt werden. Veraltete oder fehlerhafte Dokumentation für NativeScript stellen ein weiteres Hindernis dar, besonders im Bereich des Softwaretesting. Aufgrund der kleinen Community kann man sich auch nicht darauf verlassen, im Internet für seine Probleme klare Lösungen zu finden.

Diese Umstände verkomplizieren den Entwicklungsprozess erheblich und stellen in Frage, ob NativeScript tatsächlich die beste Lösung ist. Trotz des vielversprechenden Konzepts ist es bezüglich „Quality of Life“ (noch) weit von modernem Webdevelopment entfernt. ■

#### Quellen und weiterführende Informationen:

Information zu App Downloads:  
<https://de.statista.com/statistik/daten/studie/917477/umfrage/anzahl-der-downloads-von-smartphone-apps-weltweit/>

Information zu Webtraffic:  
<https://clearbridgemobile.com/mobile-app-vs-mobile-website-which-is-the-better-option/>



# razzfazz.io

*boost results*

Abbildung: SEQIS GmbH



**Michael Dinhof** ist Agile Development Wizard.

Als Juniorentwickler umfassen seine Aufgabenbereiche Webdevelopment und Softwaretesting. Als frischer Hochschulabsolvent kann er bereits auf eineinhalb Jahre Berufserfahrung und Projekterfahrung im Bereich der Softwareentwicklung zurückblicken. Softwaretesting ist für ihn die Grundlage von qualitativem Code.



**Leon Palluch** ist Agile Development Wizard.

Als Full Stack Developer sammelt er bei SEQIS seine ersten Berufserfahrungen. Von seinem Fachhochschulstudium bringt er ein Grundverständnis für etliche Technologien und Fachbereiche, wie Softwareengineering, Agile Development, User Centered Design und Softwarequalitätssicherung mit.

Eine agile Herangehensweise, ein kooperatives Arbeitsumfeld und gegenseitiger Wissensaustausch sind für ihn ausschlaggebende Merkmale des idealen Development Prozesses.

## Agile Circle Events 2021/22

von Helena Thurner

Der Agile Circle ist der Treffpunkt für UmsetzerInnen digitaler Innovationen.

Hier können sich agile Führungskräfte, Softwareentwicklerinnen und Softwareentwickler, Testerinnen und Tester, Product Owner und Scrum Master treffen, austauschen, inspirieren und neue Ideen entwickeln.

### Unser Programm

Der Agile Circle bringt die unterschiedlichsten Blickwinkel auf die agile Software Entwicklung zusammen: das Management, die Entwicklung, sowie das Testing & die Delivery.

### Agile Circle Events 2021/22



Bild: fizkes/Shutterstock.com

### Agile Circle UNCONFERENCE - Herbst 2021

Für den Herbst 2021 plant der Agile Circle ein lockeres Community Treffen. Programm hier ist ein kurzer Impulsvortrag, gefolgt von einem Lean Coffee und einem World Café.

Seien Sie im Herbst dabei und besprechen gemeinsam mit uns spannende Themen.

#AgileCircleUnconference



**Agile Circle**

Interessiert? Dann besuchen Sie gerne unsere Homepage:  
**[www.agilecircle.org](http://www.agilecircle.org)**



Bild: Rido/Shutterstock.com

### Agile Circle ONLINE - Jänner 2022

Im Jänner 2022 findet zum vierten Mal in Folge der Agile Circle ONLINE statt.

#### Geplanter Ablauf

- Am Vormittag starten wir mit Impulsvorträgen - inkl. Q&A Sessions
- Am Nachmittag folgt eine Podiumsdiskussion
- Abschließend lassen wir den Tag in einer entspannten Atmosphäre mit einer Unconference ausklingen. Hier haben Sie die Möglichkeit, noch offene Fragen zu stellen

#AgileCircleOnline



Bild: Matej Kastelic/Shutterstock.com

### Agile Circle CONFERENCE - 12.05.2022

Im Mai 2022 findet die Agile Circle CONFERENCE in Wien statt.

#### Geplanter Ablauf

- Morgens starten wir mit Workshops zu aktuellen Herausforderungen
- Ab Mittag findet die Konferenz statt: Internationalen Top Keynote Speaker bilden Start und Ende der Konferenz, dazwischen finden zumindest 3 parallele Streams mit Best-Practice-Lösungen statt
- Am Abend wartet auf Sie eine atemberaubende Party im 35. Stock, mit einem wunderschönen 360° Rundumblick über ganz Wien

#AgileCircleConference



**Helena Thurner** ist Marketing Mitarbeiterin.

Von Drucksortengestaltung über Video-dreh und -schnitt, klassischer Pressearbeit, Betreuung der SEQIS Onlinekanäle bis hin zur Organisation von Veranstaltungen übernimmt sie Marketing- und Kommunikationsagenden.

Besonders am Herzen liegt ihr das Schaffen neuer Ideen und Konzepte.

## Visual Studio Code, eine kostenlose IDE revolutioniert die Softwareentwicklung

von Alexander Vukovic

„Es ist 9:00, Markus hat die Installation von Visual Studio, der Enterprise Entwicklungsumgebung von Microsoft, auf seinem Entwicklungsnotebook gestartet. Der Download des 16 GB Images lief über Nacht. Über mehrere Tage, da der Download immer wieder über Nacht abbrach.

Nun läuft endlich die Installation, leider kann Markus sein Notebook damit die nächsten 4h nicht verwenden.“

Solche Erzählungen hört man immer wieder, egal ob es sich dabei um Visual Studio von Microsoft oder XCode von Apple handelt. Diese Entwicklungsumgebungen sind über die Jahrzehnte zu riesigen schwerfälligen Öltankern geworden, die zwar alles können, aber meist auch sehr viel können, was man als Javascript Entwickler nicht benötigt.

Das hat auch Microsoft erkannt und hat Visual Studio Code geschaffen, eine absolut leichtgewichtige Alternative zu Visual Studio. Schnell heruntergeladen und installiert, modular und kostenlos.

### Visual Studio Code ist schnell

Visual Studio Code startet in wenigen Sekunden und reagiert auch bei größeren Codebasen sehr schnell. Klar, denn Visual Studio Code nutzt unter der Haube die gleichen Technologien, die wir bei razzfazz.io als unseren präferierten Stack definiert haben: Javascript, Node, Electron, etc.

D.h. eigentlich ist Visual Studio Code eine Webapplikation, die in einem Electron-Container als Desktop App läuft. Sie ist damit vollständig plattformportabel und leicht erweiterbar.

### Visual Studio Code hat eine extrem große Community

Nach dem Release von Visual Studio Code hat die Entwicklungsumgebung in kürzester Zeit eine riesige Community herausgebildet. Das liegt zum einen an der leichten Erweiterbarkeit und zum anderen daran, dass VSCode kostenlos nutzbar ist.

### Visual Studio Code bietet unzählige kostenlose Erweiterungen

Dadurch gibt es jede Menge Extensions für die unterschiedlichsten Programmiersprachen, die das tägliche

razzfazz.io  
boost results

Abbildung: SEQIS GmbH

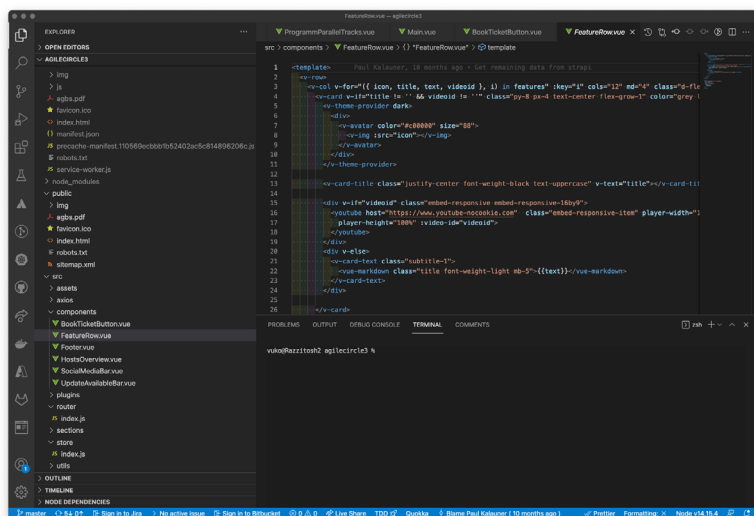


Abbildung: SEQIS GmbH



Arbeiten erleichtern.

Automatische Code Formatierungen, AI-basierte Codevorschläge, Unit Test Integrationen, Code Metriken, es gibt bisher nichts, das wir uns gewünscht haben, was nicht durch eine Extension abzudecken wäre.

Selbst das Syncen der IDE-Konfiguration zwischen Arbeitsplätzen und im Team ist möglich, wodurch alle mit den gleichen Extensions und deren Config arbeiten können.

### Visual Studio Code erleichtert die verteilte Zusammenarbeit

Eines der mächtigsten Features von VSCode ist die Möglichkeit remote gemeinsam am gleichen Sourcecode zu arbeiten. Also tatsächlich Pair Programming z. B. in Zeiten von Corona zu ermöglichen.

Das Liveshare Plugin erlaubt es live direkt im gleichen File zu programmieren, man sieht den Cursor der anderen Teammitglieder und was sie schreiben.

In weiterer Folge können über Liveshare auch Terminalfenster oder ein testweise gestarteter Server gesha-red werden. Also vollständige Remotezusammenarbeit auf ganzer Linie.

### razzfazz.io setzt auf Visual Studio Code

VSCode passt perfekt zu unserem Ansatz bei razzfazz.io: schnell, pragmatisch, effizient und effektiv.

Wenn Sie Unterstützung benötigen, wie Sie Visual Studio Code auch in Ihrem Entwicklungsteam einführen und sinnvoll einsetzen könnten: Wenden Sie sich bitte an uns, wir unterstützen Sie gerne.■

**Kontaktieren Sie uns gerne unter :  
anfrage@razzfazz.io**

## Razzfazz.io: Unsere Leistungen



INDIVIDUELLE TECHNOLOGIEN



INDIVIDUELLE OBERFLÄCHE



INDIVIDUELLE LÖSUNG



REQUIREMENTS ENGINEERING



PERFEKTE QUALITÄTSSICHERUNG



AGILER PROZESS



**Alexander Vukovic** ist SEQIS Gründer und Chief Evangelist.

Er ist erster Ansprechpartner für alle agilen, testmethodischen und test-technischen Anfragen. In der Praxis arbeitet er als Agile Quality Coach, Berater, Interims-Testmanager, CI-Experte und Lasttester. Mehr als 25 Jahre Beratertätigkeit führten ihn während seiner zahlreichen Projekte in die unterschiedlichsten Branchen und Länder.

Sein persönliches Motto „Es gibt keine Probleme, sondern nur nicht gefunden Lösungen“ spiegelt sich in jedem Projekt wider.

