



Das akquinet Technology Radar 2013

Technologien, Methoden und Tools in der Softwareentwicklung

Nicole Charlier, Torsten Fink, Martin Grund, Philipp Kumar, Thomas Matzke

Zur Auswahl einer passenden technologischen Strategie gehört die kontinuierliche Bewertung aktueller Trends. Um Effizienz und Produktivität in den Projekten zu gewährleisten, ist der richtige Zeitpunkt zum Aufspringen auf neue Entwicklungen sorgfältig zu wählen. Unser „Technology Radar“ unterstützt bei diesen Entscheidungen, indem es über ein Umfrageverfahren aktuelle Technologien, Methoden und Werkzeuge hinsichtlich Reife und Verbreitungsgrad bewertet.

The Times They are A-Changin' ...

Das Entwicklungstempo in der IT-Welt hat sich in den letzten Jahren nicht verlangsamt. Stetig entstehen neue Trends, Technologien und Werkzeuge. Wer in diesem spannenden Umfeld bestehen möchte, muss sich ständig die aktuellen Entwicklungen anschauen und unter Einbezug der gemachten Erfahrungen gezielt entscheiden, wo und wann welche Kompetenzen aufgebaut werden und welche Felder doch besser zunächst zurück gestellt werden.

Unser Technology Radar soll uns bei dieser Entscheidung helfen, indem es einen Überblick über den Status quo von Technologien, Methoden und Tools in der Softwareentwicklung für 2013/2014 gibt. Das Radar ist aus den Einschätzungen von Experten aus den Bereichen Entwicklung, Projektmanagement, User Experience und Administration entstanden. Der Fokus des Radars liegt auf den Themen, die eine hohe Attraktivität für die genannten Bereiche aufweisen.

Fragen, die wir mit dem Technology Radar beantworten möchten, sind zum Beispiel „Welches Framework rückt im nächsten Jahr in den Fokus? Welche Methoden werden angewendet, um die Projekte noch effektiver und effizienter zu machen? Welche Tools sind erwachsen geworden und können produktiv eingesetzt werden?“ Alle zu bewertenden Technologien haben wir daher in Bezug auf ihre „Verwendungshäufigkeit“ und ihre „Reife“ betrachtet. Die Bewertungen kommen hauptsächlich von unseren Kollegen. Sie stellen damit ein Kondensat unserer Projekterfahrungen und unseres Kompetenzaufbaus in neuen Technologien dar, natürlich geprägt durch unsere Ausrichtung als Java/Java EE-Firma.

Wir haben unser Technology Radar in mehrere Kategorien unterteilt, da es dieses Jahr weitaus mehr Technologien in den Betrachtungsbereich geschafft haben. Folgende Kategorien haben wir identifiziert:

- ▼ Methoden
- ▼ Platforms & Middleware
- ▼ Datenbanken
- ▼ Frameworks & Bibliotheken
- ▼ Sprachen
- ▼ Werkzeuge



Im Folgenden werden wir zunächst unsere Vorgehensweise erläutern. Anschließend präsentieren wir die Ergebnisse für die einzelnen Kategorien.

Unsere Vorgehensweise

Die Erstellung des Technology Radars erfolgte in vier Schritten.

#1 Sammeln

Wir haben eine Vorbefragung mit LimeSurvey durchgeführt, die das Ziel hatte, unsere Liste des Technology Radars von 2011 zu aktualisieren. Hier haben wir unterschiedliche Fragen gestellt, um die Interviewten anzuregen, in unterschiedlichen Richtungen zu denken. Beispiele hierfür sind: Welche Technologien, Methoden und Tools waren 2013 wichtig? Ohne welche Tools könntest du gar nicht leben?

Die Kombination aus Vorstudie und der Liste von 2011 resultierte in ca. 270 Punkten. Jeder einzelne war interessant; in Summe waren es aber zu viele, um sie alle bewerten zu können.

