

Fehler in Codeabschnitten

(Mobile Programmierung; Mit Tabris.js Apps für Android und iOS entwickeln; iX 8/2019, S. 69)

In die erste Spalte auf Seite 69 hat sich ein Fehler eingeschlichen. Am Ende des Satzes „Das eigentliche Framework einschließlich der CLI-Version installiert man per“ lautet der korrekte Befehl `npm install -g tabris-cli`.

J. EICKHOLD, VIA E-MAIL

Der Leser hat recht (d. Red.).

Kleine Lyrik zum Thema

(Programmierung; Low Code: Turbo für den Entwicklungsprozess; iX 8/2019, S. 76)

Ich bin schon lange im Geschäft und höre,
mit ständig wachsendem Vergnügen,
die Marketroids vom Himmel lügen.

Ich hab das jetzt, ich schwöre,
in jeder Firma, die ich kenne,
jedoch hier nicht beim Namen nenne,
von weisen Chefs gesagt bekommen.

Doch wurden die beim Wort genommen,
oh je, was war das für ein Graus,
sobald das Beispiel überschritten,
da mussten sie um Hilfe bitten,
sonst kam aus dem System nix raus.

WOLFRAM JAHN, AUS DEM iX-FORUM

Framework-Problem mal zehn

(Programmierung; Low Code: Turbo für den Entwicklungsprozess; iX 8/2019, S. 76)

Bei Frameworks hat man oft das Problem, dass das Arbeiten damit extrem einfach ist, solange man genau das tut, was sich der Framework-Entwickler gedacht hat. Wenn man aber nur einen Zentimeter davon abweichen will, muss man ewige Workarounds machen und es ist ein irr-sinniger Aufwand, auch nur kleine Dinge zu implementieren.

Es gibt schon Low-Code-artige Umgebungen, wo es sinnvoll ist. Zum Beispiel, um mit CMS-Systemen eine normale Website ohne besondere Features zu erstellen. WordPress nimmt jemandem, der nur Content produzieren will, das Programmieren ab. Aber wehe, er will mal ein Custom-Element einfügen.

Das große Problem von Low-Code-Umgebungen ist, dass ein Großteil der Programme nicht einfach auf Schema F passen. Webseiten sind einfach, die haben

meist die immer gleichen Elemente. Apps sind schwieriger und echte Applications bestehen fast nur aus Custom-Lösungen.

Und die einfachen Sachen, die in einer Low-Code-Umgebung auch wirklich einfach sind, sind dank Bibliotheken auch in echten Programmiersprachen einfach.

DAVID, AUS DEM iX-FORUM

Nicht erwartet

(Codescanner; Ein kritischer Blick auf statische Codeanalyse-Verfahren; iX 8/2019, S. 112)

SCA ist ein weiteres Werkzeug, um nicht böswillige Schlamperei einfacherer Sorte mit wenig Aufwand in der Breite anzugehen. Ein gutes Werkzeug liefert dabei möglichst wenig False Positives, um den Aufwand/Nutzen klein zu halten.

Aber es kann doch niemand ernsthaft erwartet haben, damit eine Code Review zu ersetzen? Oder dass mutwillig designte Lücken damit immer gefunden werden? Von daher kann doch von einem „kritischen Blick“ gar keine Rede sein.

KAI SCHWEBKE, AUS DEM iX-FORUM

Zu viele False Positives

(Codescanner; Ein kritischer Blick auf statische Codeanalyse-Verfahren; iX 8/2019, S. 112)

SCA findet nur haarsträubend verzapften Unsinn. 99 % der möglichen Sicherheitslücken kann ein SCA nicht finden.

Aber gerade in großen Firmen wird es eingesetzt, damit das Management sagen kann, sie haben alles gemacht, damit die Anwendung sicher ist. Es geht um den Haken im Excel-Sheet und letztlich „cover-my-ass“. SCA ist da nicht die einzige unsinnige Maßnahme.

JÖRG, AUS DEM iX-FORUM

Der direkte Draht zu



Direktwahl zur Redaktion: 0511 5352-387

Redaktion iX | Postfach 61 04 07
30604 Hannover | Fax: 0511 5352-361
E-Mail: post@ix.de | Web: www.ix.de

www.facebook.com/ix.magazin
twitter.com/ixmagazin (News)
twitter.com/ix (Sonstiges)

Für E-Mail-Anfragen zu Artikeln, technischen Problemen, Produkten et cetera steht die Redaktion gern zur Verfügung.