## Continuous Deployment Pipeline mit Gitlab

von Klemens Loschy

**Automatisiertes deployment auf verschiedene Umgebungen (meist DEV, TEST, PROD) ist ein essentieller Bestandteil eines funktionierenden Entwicklungsprozesses und mittlerweile auch alles andere als Rocket Science. Und trotzdem ist der initiale Aufbau dann doch mühseliger als erwartet. Wie immer kommt es auf Details an, die es dann doch mitunter spannender machen.**

### Staging bedingt Parametrisierung

URLs, Zugangsdaten, IDs, ... in Applikationen haben wir laufend mit Daten zu tun, die mit der jeweiligen Umgebung abgestimmt sein müssen. Erst bei der Einführung von Staging merkt man eigentlich, wie viele dieser Konstanten dann doch nicht ganz so konstant sind, wie anfangs gedacht und somit pro Umgebung parametrisiert werden müssen.

Unsere Applikation ist mit Jira integriert und liest bzw. manipuliert Eigenschaften (darunter natürlich auch Custom Fields) von Tickets über die Jira eigene REST-API. Custom Fields haben eindeutige IDs, soweit so einfach. Damit unser Staging durchgehend funktioniert gibt es natürlich auch eine eigene Jira Instanz in unserer Testumgebung, initial eine Kopie der Produktion - die IDs aller Custom Fields sind somit über die Umgebungen hinweg ident. Und dann die Überraschung: ein neues Custom Field in Produktion bekommt eine andere ID als in der Testumgebung, und schon haben wir ein weiteren Parameter.

### Wie wird eigentlich parametrisiert?

... das kommt drauf an: Üblich ist eine Mischung aus Command Line Parameter und Umgebungsvariablen. Command Line Parameter ermöglichen es schnell und einfach der

Applikation zum Startzeitpunkt neue, auf die Umgebung abgestimmte, Werte mitzugeben. Gerade bei Applikationen, die oft mit verschiedenen Werten durch einen Benutzer gestartet werden, macht diese Methode durchaus Sinn. Alternativ werden Werte aus den Umgebungsvariablen herausgelesen (die zuvor natürlich auch gesetzt werden müssen). Das passiert oft über den Umweg mit .env Dateien: Darin sind ganz einfache Key=Value Paare hinterlegt, die mittels passendem Framework beim Start der Applikation automatisiert gelesen und als Umgebungsvariable gesetzt werden. Dieses System erlaubt auch ganz einfach eine .env Datei pro Umgebung zu verwalten: .env.test oder .env.prod usw. Aber Achtung: Niemals dürfen Zugangsdaten in diesen Dateien hinterlegt werden, die dann letztendlich im Versionierungssystem landen! Auch wenn z.B. der Git Server selbst gehostet wird, sollte man diese Regel grundsätzlich befolgen.

Wir selbst setzen ausschließlich auf Umgebungsvariablen. Unser UI Framework Vue.js bringt das Management von Umgebungen über den „--mode“ Schalter bereits mit, serverseitig verwenden wir dafür die „custom-env“ Library.

### Staging bedingt Releasemanagement

Eigentlich ist Releasemanagement ja auch schon bei nur einer Umgebung, der Produktion, notwendig: Es muss sichergestellt werden, dass die richtige Version jeder Applikation verfügbar ist, damit die korrekte Kommunikation gewährleistet ist. Es soll ja nicht passieren, dass Applikation A plötzlich nicht mehr mit Applikation B reden kann, weil B die API geändert hat und A nicht mitgezogen ist. Releasemanagement ist in der Produktion schon nicht trivial, in den Umgebungen davor (DEV, TEST) noch deutlich schwieriger: Das Deployment-Intervall ist deutlich kürzer und die Gefahr von Inkompatibilitäten zwischen Applikation sehr hoch. Desto komplexer die Architektur (Stichwort MicroServices) und verteilter die Entwicklung (Stichwort Outsourcing) desto aufwändiger und auch wichtiger ist ein ernsthaft gelebtes Releasemanagement, sonst endet man, bildlich gesprochen, im Turmbau von Babylon.

In unserem aktuellen Projekt haben wir nur vier Teil-Applikation, von denen drei von uns direkt betreut und verantwortlich werden. Dementsprechend leichtgewichtig gestaltet sich auch unser Releasemanagement: Im

```

≡ .env.local-staging
1  NODE_ENV=development
2
3  VUE_APP_BASE_URL=/
4  VUE_APP_SEQITRACKER_SERVER_URL=http://localhost:1338
5  VUE_APP_MOCO_SERVER_URL=https://.mocoapp.com
6  VUE_APP_ADDITIONAL_APP_TITLE=LOCAL@STAGING
7
8  VUE_APP_MOCO_TASKS = [{"urlaub": 6819272, "za": 6819275, "ba": 6819274, "ks": 6819273}]

```

Abbildung: SEQIS, Einfaches .env File für unsere lokale DEV Umgebung

Wesentlichen organisieren wir den Fortschritt über Jira Tickets, die über Verlinkungen die Abhängigkeiten verdeutlichen. So haben wir im Blick, was wann wo deployed sein muss, damit die Applikation als Ganzes richtig funktioniert.

### Deployment @ razzfazz.io

Technologisch setzen wir generell auf Node.js und Vue.js. Das Backend bedingt eine laufende Node.js Applikation, das Frontend kann einfach von jedem beliebigen Web-Server ausgeliefert werden. Als Backend Server haben wir uns für das Service von Heroku (<https://www.heroku.com>) entschieden: Die Heroku Runtime ermöglicht es sehr einfach Applikationen in unterschiedlichsten Technologien zu hosten, in der Minimalausführung sogar for free. Automatisiert wird alles über Gitlab CI/CD.

Auch hier kommen wieder die Umgebungsvariablen ins Spiel, in Gitlab „Variables“. Damit kann die Applikation zur Build Time parametrisiert werden. Nach dem Deployment wird der aktuelle Code-Stand im SCM noch getagged und ein Release erstellt. Das alles passiert komplett automatisiert und wird per push in den jeweiligen Branch angestoßen. Hier noch ein Tipp: Vor dem push nach „master“ oder „staging“ werden lokal, mit husky automatisiert, alle Tests durchgeführt. Das hat uns schon oft vor fehlgeschlagenen Builds bewahrt.

### Server @ Heroku

Heroku unterstützt diverse automatisierte Deployment Varianten: anfangs haben wir noch manuell den aktuellen Stand in den „master“ branch des Heroku eigenen Git Repositories gepusht, aktuell verwenden wir das

den Start bis zu Ende ab und verifiziert den Status (pending, succeeded, failed), so kann man sich sicher sein, dass die Applikation auch tatsächlich korrekt gestartet werden konnte. Für die umgebungsabhängigen Eigenschaften gib es in Heroku die so genannten „Config Vars“, die als Umgebungsvariablen der Applikation zur Verfügung gestellt werden. Damit lässt sich das für uns notwendige Staging Konzept einfach umsetzen. Nach dem Deployment ist die App unter „<https://appname.herokuapp.com>“ verfügbar. Angenehm ist auch das automatische SSL Certificate handling: Heroku erstellt automatisch Let's Encrypt Zertifikate für die jeweilige APP, terminiert die SSL Session und leitet den traffic dann unverschlüsselt an die Applikation weiter. In der Applikation selbst muss man sich dadurch mit https oder Zertifikaten gar nicht mehr herumschlagen.

**razzfazz.io**  
*boost results*

Abbildung: SEQIS GmbH

### Build mit Gitlab CI/CD

Gitlab selbst stellt einen Continuous Integration Service zur Verfügung, das über das .gitlab-ci.yml File konfiguriert wird. Damit steuern wir den Build, den Test und das Deployment in die jeweilige Umgebung: unser „master“ branch wird auf die PROD Umgebung deployed, „staging“ auf die TEST. Davor werden natürlich alle Tests durchgeführt und die Testabdeckung berechnet.

Deployment „Tool“ Dpl (<https://github.com/travis-ci/dpl>). Dpl abstrahiert Deploymenteigenheiten diverser Hosting Anbieter (uA. natürlich AWS, GAE, OpenShift, ...) und reduziert die Komplexität auf einen einzigen einfachen command line Aufruf von Dpl. Im Hintergrund verwendet Dpl die Heroku eigene REST API um die Applikation zu stoppen, den aktualisierten Stand der Applikation hochzuladen und danach die Applikation wieder zu starten. Dabei wartet Dpl

### Client @ SSD Webhosting

Für unseren Web-Client mit Vue.js verwenden wir unseren bereits vorhandenen Webhosting Provider SSD-Webhosting, der liefert alle statischen Ressourcen an den Browser aus. Ein Application Server ist nicht notwendig. Das erleichtert das Deployment und wir können am Ende des Gitlab Builds den aktuellen Stand einfach per SFTP auf den Server kopieren, mehr ist für ein Client Deployment nicht notwendig.

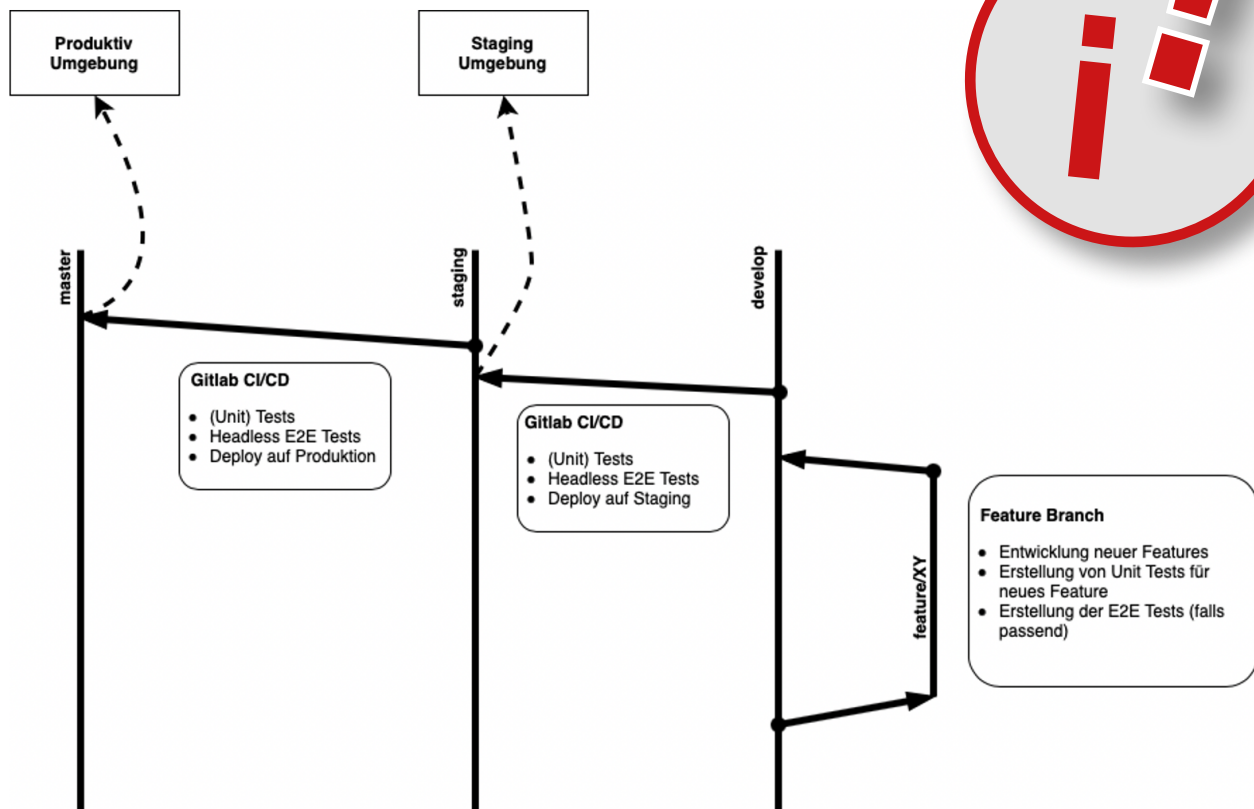


Pipeline Needs **Jobs 3** Tests 124

Status	Job ID	Name	Coverage
Test			
passed	#1055654184	test	98.92%

Abbildung: SEQIS, 124 Tests decken fast 99% des Server Codes ab





## Wrap Up

... und ganz ehrlich: ich freu mich jedes Mal wieder, wenn ein Deployment in die Produktion genau nur ein Knopfdruck ist! ■