#2 Auswählen

In einem Workshop haben wir aus den relevanten Technologien, Methoden und Tools eine Auswahl erstellt. Die zentrale Frage, die wir uns hierbei immer wieder gestellt haben, war: „Können wir mit dieser Technologie, Methode usw. webbasierte Geschäftsanwendungen entwickeln?“

Dabei haben wir allgemeine Paradigmen aus der Liste entfernt und sie durch spezifische Methoden und Technologien ersetzt. Zum Beispiel haben wir Responsive Design unter anderem durch die Frameworks Bootstrap und Foundation ersetzt. Jeder Punkt wurde dadurch spezifischer und bewertbarer im Hinblick auf unser Ziel: das Technology Radar.

#3 Bewerten

Unser Hauptziel war es, einen Überblick des aktuellen Marktes zu visualisieren. Hierfür haben wir die zwei Skalen „Reife“ und „Verwendungshäufigkeit“ gewählt.

- Die Skala der „Reife“ zeigt auf, als wie weit fortgeschritten die Technologieentwicklung wahrgenommen wird. Wird eine Technologie hier als „unreif“ bewertet, hat sie höchst wahrscheinlich noch Bugs, Architekturfehler, einen ungenügenden Funktionsumfang usw.
- Da die Reife nicht der einzige Grund für eine Technologie-Auswahl ist, haben wir auch die „Verwendungshäufigkeit in Projekten“ nachgefragt. Die „Verwendungshäufigkeit“ spiegelt hierbei die Verbreitung in den den Experten bekannten Projekten wider. Wenn eher unreife Technologien in Projekten genutzt werden, dann bieten sie wahrscheinlich Lösungen an, die innovativ sind und in den „alt-bekannt“ Technologien fehlen. Jede Skala hat hierbei vier Stufen, die in Tabelle 1 dargestellt werden.

#	Reife	Verwendungshäufigkeit
1	unreif	fast nie
2	eher unreif	eher selten
3	eher reif	eher häufig
4	reif	sehr häufig

Tabelle 1: Bewertungskriterien der Technologien

#4 Analysieren

Aus den eingesendeten Bewertungen wurden Technologien, die weniger als vier Antworten hatten, ausgeschlossen. Anschließend haben wir die Mittelwerte von Reife und Verwendungshäufigkeit berechnet, bei Nichtbetrachtung von Antworten, bei denen „keine Antwort“ gewählt worden ist.

Ergebnisse Technology Radar 2013

Im Folgenden erläutern wir die Ergebnisse des Technology Radars 2013. Die Umfrage wurde im Java-Bereich in Bezug auf die Entwicklung webbasierter Geschäftsanwendungen durchgeführt. In der grafischen Auswertung werden die Technologien in ihrer Reife als Blüte dargestellt. Je größer die Blüte, desto reifer ist die Technologie. Je länger der Ast, desto größer ihr

Verwendungshäufigkeit. Die Wurzelgrößen zeigen die Gesamtanzahl der darin befindlichen Technologien. Die genauen numerischen Ergebnisse mitsamt unserem Radar von 2011 finden sich auf der Website [TR].

Methoden

In der Anforderungsanalyse sind Methoden des Bereichs User Experience (z. B. User Context Analysis und UX Questionnaires) gut etabliert (s. Abb. 1). Szenario-basiertes Vorgehen mit einer vorgelagerten Analyse durch Beobachtungsmethoden (z. B. Field Observation) sind hierbei die Grundlage für den Entwicklungsstart einer benutzerfreundlichen Lösung.

Wireframing liegt bei der Gestaltung von Lösungen an der Spitze und unterstützt die Anforderungsanalyse und die Entwicklung ganzheitlich. Dicht dahinter folgen die Personas, die als Kommunikationsmittel zwischen allen Projektbeteiligten dienen. Für die Qualitätssicherung in Projekten sind Usability-Tests und Expertentests (Cognitive Walkthrough und Heuristische Evaluation) aktuell bewährte Mittel.

Eye Tracking liegt trotz seiner hohen Reife noch weit in der Benutzung zurück. Dies kann an erhöhten Kosten und den nur punktuellen Einsätzen liegen. Die Customer Journey Maps kommen frisch in den Methodenpool hinein und machen uns neugierig auf deren Entwicklung im nächsten Jahr.

Bei der agilen Entwicklung gehört Continuous Intergration (CI) zum Standardvorgehen. Das anspruchsvollere Continuous Delivery (CD) wird noch wenig benutzt, liegt aber beim Reifegrad nur knapp dahinter. Beide Methoden fördern die Kollaboration des gesamten Projektteams und das effiziente Ausliefern von der Idee bis zum Release.