## SEQIS: Ten more things

von Helena Thurner



### Ten more things

Online. Proven. Professional.

- ✓ Praxisbezogene Fachvorträge zu aktuellen IT-Trendthemen
- ✓ Kompetentes, fachliches Know-how aus der Praxis
- ✓ 10 Tipps & Tricks für Ihren Arbeitsalltag
- ✓ Kostenlose Teilnahme
- ✓ Online

... was bringt 2021? Wir starten unsere **Online** Eventreihe „**Ten more things**“!

Holen Sie sich Wissenswertes rund um top aktuelle Themen aus den Bereichen IT Analyse, Development, Softwaretest und Projektmanagement. Bei unseren „Ten more things“ erhalten Sie jeweils in rund 50 Minuten mindestens **10 Tipps zum Thema aus unserer Projektpraxis**. Dh. alle 5 Minuten ein „One more thing“, wie Steve Jobs bei seinen Produktpräsentationen so gerne top news eingeleitet hat. Nach diesem Vortrag haben Sie die Möglichkeit Ihre Fragen den Expertinnen und Experten zu stellen.

Unsere Sessions starteten mit vollem Erfolg! Klemens Loschy begeisterte unsere Teilnehmerinnen und Teilnehmer mit dem Thema Last- und Performancetest. Alexander Vukovic

stellte das Thema Software Escrow vor. Sie interessieren sich auch für diese Themen, konnten jedoch nicht bei den Vorträgen dabei sein? Wir haben die Lösung für Sie! Die gezeigten Folien und das aufgezeichnete Video finden Sie auf unserer Website. [www.seqis.com/de/events-index](http://www.seqis.com/de/events-index) Oder scannen Sie den QR Code.



#### Details und Anmeldung

Sie sind an unseren Ten more things Vorträgen interessiert? Dann nehmen Sie sich Zeit und melden Sie sich ganz einfach auf unserer Website [www.SEQIS.com/events-index](http://www.SEQIS.com/events-index) an. Die Teilnahme findet online statt und ist kostenlos, aber die Anzahl der Plätze ist limitiert.

Holen Sie sich Ihren kostenlosen Wissensvorsprung und innovative Lösungen für Ihren Erfolg! ■

Spannend geht es auch mit unserem Ten more things Programm weiter.

Melanie Gau	Gamification	16.09.2021 🕒 16:00 - 17:10	
Stefan Ladstätter	UX und Usability	14.10.2021 🕒 16:00 - 17:10	
Markus Schwabeneder	KI und Machine Learning	11.11.2021 🕒 16:00 - 17:10	



**Helena Thurner** ist Marketing Mitarbeiterin.

Von Drucksortengestaltung über Videodreh und -schnitt, klassischer Pressearbeit, Betreuung der SEQIS Onlinekanäle bis hin zur Organisation von Veranstaltungen übernimmt sie Marketing- und Kommunikationsagenden.

Besonders am Herzen liegt ihr das Schaffen neuer Ideen und Konzepte.

Die nächsten  
Termine im  
Überblick:



Ten more **things**

### Gamification

**16. September 2021**

Weitere Informationen finden  
Sie auf **Seite 15** und auf  
unserer Website.  
[www.SEQIS.com/events-index](http://www.SEQIS.com/events-index)



Haben Sie schon Ihre  
nächste Weiterbildung  
geplant?

[www.SEQIS.com](http://www.SEQIS.com)

### August

<b>1 So</b>
<b>2 Mo</b>
<b>3 Di</b>
<b>4 Mi</b>
<b>5 Do</b>
<b>6 Fr</b>
<b>7 Sa</b>
<b>8 So</b>
<b>9 Mo</b>
<b>10 Di</b>
<b>11 Mi</b>
<b>12 Do</b>
<b>13 Fr</b>
<b>14 Sa</b>
<b>15 So</b> Mariä Himmelfahrt
<b>16 Mo</b>
<b>17 Di</b>
<b>18 Mi</b>
<b>19 Do</b>
<b>20 Fr</b>
<b>21 Sa</b>
<b>22 So</b>
<b>23 Mo</b>
<b>24 Di</b>
<b>25 Mi</b>
<b>26 Do</b>
<b>27 Fr</b>
<b>28 Sa</b>
<b>29 So</b>
<b>30 Mo</b>
<b>31 Di</b>

### September

<b>1 Mi</b>
<b>2 Do</b>
<b>3 Fr</b>
<b>4 Sa</b>
<b>5 So</b>
<b>6 Mo</b>
<b>7 Di</b>
<b>8 Mi</b>
<b>9 Do</b>
<b>10 Fr</b>
<b>11 Sa</b>
<b>12 So</b>
<b>13 Mo</b>
<b>14 Di</b>
<b>15 Mi</b>
<b>16 Do</b> Ten more things Gamification
<b>17 Fr</b>
<b>18 Sa</b>
<b>19 So</b>
<b>20 Mo</b>
<b>21 Di</b>
<b>22 Mi</b>
<b>23 Do</b>
<b>24 Fr</b>
<b>25 Sa</b>
<b>26 So</b>
<b>27 Mo</b>
<b>28 Di</b>
<b>29 Mi</b>
<b>30 Do</b>



Oktober	
1 Fr	
2 Sa	
3 So	
4 Mo	
5 Di	
6 Mi	
7 Do	
8 Fr	
9 Sa	
10 So	
11 Mo	
12 Di	
13 Mi	
14 Do	Ten more things UX und Usability
15 Fr	
16 Sa	
17 So	
18 Mo	
19 Di	
20 Mi	
21 Do	
22 Fr	
23 Sa	
24 So	
25 Mo	
26 Di	Nationalfeiertag
27 Mi	
28 Do	
29 Fr	
30 Sa	
31 So	

November	
1 Mo	Allerheiligen
2 Di	
3 Mi	
4 Do	
5 Fr	
6 Sa	
7 So	
8 Mo	
9 Di	
10 Mi	
11 Do	Ten more things Skynet ante portas?
12 Fr	
13 Sa	
14 So	
15 Mo	
16 Di	
17 Mi	
18 Do	
19 Fr	
20 Sa	
21 So	
22 Mo	
23 Di	
24 Mi	
25 Do	
26 Fr	
27 Sa	
28 So	
29 Mo	
30 Di	

Die nächsten Termine im Überblick:



Ten more things

UX und Usability  
14. Oktober 2021



Skynet ante portas?  
11. November 2021



Weitere Informationen finden Sie auf [Seite 15](#) und auf unserer Website.  
[www.SEQIS.com/events-index](http://www.SEQIS.com/events-index)



## Interview mit Mag. Amir Moussa Managing Director

Datum: 16. Februar 2021

Ort: Mödling, Niederösterreich

### Über Peacequare GmbH:

Seit der Gründung 2014 ist Peacequare der Impulsgeber bei der Entwicklung webbasierter Applikationen im medizinischen Bereich. Mit Care01 konnten wir das erste webbasierte Praxisverwaltungssystem auf den österreichischen Markt bringen und somit den Bedarf nach Flexibilität, Mobilität und Datensicherheit abdecken. Den Praxisalltag durch leistungsstarke Anwendungen zu unterstützen, sehen wir als unsere wichtigste Aufgabe. Wir nutzen Technologie, um unsere Applikationen und den Service für den medizinischen Bereich stetig weiter zu entwickeln. So ist Care01 die erste und einzige webbasierte Anwendung, die medizinische Geräte einbinden kann sowie den Anforderungen von Kassenärzten gerecht wird.

### Spotlight - SEQIS Kunden im Rampenlicht

Unsere Kunden stehen bei all unseren Aktivitäten im Fokus. Egal welche Branche, SEQIS unterstützt Ihre IT Projekte. Eine erfolgreiche Realisierung von Analyse-, Entwicklungs-, Test- und Managementprojekten sowie all den zugehörigen Tätigkeiten in diesen Bereichen ist mit SEQIS Unterstützung gewährleistet.

In der Rubrik Spotlight stellen wir regelmäßig unsere Kunden auch in den SEQIS QualityNews in den Vordergrund und geben dadurch einen Einblick in unsere unterschiedlichen Projekte und Dienstleistungen.

### Interview mit Mag. Amir Moussa, Managing Director

**Dominik Tuchschrnmidt, SEQIS:** Sehr geehrter Herr Moussa, seit dem letzten Jahr gab es in den Bereichen Softwareentwicklung immer wieder Themen, die Sie gemeinsam mit SEQIS erfolgreich umgesetzt haben. Aktuell unterstützt SEQIS Sie in der Softwareentwicklung. Welche Erwartungen setzen Sie in die Zusammenarbeit mit SEQIS und in welcher Weise werden Ihre Erwartungen erfüllt?

**Amir Moussa, Peacequare GmbH:** Besonders wichtig für unsere Softwareentwicklung ist die rasche Umsetzung und Anforderung unter Einhaltung unserer Richtlinien. Essenziell für uns ist auch die Lieferung eines Codes mit einer hohen Codequalität.

**Dominik Tuchschrnmidt, SEQIS:** Könnten Sie das technische Umfeld der Aufgabenstellung im Bereich Software Entwicklung nochmals kurz skizzieren?