Test-Driven Development (TDD) ist ausgereift und gut verbreitet. Die aktuellen Ansätze Behaviour Driven Development (BDD) und Acceptance Test-Driven Development (ATDD) liegen interessanterweise bezüglich Reifegrad und Verbreitung deutlich dahinter.

Plattformen & Middleware

Schauen wir auf die Plattformen (s. Abb. 2), dann sehen wir auf gleichwertiger Höhe Android und iOS. Android scheint spä-

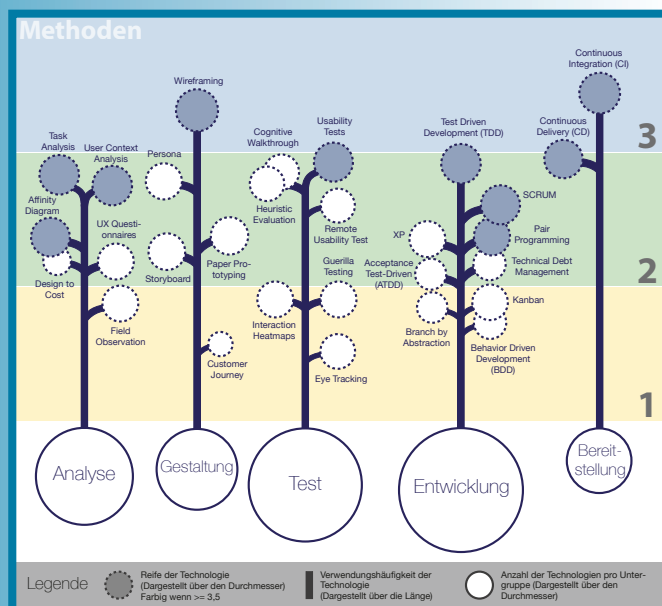


Abb. 1: Methoden

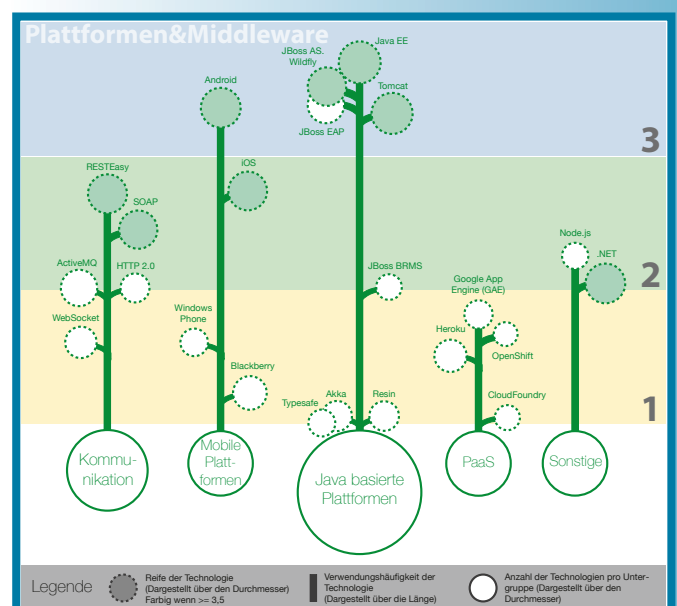


Abb. 2: Plattformen & Middleware



testens mit Einführung der Version 4.0 (Ice Cream Sandwich) zu iOS aufgeschlossen zu haben. Im Gegensatz dazu liegen BlackBerry und Windows Phone weit dahinter. Jedoch ist Windows Phone wahrscheinlich deutlich verbreiteter, wenn wir dies nicht im Java-Umfeld betrachten würden. BlackBerry profitiert noch von seiner Vergangenheit, wird aber im nächsten Jahr wohl nicht mehr hier auftauchen.

Die JBoss-Plattformen zeigen Dominanz in ihrer Reife und ihrer Verbreitungshäufigkeit. Interessanterweise wird Tomcat als ausgereifter angesehen als die Java EE-Plattformen, die im Allgemeinen selbst einen Tomcat beinhalten.