**Amir Moussa, Peacequare GmbH:** Care01 ist eine Browserapplikation basierend auf Node.js. Das zugehörige Backend ist ein Server, ebenfalls auf Node.js Basis mit MSSQL Persistenz gehostet in der Azure Cloud. Eine besondere Spezialität ist, dass wir am Client sukzessive von der Component Library Kendo UI auf AngularJS migrieren, d.h. aktuell werden beide Technologien in Kombination eingesetzt, was zu zahlreichen Herausfor-



**PEACEQUARE**

**care01**  
The Mobile Practice

derungen und kreativen Lösungen führt.

**Dominik Tuchschrnmidt, SEQIS:** Ihre Services werden national von vielen Ärzten sowie Gesundheitseinrichtungen genutzt, und die Qualität Ihrer Produkte ist Ihnen ein besonderes Anliegen. Was möchten Sie als besondere Herausforderung im Detail herausstreichen?

**Amir Moussa, Peacequare GmbH:** Mit Care01 wollen wir alle Ärzte bei ihrer täglichen Arbeit bestmöglich unterstützen. Care01 deckt daher sehr viele Anforderungen der verschiedenen ärztlichen Fachgebiete ab. Sie können sich vorstellen, dass das eine ganze Menge ist. Die Qualitätssicherung dieser breiten Funktionalität ist daher für uns eine besondere Herausforderung, die wir aber bisher gut im Griff haben.

**Dominik Tuchschrnmidt, SEQIS:** Unser Motto bei razzfazz.io ist ja „Boost Results“, inwieweit konnten wir das

bei Ihnen umsetzen, also Ihren Ergebnissen einen Boost geben?

**Amir Moussa, Peacequare GmbH:**

Die Unterstützung war schnell und unkompliziert. Die beiden Kollegen konnten sich rasch in die Materie einarbeiten und damit bald echten Mehrwert stiften. Die Lieferungen waren qualitativ hochwertig und konnten damit einfach in Care01 integriert werden.

**Dominik Tuchschiidt, SEQIS:** Gibt es einen Aspekt in der Zusammenarbeit mit SEQIS, der Sie besonders beeindruckt hat, bzw. etwas, das Sie als wesentlichen Nutzen für Ihr Unternehmen im Rahmen der Zusammenarbeit empfinden?

**Amir Moussa, Peacequare GmbH:**

Die Selbstständigkeit bei der Unterstützung hat auf unserer Seite kaum Mehraufwand bedeutet, d.h. keine langen Abstimmungen oder Erklärungen waren notwendig. Bei kurzer und punktueller Unterstützung ist einfach keine Zeit für langes und aufwendiges Onboarding.

**Ausführende Firma: razzfazz.io**

razzfazz.io ist eine Marke der SEQIS GmbH für individuelle Softwareentwicklung!

Wir wickeln Projekte agil und unter starker regelmäßiger Einbindung des Kundenfeedbacks ab. Dabei setzen wir auf eine fundierte Business Analyse und Requirement Engineering, sodass für den Kunden von Anfang an klar ist, welche Lösungen er in den kommenden Iterationen übergeben bekommt. Klarerweise entwickeln sich auch die Anforderungen über die Projektdauer weiter, was im agilen Vorgehensmodell von razzfazz.io ein unterstützter und gewünschter Weg zu einer besseren Lösung für den Kunden ist.

Unsere Wurzeln sind auch in der Qualitätssicherung und dem Wissen, dass wir im Zuge der Beratung und Testunterstützung vieler Entwicklerteams gewonnen haben. Wir wissen, wie man richtig Qualität von Anfang an in der Software verankert. Unit Tests, Test Automation, Last- und Performancetests, Usability und Securitytests sind integraler Bestandteil unserer Entwicklungsprojekte. ■



[www.razzfazz.io](http://www.razzfazz.io)

**razzfazz.io**  
*boost results*

Abbildung: SEQIS GmbH

Das Interview mit Herrn Amir Moussa finden Sie auch auf unserer Webseite:



**Dominik Tuchschiidt** ist Sales Mitarbeiter.

Als Allrounder deckt er ein breites Spektrum an Aufgaben ab. Der Schwerpunkt seiner Tätigkeit liegt im Bereich der Neukunden Akquise für die Geschäftsfelder: Analyse, Development, Test und Management.

Ganz oben auf der Prioritätenliste des Sales Profis steht, einen Nutzen und Mehrwert in den IT Projekten seiner Partner zu generieren.

## Eine kurze Vorstellung von R

von Claudia Amon



Abbildung: Hadley Wickham and others at RStudio ([https://commons.wikimedia.org/wiki/File:R\\_logo.svg](https://commons.wikimedia.org/wiki/File:R_logo.svg)), <https://creativecommons.org/licenses/by-sa/4.0/legalcode>, keine Änderungen

Kennen Sie R? Sie ist eine der bekanntesten Programmiersprachen<sup>1</sup> und trotzdem sind nur wenige damit vertraut. R ist der Name der Programmiersprache – und auch der der Software. Sie wurde Anfang der 1990er entwickelt und wird für statistische Berechnungen und Grafiken eingesetzt. Unter den Bedingungen der GNU General Public License (GPL) kann sie kostenfrei eingesetzt werden. Da die Benutzeroberfläche von R eingeschränkt ist, bietet es sich an, eine externe Benutzeroberfläche zu installieren. Dafür gibt es mehrere Möglichkeiten. Meine Empfehlung ist RStudio<sup>2</sup>, es ist ebenfalls kostenlos verfügbar und leicht bedienbar. Die Syntax von R ist relativ einfach. Den Variablen muss kein Datentyp, wie int oder char, zugewiesen werden, R bestimmt den Typ selbst. Hingegen bezeichnen a und A verschiedenartige Symbole und beziehen sich somit auf unterschiedliche Variablen.

Solch eine Variable kann aus einem oder auch aus mehreren Buchstaben bestehen. Einige Wörter („Keywords“) dürfen jedoch nicht als Variablennamen genutzt werden z.B. function, for, if, else, while und TRUE. Das Programm kann als Taschenrechner verwendet werden. Möchte man das errechnete Ergebnis künftig weiterverwenden, muss man diesen Wert einer Variable zuweisen.

```
Beispiel:
x <- 2+3
y <- 4
z <- x*y
print(z)
```

```
Output:
>20
```

*To understand computations in R, two slogans are helpful: Everything that exists is an object. Everything that happens is a function call.*  
John M. Chambers<sup>3</sup>

John M. Chambers, ein Mitentwickler von R, erklärt in seinem Paper<sup>3</sup>, „Um Berechnungen in R zu verstehen sind zwei Sätze hilfreich: Alles, was existiert, ist ein Objekt. Alles, was passiert, ist ein Funktionsaufruf.“ Das heißt, Berechnungen, die komplexer als eine Konstante oder ein Name sind, werden als Funktionsaufruf durch R verarbeitet.

R enthält Basispakete, die die Standardfunktionen, grundlegende Statistik- und Grafikfunktionen, sowie auch Beispieldatensätze zum Ausprobieren beinhalten. Der Funktionsumfang des Programms kann erweitert werden, um etwa Zugriff zu speziellen Daten oder außergewöhnlicheren statistischen Methoden zu erhalten. Zudem existieren packages, die Schnittstellen zu Datenbanken (RMySQL), anderen Programmiersprachen (rPython) oder Webservices (RSelenium) liefern. Diese müssen manuell installiert werden.

Eines dieser Pakete stelle ich folgend näher vor, und zwar testthat. Es funktioniert wie ein Unittest, d.h. es erleichtert das Testen eines Codes, wenn man unter anderem Änderungen an ihm vornimmt. Genauer gesagt, wird es verwendet, um zu überprüfen, ob der geschriebene Code auch das richtige Resultat liefert.





```
Ein einfaches Beispiel:
test_that("Quadratwurzel wird richtig berechnet", {
  expect_equal(sqrt(4),3)
  expect_equal(sqrt(4),2)
})
```

```
Dadurch wird folgender Output erzeugt:
Error: Test failed: Quadratwurzel wird richtig berechnet'
* sqrt(4) not equal to 3.
1/1 mismatches
[1] 2 - 3 == -1
```

Dieser Test besteht aus einem Testnamen und den zu testenden Code zuzüglich seines erwarteten Ergebnisses. Ein Vorteil dieses packages ist, dass, wenn der Test fehlschlägt, man am Output die Ursache erkennen kann, wie auch am Beispiel zu erkennen ist. Die 2. Zeile liefert keinen Output, da sie korrekt ist, das Ergebnis entspricht dem erwarteten Wert.

Hilfe zu Paketen, Funktionen oder Operatoren erhält man direkt in R durch Eintippen eines Kommandos. Mittels `help.start()` öffnet sich die HTML Version von R's Onlinedokumentation. Benötigt man Hilfe bzgl. eines speziellen Befehls, beispielsweise wie man einen Vektor erzeugt, erhält man diese anhand der Eingabe von `help(vector)` oder `?vector`.

Zusammengefasst ist R als Open-Source-Software, sowie auch als Programmiersprache, sehr vielseitig. Zugleich werden keine tiefergreifenden Programmierkenntnisse benötigt und bei Schwierigkeiten erhält man viele Lösungsansätze im Internet. Für jeden, der sich für Berechnungen und Datenanalyse interessiert, ist R klar zu empfehlen. ■



#### Literatur:

- <sup>1</sup> <https://pypl.github.io/PYPL.html>
- <sup>2</sup> <https://rstudio.com/>
- <sup>3</sup> John M. Chambers (2014); Object-Oriented Programming, Functional Programming and R. Statistical Science 2014, Vol. 29, No. 2, pp. 167–180.

#### Weiterführende Literatur:

Hadley Wickham (2011); *testthat: Get Started with Testing*. The R Journal, vol. 3, no. 1, pp. 5–10.

Daniel Wollschläger (2017); *Grundlagen der Datenanalyse mit R*. Berlin: Springer, 4. Auflage.

Weitere spannende  
Blogartikel finden Sie  
hier:

[https://www.seqis.com/  
de/blog-index](https://www.seqis.com/de/blog-index)



**Claudia Amon** ist Consultant.

Durch ihr Studium entdeckte sie das Interesse an Programmiersprachen. Sie ist vorwiegend in den Bereichen Testfallerstellung, Testdurchführung, sowie auch Testunterstützung tätig.

Klare Anforderungen und eine gewissenhafte Arbeitsweise sind ihr wichtig, um eine hochwertige Softwarequalität zu erhalten.

## Model-Driven Development

von Daniel Kiesenhofer

Mit stetig steigender Komplexität von IT-Projekten sind Modelle ein wesentlicher Faktor, um ein gemeinsames Verständnis der zu entwickelten Software zu bilden – sowohl für Entwickler als auch Fachabteilungen. Denn anders als Code sind Modelle auch für Projektbeteiligte ohne Programmiererfahrung lesbar. Der Model-Driven Development (= MDD) Ansatz macht sich diesen wesentlichen Vorteil zu Nutze. Dabei werden Softwaremodelle ins Zentrum der Entwicklung gehoben und Codegeneratoren, um auf Basis der Modelle Teile der Software automatisiert zu erzeugen, verwendet.