Als Vertreter des REST-Ansatzes wird RestEasy als ausgereifter und verbreiteter als SOAP betrachtet. Damit scheint die Bedeutung von SOAP kontinuierlich abzunehmen.

Datenbanken

Microsoft SQL ist in Microsoft-orientierten Umgebungen wahrscheinlich häufiger zu sehen. Im Java-Umfeld ist deren Verbreitungshäufigkeit geringer eingeschätzt. Relevant ist, dass PostgreSQL – mit einem leichten Vorsprung – und MySQL als genauso reif bewertet werden wie die kommerziellen Alternativen (s. Abb. 3). Interessant ist der große Abstand zum MySQL-Fork MariaDB.

MongoDB wird hier als reif angesehen, jedoch selten in Projekten eingesetzt. Damit scheint diese Datenbank eine Speziallösung zu bieten, während die anderen als universelle Datenbanken zum Einsatz kommen.

IndexedDB als neue klientenseitige Datenbank im HTML5-Umfeld spielt noch keine bedeutende Rolle, vermutlich auch deshalb, weil der Browsersupport vor allem auf Mobilgeräten noch stark zu wünschen übrig lässt. Apple scheint derzeit keine Pläne zu haben, IndexedDB auf iOS-Geräten zu unterstützen.

Frameworks & Bibliotheken

Als besonders reifer und weit verbreiteter Technologie-Stack schneidet Java EE (s. Abb. 4) sehr gut ab, insbesondere die Spezifikationen JPA und JSF sowie die JSF-Bibliotheken (RichFaces, PrimeFaces). Das Web-Framework Spring wird als ge-

nauso reif wie Java EE angesehen. AngularJS hat sich vergleichsweise schnell etabliert.

Der Einsatz von CSS-Präprozessoren wie LESS wird zum Standard. Dies ist nicht verwunderlich, da reines CSS schwer zu warten ist. CSS-Frameworks sind ausgereift und schon breit im Einsatz, wenn auch noch nicht selbstverständlich. Weit verbreitet ist hier die HTML-CSS-JavaScript-Komponentenbibliothek Bootstrap.

jQuery gehört weiterhin zu den beliebtesten und am stärksten verbreiteten JavaScript-Bibliotheken. Sie gilt als sehr ausgereift und bietet einfache APIs zur Lösung grundlegender, wiederkehrender Probleme. Zudem hängen andere Frameworks und Bibliotheken oft von jQuery ab, was die Verbreitung weiter fördert.

PhoneGap gilt mittlerweile als reif und auch zu einem gewissen Grad als verbreitet. Es könnte zusammen mit den Apps auf HTML5-Basis an Bedeutung gewinnen. Gleichzeitig gelten Sencha Touch und jQuery Mobile als On-Top-Lösungen zu PhoneGap nicht als ausgereift. Für mobile Geräte konnte sich noch kein JavaScript-Framework durchsetzen.

Feature Detection vor allem auf Mobilgeräten scheint immer wichtiger zu werden. Modernizr als zugehöriges Framework ist relativ verbreitet.

Sprachen

Java führt als meist verwendeter und reifster Kandidat den Bereich Sprachen (s. Abb. 5) an. Interessanterweise wird JavaScript als ebenso ausgereift angesehen. Der Verbreitungsgrad von JavaScript rührt sicherlich aus dem Alleinstellungsmerkmal als Browser Sprache. CoffeeScript als sauberer Aufsatz zu JavaScript wird kaum angetroffen und gilt als vergleichsweise unausgereift und damit risikoreich. Weitere Alternativen wie Dart gelten als unausgereift und werden kaum verwendet.

HTML5 setzt sich stark durch. Viele produktive Lösungen setzen mittlerweile sehr bewusst auf HTML5-Technologien. Es bietet von Haus aus einige Funktionalitäten, die früher mit JavaScript umgesetzt werden mussten. Außerdem bietet es multimodale Schnittstellen für mobile Geräte.