### Ist MDD für jede Art von Projekt geeignet?

Nein, es eignet sich vor allem für komplexe Software mit vielen an der Entwicklung beteiligten Stakeholdern. Denn durch den höheren Detailgrad an zu entwickelnden Code steigt auch der Mehrwert von Modellen, welche durch Abstraktion diesen Detailgrad verringern und nur ein vereinfachtes Bild der Realität abbilden. Zudem eignet sich MDD sehr gut in Projekten bei denen die Funktionalität laufend erweitert wird. Für das Hinzufügen einer neuen Programmschnittstelle muss beispielsweise nur ein neues Schnittstellenmodell erstellt werden. Der Code für die Realisierung der Schnittstelle wird dann auf Basis dieses Modells automatisch erzeugt. Bei kleineren Projekten übersteigt jedoch der zusätzliche Entwicklungsaufwand den Nutzen und MDD könnte zu einem Overhead werden.



### Step by Step Umsetzung

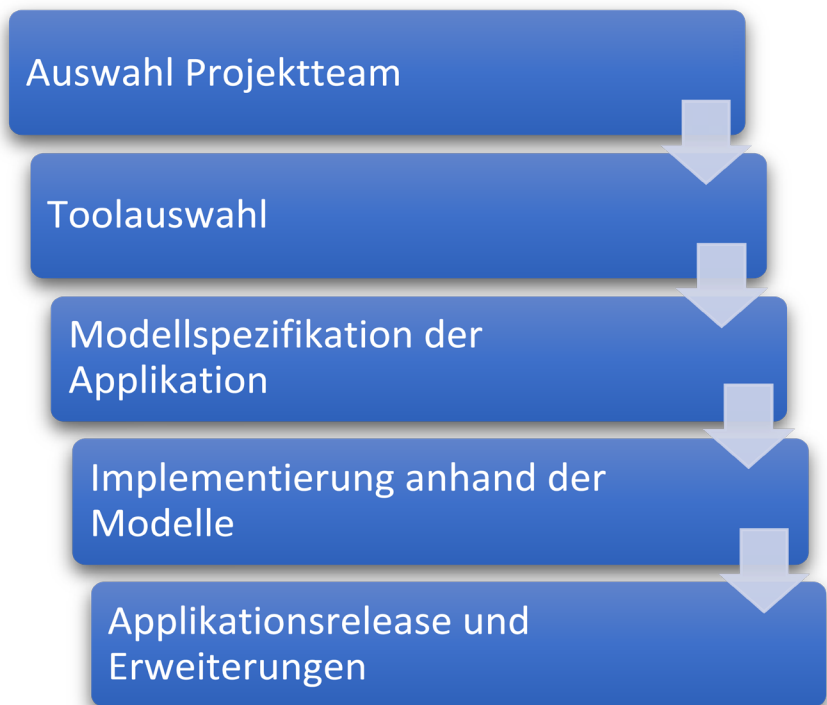


Abbildung: SEQIS GmbH

#### 1. Auswahl Projektteam

Vor dem Projektstart ist die Auswahl des Entwicklungsteams ein wesentlicher Erfolgsfaktor. Denn MDD ist kein Selbstläufer. Es erfordert zumindest einen MDD-erfahrenen Entwickler, der den Rest des Teams in Bezug auf Methodik und Tools coacht. Auch ein guter Kontakt zu den Fachabteilungen ist wichtig, denn diese stellen domänenspezifisches Wissen bereit, welches für die Modellbildung benötigt wird.

#### 2. Toolauswahl

Um die Modelle bestmöglich in den Entwicklungsprozess einzugliedern und daraus Code zu generieren, müssen dafür die richtigen Tools ausgewählt werden. Ein in diesem Bereich verbreitetes Tool ist das Modelling Framework von Eclipse (EMF). Dieses macht Gebrauch von etablierten

Modellierungssprachen wie UML um die zu entwickelnde Applikation in einem zusammenhängenden Model abzubilden, dem sogenannten ECore Model. Die im EMF-Framework integrierten Codegeneratoren können im Anschluss per Mausklick den modellbasierten Applikationscode erzeugen.

#### 3. Abstimmung und Spezifikation

Ein weiteres Erfolgskriterium ist die Kommunikation zwischen Fachbereich und Entwicklung. Für die Modellerstellung ist beides erforderlich: Domänenspezifisches und technisches Wissen. Und nur durch gute Abstimmung kann daraus ein harmonisches Softwarekonzept entstehen.

#### 4. Implementierung

Sobald die Modelle in einem release-fertigen Zustand sind, kann davon der Softwarecode erzeugt werden.

Dieser reicht je nach Detailgrad der Modelle von einem Grundgerüst der Programmarchitektur bis zu einer vollständigen Implementierung der Software. Üblicherweise sind jedoch nicht alle Codeteile durch Modelle abbildbar bzw. ist dies aus Kostengründen nicht vertretbar. Als Entscheidungskriterien dafür dienen Wiederverwendbarkeit von Code, Time-To-Market von Releases und Softwarequalität. Je wichtiger diese sind, desto sinnvoller eignet sich die modellbasierte Implementierung.

lierung und man sollte daher immer abwägen, ob dies vorteilhafter als eine manuelle Implementierung ist.

### Fazit

Der MDD-Ansatz kann die Komplexität von großen Softwareprojekten wesentlich reduzieren und so zu einem besseren Gesamtverständnis aller Stakeholder beitragen. Dies resultiert in einer guten Kommunikation und Umsetzung der Anforderungen sowie Kontrolle, ob diese im System umgesetzt wurden. Bei kleinen

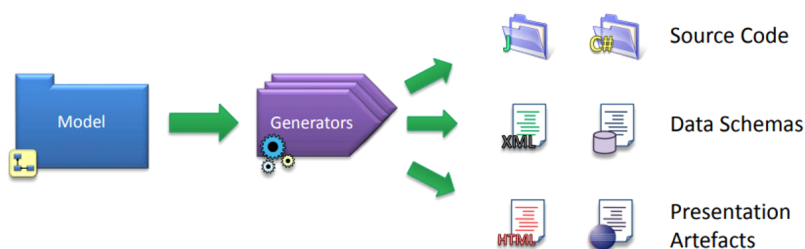


Abbildung: [https://wiki.eclipse.org/images/8/8d/Using\\_MDD\\_Eclipse\\_Technology\\_to\\_implement\\_SOA.pdf](https://wiki.eclipse.org/images/8/8d/Using_MDD_Eclipse_Technology_to_implement_SOA.pdf)

Parameter, welche dagegen sprechen, sind Geschäftslogiken wie etwa mathematische Berechnungen, welche nicht sinnvoll modelliert werden können oder einfachere, änderungsresistente Programmteile wie eine Loginmaske. Die letztendliche Entscheidung über die Sinnhaftigkeit sollte gemeinsam durch MDD-Experten und Management getroffen werden. Dann folgt nach Erzeugung des Grundgerüsts (auch Skeleton genannt) eine nachfolgende Implementierung der fehlenden Geschäftslogik. MDD eignet sich auch für agile Entwicklungsmethoden, da iterative Erweiterungen in den Modellen auch immer automatisch die entsprechenden Programmteile erweitern.

### 5. Erweiterung

Mit steigender Anzahl der zu entwickelnden Systemkomponenten steigt der Vorteil von MDD, da Modelle bzw. deren Konzepte wiederverwendet werden können. Das Hinzufügen von neuen Schnittstellen oder Klassen ist daher sehr einfach. Komplett neue Programmlogiken hingegen erfordern initialen Mehraufwand bei der Model-

Projekten steht jedoch der zusätzliche Aufwand für die Modellierung und Tooleinbettung in keinem guten Verhältnis zum zusätzlichen Nutzen und es soll daher auf andere Entwicklungsmodelle ausgewichen werden. ■

### MDD Gegenüberstellung

#### Vorteile:

- Steigende Qualität durch verbessertes Softwareverständnis und Zusammenarbeit aller Stakeholder
- Reduktion von Entwicklungszeit durch Codegenerierung
- Bessere Wiederverwendbarkeit aufgrund adaptierbarer Modelle
- Up-To-Date Modelle bieten gute Projektdokumentation

#### Nachteile:

- Zusätzliches Wissen über Modelle und Tools erforderlich
- Modelle können oft nicht die gesamte Funktionalität der Software darstellen
- Es kann oft nur ein Grundgerüst der zu entwickelnden Software automatisiert auf Basis der Modelle generiert werden



**Daniel Kiesenhofer** ist Consultant.

Zu seinen Projekterfahrungen zählen Einsätze in den Bereichen Testautomatisierung und Testdurchführung. Eine gute Integration von Softwaretests in den Entwicklungsprozess ist für ihn ein Hauptanliegen für effiziente und erfolgreiche Softwareentwicklung. Er sieht zudem Qualitätssicherung als wesentlichen Beitrag zur Erreichung von Kundenzufriedenheit.

## Entwicklung von Responsive Web Apps: Welches Framework sollte man nutzen?

von Markus Schwabeneder

Wie immer in der Software-Entwicklung stellt sich schnell die Frage, auf welche Technologien bzw. Frameworks man setzen sollte. Der Frage gehe ich in diesem Artikel nach. Damit hier der Rahmen nicht völlig gesprengt wird, konzentriere ich mich auf die drei meist verbreitetsten, auf JavaScript aufbauenden, Frameworks, namentlich Angular, React und Vue.js. Ich werde am Ende eine einfache Entscheidungshilfe anbieten, in der natürlich auch meine subjektive, aber faktenbasierte Meinung einfließt.

Zuerst aber werde ich kurz auf die Begrifflichkeiten eingehen und auch eine Übersicht über die Unterschiede und Eigenschaften dieser Frameworks geben.

### Progressive Web App

Eine Progressive Web App (= PWA) ist eigentlich einfach eine Website mit der Besonderheit, dass sie, wenn man auf sie mit einem mobilen Endgerät zugreift, viele Funktionen bietet, die früher nur native Apps hatten. So kann man eine PWA unter anderem auch "installieren", also sie als Icon auf dem Homescreen anzeigen oder mittels "Push Notifications" Nachrichten an den User schicken, auch wenn er die App gerade nicht benutzt.

#### Vorteile

- Es reicht eine gemeinsame Website für alle Plattformen zu programmieren, man muss keine zusätzlichen Apps für iOS und Android schreiben.
- Man ist nicht an die Stores von Apple oder Google gebunden, man kann auch Microtransactions durchführen, ohne einen Obolus an Dritte entrichten zu müssen.

#### Nachteile

- Man kann die Stores von Apple oder Google nicht zum Verteilen nutzen.
- Man ist an die Kompatibilität des auf dem Gerät installierten Browsers gebunden, Chrome und Firefox bieten derzeit eine 100%-Unterstützung an, Safari als Browser bedeutet noch diverse Einschränkungen.

### Angular

#### Eigenschaften

- "Vater" dieses Frameworks ist das noch immer verbreitete, aber nicht mehr weiterentwickelte, "AngularJS"
- Trotz der namentlichen Ähnlichkeit zu "AngularJS" sind beide nicht kompatibel und der Umstieg bedeutet mehr Aufwand und Probleme als man vermuten könnte
- Erscheinungsjahr 2016
- Aktuelle Version 11.2.4
- Ist Open-Source, wird aber von Google entwickelt und gewartet
- Der Code ist TypeScript, eine Erweiterung zu JavaScript. JavaScript-Programme sind gültige TypeScript-Programme, aber nicht umgekehrt

#### Vorteile

- Weit verbreitet und erprobt
- Viele erfahrene Entwickler
- Gut dokumentiert
- Stabil und gut gewartet
- "Schweizer Taschenmesser", bietet viele Funktionen, für die die Konkurrenz sich weiterer Frameworks bedient

#### Nachteile

- Hohe Lernkurve
- Großer monolithischer Block, der nicht sehr gut auf die eigenen Bedürfnisse reduziert werden

kann

- Dadurch auch etwas längere Ladezeiten
- TypeScript wird benötigt, nur JavaScript-Kenntnisse sind im Normalfall nicht ausreichend
- Hat ein etwas angestaubtes Image

### React

#### Eigenschaften

- Erscheinungsjahr 2013
- Aktuelle Version 17.0.1
- Ist Open-Source, wird aber von Facebook entwickelt und gewartet
- JavaScript-Code mit sogenannten JSX (Javascript XML)-Tags als Erweiterung

#### Vorteile

- Weit verbreitet und erprobt
- Gut dokumentiert
- Stabil und gut gewartet
- Moderate Lernkurve
- Prinzipiell Entwicklung auch mit "plain" JavaScript möglich, JSX ist optional
- Bietet mit dem Programm npx eine einfache Möglichkeit, über die Kommandozeile bereits funktionierende Web Apps zu konstruieren, die dann angepasst und erweitert werden können
- Das Rendering der Seiten ist extrem schnell, da eine Technologie namens Virtual-DOM benutzt wird, die aufwendige Operationen auf dem echten DOM (der Struktur) der HTML-Seite vermeidet

#### Nachteile

- Viele Funktionen müssen über Plugins erst hinzu konfiguriert werden
- Die große Auswahl von Third-Party-Plugins kann verwirrend sein und man kann sich



in den vielen Möglichkeiten, ein- und dieselbe Aufgabe zu lösen, verlieren

## Vue.js

### Eigenschaften

- Erscheinungsjahr 2014
- Aktuelle Version 3.0.7
- Ist Open-Source, es steht kein großes Unternehmen dahinter, sondern wird hauptsächlich durch die Community entwickelt und gewartet
- Bietet über sogenannte Direktiven die Möglichkeit, HTML-Code und JavaScript-Code verschmelzen zu lassen

### Vorteile

- Gut dokumentiert
- Flache Lernkurve
- Prinzipiell Entwicklung auch mit "plain" JavaScript möglich
- Direktiven erlauben es, sehr viel Funktionalität schon im HTML zu deklarieren, damit können auch Laien mit HTML-Kenntnisse eventuell Anpassungen vornehmen beziehungsweise den Code besser verstehen und nachvollziehen
- Das Rendering der Seiten ist extrem schnell, auch Vue.js nutzt ein Virtual DOM
- Sehr leichtgewichtiges Framework, dass sich sehr gut an die Bedürfnisse anpassen lässt und nur benötigte Funktionalität lädt
- Hat ein sehr modernes und fortschrittliches Image
- Ist laut Github das unter den Entwicklern beliebteste (nicht meist verwendete) Framework

### Nachteile

- Weniger Support als die beiden Alternativen
- Viele Funktionen müssen über Plugins erst hinzu konfiguriert werden.
- Weniger erfahrene Entwickler am Markt
- Weniger etabliert

## Entscheidungshilfe

Ich habe mir eine Tabelle überlegt, die Ihnen vielleicht bei der Entscheidung zwischen den Frameworks hilft. Überlegen Sie sich, wie wichtig die einzelnen Eigenschaften für Sie sind und bestimmen Sie einen Faktor dafür (0 - unwichtig bis 3 - sehr wichtig) und multiplizieren Sie die Werte in der entsprechenden Zeile damit. Dann addieren Sie die Werte für jedes Framework und das mit den meisten Punkten ist ein guter Kandidat als Basistechnologie für ihre Progressive Web App. ■

Eigenschaften	Angular	React	Vue.js
Leichtgewichtig, nur benötigte Funktionalität wird geladen	2	6	10
Umfassend, auch ohne Plugins wird ein hoher Funktionsumfang erlangt	10	3	2
Erprobt und angesehen, die Technologieentscheidung kann nur schwer angegriffen werden	10	10	5
Fortschrittlich und modern	5	7	10
Einfach zu lernen	3	6	10
Gut dokumentiert	10	10	8
Eine große Firma steht dahinter und sichert Weiterentwicklung und Bugfixes	10	10	2
Guter Support	10	9	5
Es gibt Entwickler am Markt, die erfahren in der Technologie sind	10	5	2
Entwickler wollen gern mit diesem Framework arbeiten	3	6	10
Geschwindigkeit des Startens der App	3	6	10
Geschwindigkeit während des Benutzens	8	10	9
Das Framework ist weit verbreitet	9	10	7
Die Userbase wächst schnell	7	2	10



**Markus Schwabeneder** ist Senior Consultant.

Er begann seine Karriere in der Softwarebranche als Consultant für komplexe mathematische Themen und Optimierungsaufgaben. Seitdem ist er in den Bereichen Softwareentwicklung, Anforderungsanalyse, Testing und Softwarearchitektur tätig. Seine Fähigkeiten setzt er in unterschiedlichen Branchen ein.

Besonders begeistert er sich für das Erarbeiten von komplexen Anforderungen und spezifischen Vorgehensweisen.

## ScriptRunner for Jira

von Stefan Ladstätter

In den letzten Jahren hat sich Jira als Projektmanagement-Tool für Software-Projekte durchgesetzt. In diesem Artikel möchte ich Ihnen ScriptRunner, das Schweizer Messer unter den Jira-Addons, vorstellen.

### Für wen ist ScriptRunner interessant?

Jira-Admins, die Jira um eigene Funktionalität erweitern wollen und nicht für Funktionalität, die von einer geringen Anzahl an Usern verwendet wird, ein eigenes Jira-Addon kaufen wollen.

### Jira ist überall

Einer der Hauptgründe für die Beliebtheit von Jira ist sicherlich die Erweiterbarkeit. Für jeden erdenklichen Einsatzzweck gibt es ein Plugin. Ob Testmanagement mit Xray, Multi-Projektmanagement samt Team-Kapazitätsplanung mit BigPicture oder dem Erstellen kompletter BI-Analysen und Reports mit EazyBI ist alles vertreten.

Aber auch für kleine Erleichterungen des Lebens gibt es Addons, z.B. für die Validierung mancher Workflow-Übergänge. Solche Verbesserungen machen sich rasch bezahlt, wenn eine große Anzahl an Benutzern davon profitiert. Wenn jedoch ein Feature nur von sehr wenigen Usern genutzt wird, fragt es sich, ob sich das Investment in mehrere kleinere Addons lohnt. Hier kommt ScriptRunner zum Zug.

### Das Schweizer Messer unter den Jira-Addons

ScriptRunner erlaubt es, den Funktionsumfang von Jira durch vorgefertigte und eigene Skripte zu erweitern und zu automatisieren. Die Liste der Möglichkeiten ist schier endlos:

Hinzufügen von Buttons, dynamische Formulare mit Default-Werten und Abhängigkeiten, Felder mit errechneten Werten sowie Workflow-Erweiterungen wie Validierungen, Bedingungen und automatisierten Aktionen nach einem Statusübergang, wie zum Beispiel Email-Benachrichtigungen, Hinzufügen von Sub-Tickets, uvm. Darüber hinaus kann man mit ScriptRunner REST-Endpoints hinzufügen und damit eine Schnittstelle zu externen Applikationen schaffen. Darüber hinaus auch umfangreiche JQL-Funktionen, die das Finden und Filtern von Jira-Tickets erheblich erleichtern.

### Custom Script Beispiel

Die mitgelieferte Script-Bibliothek kann um eigene Skripte erweitert werden, die in Groovy programmiert werden. Dies war zuletzt auch in meiner Praxis in einem Kundenprojekt nötig. Die Anforderung: Jedes Monat soll ein Report erzeugt werden, mit den Jira-Tickets aus einem bestimmten Projekt und ihrem aktuellen Status, sowie den Personen, die das je-

weilige Ticket bearbeiten. Der Report soll den gegenwärtigen Zustand zu Dokumentationszwecken archivieren. Dafür lohnte es nicht, ein eigenes Report/Export-Addon zu kaufen. Da Jira keine Standardfunktion für das Erzeugen und Abspeichern von Dateien mitbringt, haben wir uns für einen kleinen Trick entschieden: Statt einer Datei stellen wir eine Email-Nachricht mit Attachment zusammen, die wir anschließend versenden. Das Attachment hat das Format einer CSV-Datei. Der Empfänger des Emails kann das Attachment öffnen und lokal speichern.

### Hier die Bestandteile des Scripts:

1. Import von Jira-Scriptkomponenten (für Email, ScriptRunner, Suche, etc.)
2. Für jedes der betroffenen Jira-Projekte wird eine JQL-Query erzeugt, anhand der festgelegten Suchparameter
3. Die Liste der gefundenen Issues wird Eintrag für Eintrag ausgelesen, die gewünschten Felder (teilweise Custom-Fields) in eine



Titel: „Sommer-Landschaft“, Hans Steininger, Technik: Buntstift-Zeichnung

4. Liste gespeichert
5. Jede dieser Listen wird mit join() und dem Semikolon als Trennzeichen zu einer Textzeile umgewandelt
6. Hierbei wird jeder Feldeintrag via RegEx umgewandelt, damit Zeilenumbrüche, Anführungszeichen, etc. erhalten bleiben und die CSV-Datei korrekt in Excel importiert werden kann
7. Der resultierende String bekommt noch eine Titelzeile mit den Spaltennamen und wird dann an ein Multipart-Email angehängt
8. Am Ende wird das Email versendet

#### Scheduling

Am Ende muss das Skript noch über ein weiteres praktisches Feature von ScriptRunner in Betrieb gesetzt werden, über einen sogenannten Scheduled Job. Solche Skripte laufen zu einem bestimmten Zeitpunkt oder nach einem bestimmten Intervall ab, um automatisierte Aktionen durchzuführen. ■

#### Interesse?

Sie interessieren sich für maßgeschneiderte Zusatzfunktionen für Jira? Wir beraten Sie gerne.