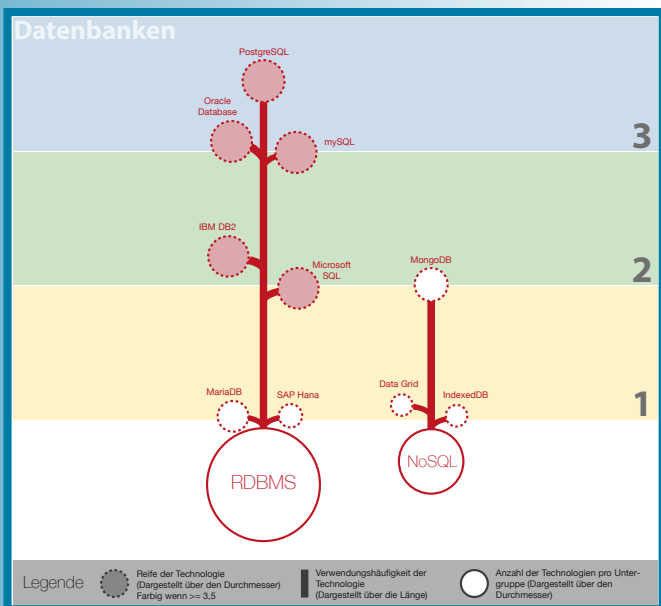


Abb. 3: Datenbanken

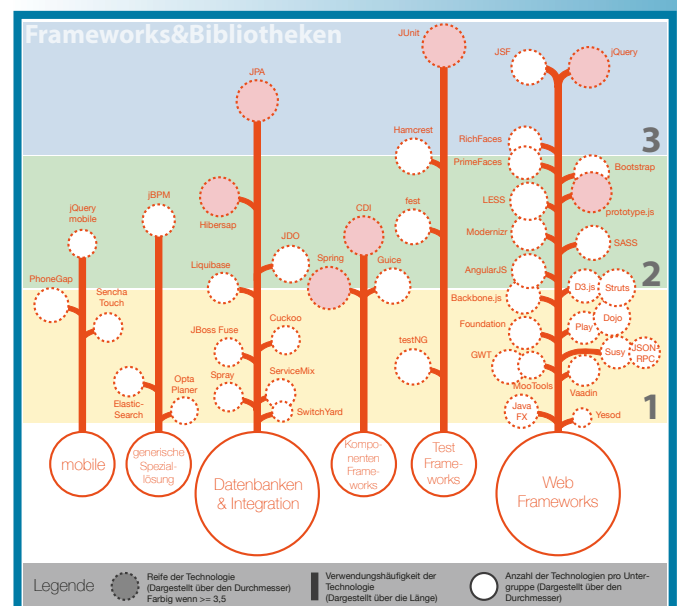


Abb. 4: Frameworks & Bibliotheken

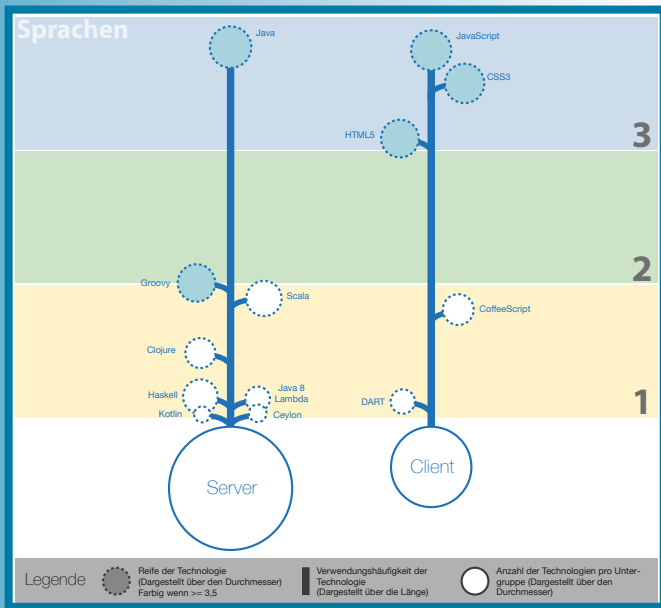


Abb. 5: Sprachen

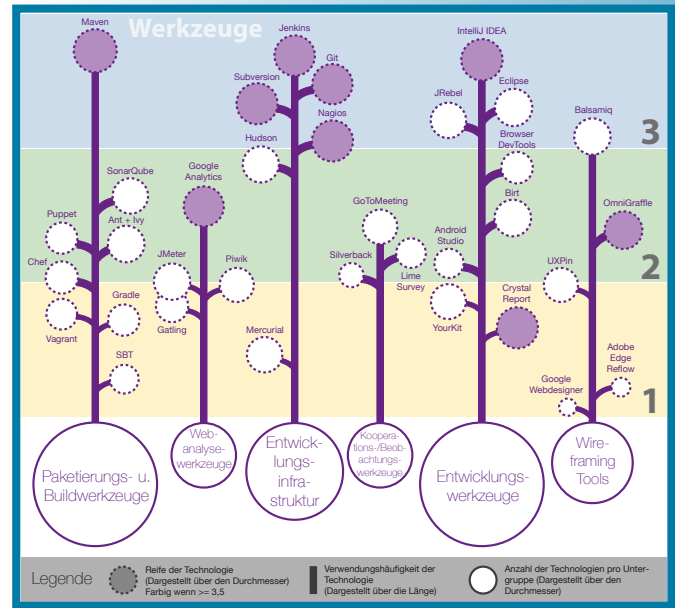


Abb. 6: Werkzeuge

Der Trend geht dazu, auch serverseitige Sprachen beim Client verfügbar zu machen, in dem man die Sprachen nach JavaScript übersetzt. Ceylon und Kotlin sind zwei Beispiele für Sprachen, die für die JVM und zu JavaScript kompiliert werden können. Zudem wird JavaScript mittlerweile auch auf dem Server ausgeführt, durch Node.js.

Haskell, dem akademischen Elfenbeinturm entstammend, gilt interessanterweise als schon sehr ausgereift. Dies gilt auch für Scala, aber es wird selten in Projekten eingesetzt.

Werkzeuge

Maven ist das dominierende Build-Tool (s. Abb. 6). Gradle sowie SBT nehmen stark an Reife und Verbreitung zu, liegen aber noch deutlich hinter Maven. Alle großen IDEs unterstützen Gradle mittlerweile von Haus aus. Im Android-Bereich setzt Google nun auf Gradle als Build-System für Android-Projekte, was die Verbreitung weiter fördert. Gradle entwickelt sich zu einem ernststen Herausforderer für Maven.

Idea liegt deutlich vor Eclipse. Beide IDEs gelten aber als ausgereift und verbreitet. Browsertools als Entwicklungswerkzeuge werden immer wichtiger und deuten auf einen zunehmenden Einsatz von Client-Technologien und -Logik, die sich im Browser evaluieren und manipulieren lassen. JRebel scheint ein Quasi-Standard zu werden, der auch ausgereift ist. Es deutet auf einen großen Bedarf nach solchen Lösungen hin, die Turnaround-Zeiten reduzieren und somit die Effizienz der Entwicklung begünstigen können. Android Studio, die Entwicklungsumgebung von Google, ist trotz seines Preview-Status schon sehr weit und wird weiter zulegen.

Git scheint Subversion als De-facto-Standard in Projekten abzulösen, denn die Tendenz von Subversion ist rückläufig. Der hohe Nutzungsgrad von Git hängt aber auch mit der enorm hohen Verbreitung im Open-Source-Umfeld zusammen. Mercurial wird insbesondere im Vergleich zu Git kaum angetroffen.

Crystal Reports gilt als etwas ausgereifter als BIRT, wird aber deutlich seltener angetroffen.

Bei den Tools im Bereich Wireframing/Prototyping hat sich Balsamiq immer noch konstant gehalten. Es werden jedoch aus

Sicht des Responsive Webdesigns bald neue Tools erwartet, die diesen Entwicklungsansatz bestmöglich unterstützen. Adobe Edge Reflow und Google Webdesigner zeigen bereits entsprechende Ansätze. Diese sind allerdings noch unausgereift und nicht produktiv einsetzbar.

Silverback als Vertreter der Usability-Test-Tools verliert etwas an Bedeutung, da Remote Tests immer wichtiger werden. Hier sind Online-Meeting-Tools wie GoToMeeting und Umfragetools wie LimeSurvey nach vorne gerückt.