Besuchen Sie gerne unsere Website [www.SEQIS.com](http://www.SEQIS.com) und kontaktieren Sie uns.



**SEQIS**  
**razzfazz.io**  
*boost results*

Abbildung: SEQIS GmbH



Interessante Videos aus den Bereichen IT Analyse, Development, Softwaretest und Projektmanagement finden Sie auf unserem Videoblog.

[www.seqis.com/youtube](http://www.seqis.com/youtube)



**Stefan Ladstätter-Thaa** ist Consultant.

In seiner mehr als 20-jährigen Tätigkeit als Entwickler, Requirements Engineer, Usability-Spezialist und Projektmanager hat er Einblick in die unterschiedlichsten Aspekte der Umsetzung von IT-Projekten gewonnen und kann dabei auf einen tiefen Erfahrungsschatz im klassischen und agilen Umfeld vorweisen.

Er sieht erfolgreiches Projektmanagement als Dienst an allen Projektbeteiligten – Auftraggeber, Entwicklungsteam und Anwender.

## Warum Node.js und Full Stack JavaScript die etablierten Programmiersprachen im Enterprise Umfeld ablösen wird...

von Alexander Vukovic

**Node.js is everywhere! Node.js ist eine Laufzeitumgebung für Backend-Javascript Applikationen. Als open-source und cross-platform Applikation konzipiert, ist Node.js auf jeder gängigen Serverplattform verfügbar.**

Die Anfänge von Node.js liegen im Jahr 2009, als Ryan Dahl Node.js entwickelt und auf der JSConf 2009 vorgestellt hat.

Die Laufzeitumgebung kombiniert die stärkste verfügbare JavaScript Engine, nämlich Googles V8 zusammen mit einer Event Loop und einem Subsystem für asynchrone I/O-Vorgänge. Später kam auch ein Paketmanager für Node.js genannt npm dazu. Anfängliche Uneinigkeit in der Community führten 2014 zu einem Community-Fork von Node.js, dieser wurde aber später wieder mit dem Core Projekt vereint und heute wird Node.js von der OpenJS-Foundation definiert und weiterentwickelt.

Heute ist Node.js eine ernsthafte Konkurrenz für die etablierten Größen wie Java, .Net oder PHP.

Aber warum ist das so? Warum setzen wir bei razzfazz.io primär auf Node.js?

Die Antworten sind vielfältig, hier eine Auswahl:

**Node.js ist für die Entwicklung von I/O-lastigen Backends optimiert und erreicht erstaunliche Performance** Microservices sind das heute gängige Architekturpattern. Diese meist atomaren kleinen Einheiten bilden einen kleinen Teil einer größeren Applikation isoliert entwickelbar, deploybar und skalierbar ab. Dabei werden häufig nur Daten zwischen Microservices

vermittelt oder aus einer Datenbank gelesen oder in diese gespeichert.

Klassische Ansätze, wie sie Java oder .Net verfolgen basieren auf Multithreading. D.h. ein neuer Request an den Applikationsserver führt dazu, dass ein neuer Thread eröffnet wird, dieser liest dann Daten aus der Datenbank, indem er mehrere Queries gegen die Datenbank absetzt und auf deren Ergebnisse wartet. Aus den Ergebnissen wird möglicherweise eine Antwort aggregiert und als Antwort auf den ursprünglichen Request zurückgeschickt.

D.h. der Thread besteht für die Dauer bis der Request vollständig abgearbeitet wurde. Das könnte bei hoher Belastung dazu führen, dass zu viele Threads gleichzeitig offen sind und z. B. im Falle von Java die JVM an ihre Grenzen stößt.

PHP läuft beispielsweise in Worker Threads des Webserver, die auch wiederum limitiert sind und bei hoher Last geshared werden müssen.

Node.js löst dieses Problem vollkommen anders. Eine Node-Instanz besteht im ersten Schritt aus genau einem Thread, der die sogenannte Event Loop betreibt. Sobald ein Request eintrifft entsteht ein Event, das abgearbeitet wird. Die Abarbeitung erfolgt dann durch sogenanntes Non-Blocking IO, d.h. Node ist dazu in der Lage IO-Requests in parallelen Threads zu starten und auf deren asynchrones Ergebnis zu warten. Im Falle einer Datenbankoperation würde eine Query abgesetzt werden, die Event Loop wartet nicht auf das Ergebnis, sondern, sobald das Ergebnis vorliegt, wird ein neues Event in der Event Loop generiert und im Javascript Code auf dieses asynchro-

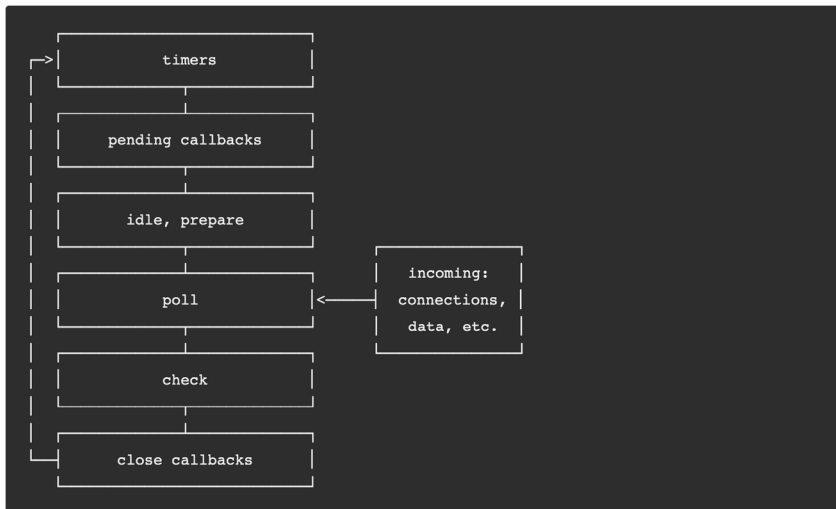
ne Event reagiert. Für hardwarenahes IO wie Netzwerk und Disk IO wird dieses in einen zusätzlichen Thread Pool ausgelagert.



Abbildung: SEQIS GmbH



The following diagram shows a simplified overview of the event loop's order of operations.



Each box will be referred to as a "phase" of the event loop.

Abbildung: <https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/>

Der große Vorteil dieses Ansatzes liegt darin, dass sehr IO-lastige Applikationen mit viel weniger Threads und mit viel effizienterer Ressourcennutzung auskommen. Mit Node.js sind deshalb viel performantere und skalierbare Applikationen möglich.

Der Nachteil von Node entsteht bei rechenintensiven Operationen, wie z. B. komplexen Kalkulationen. Diese blockieren die Eventloop und führen dadurch zu einer deutlichen Verlangsamung. Hierfür bietet Node zwar auch Workarounds mit echten Threads, aber man muss klar sagen, dass hierfür z. B. die Entwicklung einer C++-Extension zu Node zu einem performanteren Ergebnis führen.

Für I/O-lastige Aufgabenstellungen hingegen ist Node.js unschlagbar.

### Node.js ist durch die riesige Community effizient zu entwickeln

Nach anfänglichem Hickhack in der Node Community hat sich eine non-profit Organisation zur Weiterentwicklung von Node.js gebildet, die nunmehr seit vielen Jahren für eine stabile Weiterentwicklung sorgt. Das Zusammenspiel mit dem Node Package Manager Repository (NPM, [www.npmjs.com](http://www.npmjs.com)) hat dazu geführt,

dass eine riesige Entwicklungscommunity entstanden ist, die frei kommerziell verwendbare Frameworks und Lösungen auf Basis von Node bereitstellt.

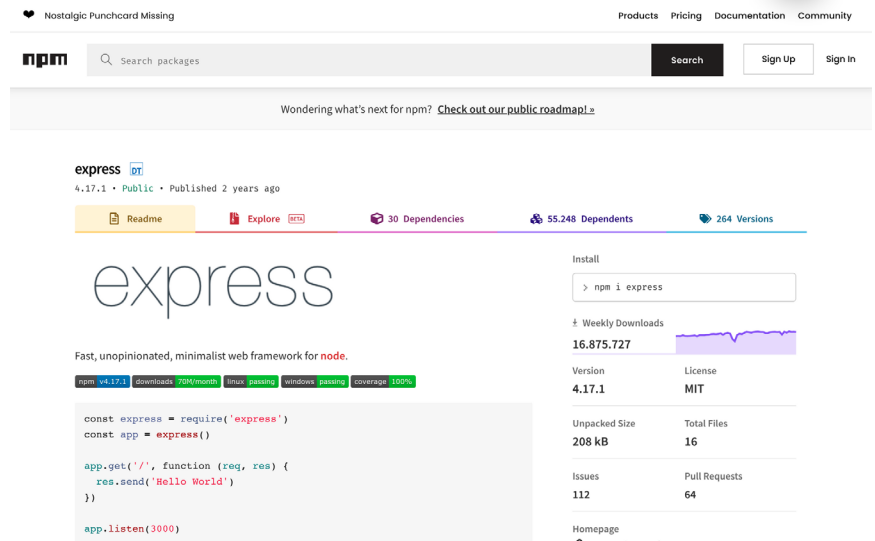


Abbildung: SEQIS GmbH

stellung, die sich bei der Entwicklung von Backends oder Microservices stellen.

### Die gleiche Programmiersprache im Frontend und im Backend

Im Bereich der Webclients im Browser hat sich Javascript seit vielen Jahren bereits durchgesetzt. Sobald eine Lösung einen Webclient mit etwas mehr Interaktivität benötigt, ist es notwendig, dass die Entwickler auch Javascript sprechen.

Deshalb ist es naheliegend, die gleiche Programmiersprache für das Backend zu verwenden. Daraus ergeben sich weitere Sharingmöglichkeiten und die Möglichkeit den Code aus dem Backend für das Frontend wiederzuverwenden.



### Optimierte Entwicklungsumgebungen ermöglichen erstaunliche Entwicklungsgeschwindigkeit

Ein gewichtiges Argument für Node.js ist die Performance. Dabei geht es nicht nur um die Ausführungsgeschwindigkeit (siehe Optimierung für I/O-lastige Aufgaben). Es geht auch um die Entwicklungsgeschwindigkeit. Das Node.js Ökosystem weist einen

sehr hohen Entwicklungsautomationsgrad auf, Scaffolding ermöglicht einen leichten Start für ein Projekt, in dem bereits alle Tools ebenfalls bereitstehen (z. B. Unit Test Framework, Oberflächenautomation, Buildautomation, etc.)

Die Buildschritte werden über ein ebenfalls Node.js basiertes Command Line Interface automatisiert, es ist dem Entwickler immer möglich, Veränderungen vorzunehmen und einzugreifen. Dabei hilft auch die extensive Unterstützung bei der Entwicklung durch das Integrated Development Environment (IDE). Dabei hat sich Visual Studio Code neben IntelliJ Webstorm als die beste Lösung herauskristallisiert.

Node.js ermöglicht es mit sehr wenig Zeilen Code sehr viel Funktionalität zu schaffen. Das in Kombination mit dem ganzen Toolstack ermöglicht erstaunliche Zeitgewinne bei der Entwicklung, die Node.js damit outstanding machen. Insbesondere zu den traditionellen Sprachen wie Java und .Net.

#### **Node.js ermöglicht entwicklungsnahe Qualität von Anfang an**

Insbesondere Unit Tests sind ein herausragendes Werkzeug, um Qualität in einer Lösung von Anfang an miteinzubauen. Als feinmaschiges Fallnetz ermöglichen die Unit Tests jede Veränderung direkt abzusichern und bei jedem Speichern mitzulaufen.

Node.js bietet hier viele Möglichkeiten für das Mocking und das schnelle Erstellen von Unit Tests.

Über NPM wird auch die Qualität der verwendeten Pakete sichergestellt und z. B. geprüft, ob es neue bekannte Security-Lücken in einem verwendeten Paket gibt und ein Upgrade empfohlen wird.

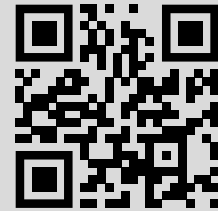
#### **Warum also Node.js statt ... ?**

Wir haben uns für Node.js entschieden, da Node.js die moderne Basis für

moderne Applikationen darstellt. Von Anfang an mit Qualität und Performance gedacht und höchst effizient umgesetzt sichert uns Node.js einen entscheidenden Wettbewerbsvorteil gegenüber anderen Lösungsanbietern.

Wir sind überzeugt, dass in den nächsten Jahren Node.js seinen aktuellen Erfolg im Enterprise Umfeld fortsetzen wird und die historisch etablierten Sprachen wie Java und .Net sukzessive verdrängen wird. ■

Wenn Sie Fragen zur Einführung von Node.js in Ihrem Entwicklungsteam haben, wenden Sie sich gerne an uns: Wir unterstützen Sie gerne und teilen unsere Erfahrungen mit Ihnen.



[www.razzfazz.io](http://www.razzfazz.io)

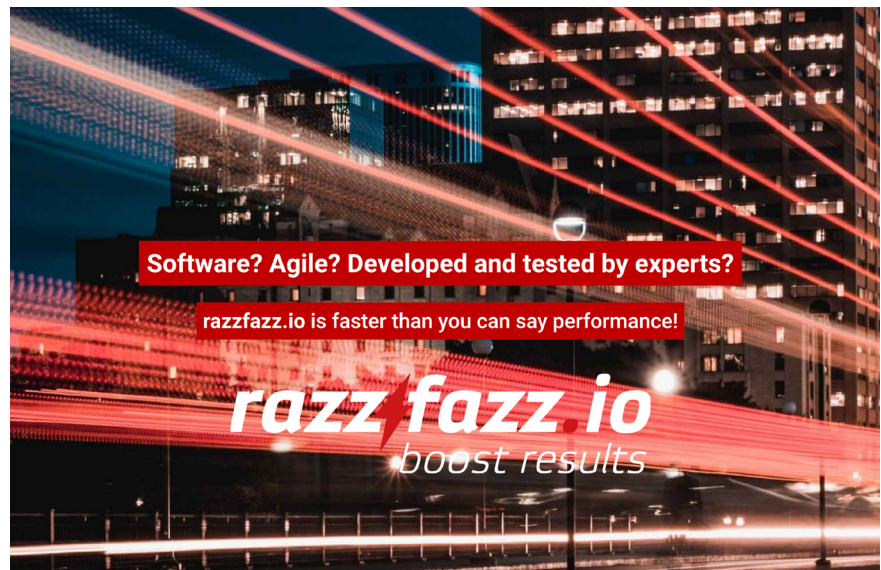


Abbildung: razzfazz.io Homepage



**Alexander Vukovic** ist SEQIS Gründer und Chief Evangelist.

Er ist erster Ansprechpartner für alle agilen, testmethodischen und testtechnischen Anfragen. In der Praxis arbeitet er als Agile Quality Coach, Berater, Interims-Testmanager, CI-Experte und Lasttester. Mehr als 25 Jahre Beratertätigkeit führten ihn während seiner zahlreichen Projekte in die unterschiedlichsten Branchen und Länder.

Sein persönliches Motto „Es gibt keine Probleme, sondern nur nicht gefunden Lösungen“ spiegelt sich in jedem Projekt wider.

## 5 Fragen an...

von Alexander Weichselberger

**Bezugnehmend auf die aktuelle QN mit dem Schwerpunkt Coding & Development fünf Fragen an: Alexander Weichselberger, Managing Partner SEQIS und verantwortlich für die SEQIS Expertinnen und Experten.**

### 1. *Muss man künftig codieren können?*

Wer programmieren kann, wird von Firmen umworben wie nie zuvor. Der Digitalisierungsschub seit dem Millenniumswechsel wurde nochmals verstärkt durch die letzten COVID-Monate. Dennoch muss nicht jeder coden können – aber jeder muss verstehen, wie die eigene Leistung mit der Wertschöpfung verknüpft ist.

Konkret: Starke Teams brauchen Ideen, Umsetzung, Qualitätssicherung, Support, Betrieb sowie ökonomische und ökologische Begleitung – nur dann entsteht Nachhaltiges. Welche Rollen zu einem passen oder wie man dabei mithilft – diese Entscheidung liegt bei einem selbst. Jedenfalls steht nichts für sich allein.

### 2. *Welche Programmiersprache sollte man lernen?*

Jährlich entstehen neuen Programmiersprachen – und nur manche sind gekommen um zu bleiben. Die Trends 2021 liegen lt. einer kurzen Internet-Recherche (Stack Overflow, statista, it-daily.net) bei Python, Objective-C, TypeScript und C/C++ – die Klassiker sind sicherlich Java, JavaScript, PHP und C#; letztere gerade auch wegen ihrer kontinuierlichen Weiterentwicklung.

Generell sollte man sich bei der Auswahl von Programmiersprachen fragen, was beruflich nachgefragt wird oder was man privat gerne machen möchte. Sicher ist: Wenn man die Sprache nicht anwenden kann, ist der

Wissenserhalt enorm schwierig und man fällt weit hinter den (Erfahrungs) Durchschnitt aller anderen.

### 3. *Wie kommt man zu Programmier-Wissen?*

... in meiner Praxis zeigt sich, dass das Studium bei Programmierern oft nur sekundär ist. Der tatsächliche Erfahrungsboost entsteht in der beruflichen Praxis und konkreten Umsetzung. Darüber hinaus treffen und tauschen sich Interessierte bei Hackerspaces oder Hackerthons aus. Natürlich werden zur Vernetzung soziale Netzwerke genutzt, eine Übersicht zu den aktuellen oder geplante Hackerspaces liefert zum Beispiel das Projekt [hackerspaces.org](https://hackerspaces.org), im Wikipedia findet man unter <https://de.wikipedia.org/wiki/Hackathon> eine Vielzahl von Angeboten.

### 4. *Reichen Programmiersprachen um erfolgreich zu sein?*

Hier ist die Antwort zumindest zweigeteilt: „Nur“ Programmiersprachen reichen nicht – im technischen Umfeld müssen Zusätze wie HTML, CSS, SQL bzw. Bash/Powershell auch beherrscht werden. Dies ist abhängig,

ob man mehr im Front- oder Backend tätig ist.

Darüber hinaus ist ein fundiertes Wissen von Build- und Deploymentprozessen bis hin zur Netzwerktechnik und Hardware obligat. Eins ist klar, niemand kann alles in der IT. Aber jedes Wissen ist hilfreich, und sich damit breitbandig auseinanderzusetzen, zu spielen und zu üben bringt den Erfolg.

### 5. *Muss man heute neue Algorithmen schreiben – oder reicht's zu wissen, was man zusammen"kopiert"?*

Mein Partner, Alexander Vukovic, hat das Thema in einem Video [www.youtube.com/watch?v=K7RBAaMh-4xE](https://www.youtube.com/watch?v=K7RBAaMh-4xE) beantwortet. Die Grundaussage: Heute gibt es bereits für alle bekannten Aufgabenstellungen Lösungen wie Sand am Meer. Zum Beispiel programmiert heute keiner mehr einen Algorithmus für Sortieraufgaben. Hier gilt es zu wissen, was man wie orchestrieren muss, damit es funktioniert.

Lediglich bei ganz Neuen ist wohl wieder ein Ausflug in die Grundlagen notwendig. ■



**Alexander Weichselberger** ist Managing Partner.

Er hat seine Einsatzschwerpunkte in den Bereichen Systemanalyse, Softwaretest, Koordination und Management von exponierten Großprojekten und kann auf jahrelange Erfahrung zurückblicken.

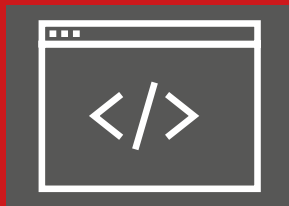
Dieses Wissen gibt er gerne in Form von Coachings, Methodentrainings und Fachvorträgen weiter.

SEQIS ist der führende österreichische Anbieter in den Spezialbereichen  
**IT Analyse, Development, Softwaretest und Projektmanagement.**  
Beratung, Verstärkung und Ausbildung:  
Ihr Partner für hochwertige IT-Qualitätssicherung.



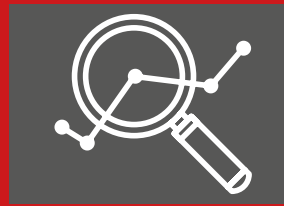
#### **IT ANALYSE**

Notwendige Änderungen analysieren und IT-gerecht aufbereiten



#### **DEVELOPMENT**

Guten Code schreiben und schlechten Code überarbeiten



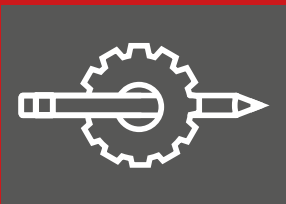
#### **TESTING**

Probleme durch methodischen Soll-Ist Vergleich erkennen



#### **RELEASE & OPERATE**

Reibungsloser Go Live und Betrieb der IT-Lösungen



#### **DEVOPS**

Neuerungen abgestimmt mit Entwicklung und Betrieb live setzen



#### **METHODOLOGY & TOOLS**

Vorgehensweisen optimieren und auf die richtigen Tools setzen



#### **TRAINING & WORKSHOPS**

Mitarbeiter Know-how stärken - standardisiert oder maßgeschneidert



#### **PROJEKT-MANAGEMENT**

Verantwortung übernehmen, zielorientiert und pragmatisch agieren