Reife-Veränderung – 2011 vs. 2013

Wichtig neben einer Bestandsaufnahme ist natürlich auch die Veränderung zu den Vorjahren. Hierfür haben wir die Veränderung der Reife für alle Punkte berechnet, die auch schon bei der Erstellung des vorherigen Radars 2011 dabei waren. Es waren insgesamt 46 Technologien, Methoden und Werkzeuge. Von diesen hatten zehn eine Veränderung von über

Technologie	Reife Veränderung 2011 zu 2013	Reife 2013 (norm.)	Aktivität 2011
Node.js	+0,46	0,86	3,00
Guerilla Testing	+0,33	0,83	3,00
PhoneGap	+0,31	0,81	2,67
jQuery Mobile	+0,29	0,68	2,80
Play	+0,28	0,79	3,00
Scala	+0,28	0,82	2,73
Continuous Delivery (CD)	+0,26	0,88	2,87
Sencha Touch	+0,25	0,67	3,00
Groovy	+0,22	0,88	2,25
HTML5	+0,20	0,88	3,00

Tabelle 2: Reife und Aktivität



$\pm 0,20$ auf einer standardisierten Skala. Die Veränderung der Reife korrelierte insgesamt hoch mit der Aktivität in 2011 ($r = 0,65$). Die Skala für Aktivität spiegelte die Anziehung für Nutzer und Entwickler sowie die Geschwindigkeit der weiteren Entwicklung.

Tabelle 2 zeigt die Ergebnisse. Node.js etabliert sich immer mehr als Herausforderer für klassische Anwendungsserver. PhoneGap gilt mittlerweile als ausgereifte Lösung, um native Apps auf Basis von Webtechnologien umzusetzen. jQuery Mobile und Sencha Touch wurden stark weiterentwickelt, werden aber noch immer nicht als ausgereift angesehen. Scala hat weiterhin ein starkes Momentum und entwickelt sich zu einem Key-Player bei den JVM-Sprachen. Continuous Delivery ist mittlerweile eine ausgereifte und praktikable Alternative zum Big-Bang-Release. HTML5 ist jetzt schon der etablierte und breit unterstützte Standard für die Webentwicklung. Es ist kein ernsthafter Mitbewerber zu erkennen.

Zusammenfassung und Ausblick

Unser Technology Radar hilft uns bei Bestimmung und Validierung unserer technologischen Strategie. Es gibt wertvolle Hinweise, um erfolgreich im Spannungsfeld zwischen dem Hinterherlaufen jeden Hypes und dem Verpassen des technischen Anschlusses zu navigieren. Auch 2015 werden wir ein neues Technology Radar erstellen und sind schon jetzt gespannt auf die kommenden Entwicklungen.

Links

[TR] Technology Radar – akquinet,
<http://radar.akquinet.de/radar>



Nicole Charlier ist Leiterin des Competence Center für User Experience bei der akquinet. Sie ist Usability-Beraterin und leitet Projekte mit dem Schwerpunkt User Centered Design.

E-Mail: nicole.charlier@akquinet.de



Dr. Torsten Fink ist Geschäftsführer der Berliner Einheit der akquinet. Neben der Leitung von Java EE-Projekten führt er Architektur- und Technologieberatungen durch.

E-Mail: torsten.fink@akquinet.de



Martin Grund ist verantwortlich für User Experience bei der akquinet. Mit seinen Studienschwerpunkten in Cognitive Neuroscience und Human-Computer Interaction setzt er strategische Impulse, um User-centered Design mit unserem agilen Entwicklungsprozess zu kombinieren.

E-Mail: martin.grund@akquinet.de



Philipp Kumar ist Leiter des Competence Center für mobile Lösungen bei der akquinet. Seine Schwerpunkte liegen in der Anforderungsanalyse mobiler Systeme, auf Android-basierten Anwendungen und auf der Integration mobiler Lösungen in Unternehmensinfrastrukturen.

E-Mail: philipp.kumar@akquinet.de



Thomas Matzke mobiler Entwickler bei der akquinet. Er ist hauptsächlich in Projekten tätig, in denen Android, iOS oder mobile Web zum Einsatz kommt.

E-Mail: thomas.matzke@akquinet.de