Listing-Service:
Sämtliche in iX seit 1990 veröffentlichten Listings sind über den iX-FTP-Server erhältlich: [ftp.heise.de/pub/ix/](ftp://ftp.heise.de/pub/ix/)

Auflösung des Bilderrätsels im iX-Special „IT heute“

Im iX-Special „IT heute“ fragten wir im Bilderrätsel nach zehn Ereignissen der letzten 30 Jahre IT-Geschichte und nach ihren Jahreszahlen. Hier die Auflösung:

- Q) Die drei plattformunabhängigen respektive plattformfreien Programmiersprachen heißen **Java**, **JavaScript** und **PHP**. Alle drei entstanden im Jahr **1995**.
- I) Im Jahr **2001** veröffentlichte das NIST den Datenverschlüsselungsstandard **AES**. Ihm zugrunde liegt der belgische Algorithmus Rijndael, der die Ausschreibung des NIST gewann.
- U) Im April **2011** prägte die Hannover Messe den Begriff **Industrie 4.0**. Ein paar Monate später, im September, veröffentlichte das NIST seine noch heute gültige Cloud-Definition „SP 800-145 The NIST Definition of **Cloud Computing**“.
- T) **PCI Express** oder kurz PCIe heißt der serielle Systembus, den die PCI-SIG im Jahr **2003** einführt.
- U) **1998** veröffentlichte das W3C die Metasprache **XML 1.0** als Recommendation. Im selben Jahr erschien auch die einfache Open-Source-Datenbank **NoSQL**, die kein SQL verstand und den

Namen gab für einen neuen Trend der Datenorganisation.

- B) **Linux** erblickte **1991** das Licht der Welt.
- O) Seit Apple im Jahr **2007** das **iPhone** einführt, sind Smartphones nicht mehr wegzudenken.
- S) Mit **HTML 5** schuf das W3C im Jahr **2014** eine neue Version, die sich nicht nur für die Web-, sondern auch die App-Entwicklung eignet.
- U) 30-jähriges Jubiläum feierte nicht nur iX, sondern auch das WWW. Dessen Vater **Tim Berners-Lee** schlug die Queen **2004** zum Ritter.
- I) **1992** stellte DEC den **64-Bit-Prozessor Alpha** vor. 1998 präsentierte AMD die K7-Architektur, Grundlage von Athlon und Opteron, für deren Entwicklung AMD eine Lizenz der Alpha-Technik erworben hatte. Zu späten Ehren kam der Alpha nochmals 2011, als AMD die Clustered-Integer-Core-Architektur des Alpha 21264 oder EV6 in die eigene Bulldozer-Architektur übernahm.

Das Lösungswort lautet **UBIQUITOUS**. Die 30 Gewinner haben wir bereits benachrichtigt. Wir bedanken uns bei allen, die teilgenommen haben.
(sun@ix.de)

Alte Werkzeuge

(Programmierung; Low Code: Turbo für den Entwicklungsprozess; iX 8/2019, S. 76)

Ich hatte das zweifelhafte Vergnügen, zwei Jahre in einem Salesforce-Implementierungsprojekt gelandet zu sein. In dieser Zeit habe ich mich bis zum Application Architect durchzertifiziert, in der Hoffnung, dieses Projekt technisch irgendwie unter Kontrolle zu bekommen.

Kurz gesagt: Der Low-Code-Ansatz lässt sich gut verkaufen, praktisch ist er schon bei trivialsten Ansätzen wertlos. Spätestens wenn in einer Installation mehrere abgeschottete Datentöpfe gebildet werden sollen (zum Beispiel weil eine Holding globalen Zugriff braucht), hat sich das Entwickeln durch Businessanwender schon wieder erledigt; schlecht designte Berechtigungsstrukturen geben ihr Übriges dazu.

Von nicht funktionierenden Deployment-Mechanismen über Trigger in zufälliger Reihenfolge, eine hoffnungslos veraltete Programmiersprache bis hin zur faktischen Unmöglichkeit partieller Restores reichen die Probleme, die zu unglaublich langatmiger Entwicklung führen. Die Schnelligkeit am Anfang dreht sich dann ganz schnell zu einer unglaublichen Zähigkeit. Arbeit, für die man übrigens unmöglich „normale“ Entwickler begeistern kann, die es wegen der Komplexität jedoch braucht. Es fühlt sich kon-

sequent so an, als wolle man Probleme von heute mit Mitteln von 2004 lösen.

Dies lässt sich aus Unternehmenssicht ja vielleicht noch ignorieren, aber wer sich freiwillig mit seinen Kernprozessen solch einer Plattform ausliefert, kauft sich gegebenenfalls auf Jahrzehnte ein Locked-in-Syndrom ein und wird in Zukunft hemmungslos gemolken.

MARCEL NORMANN, AUS DEM iX-FORUM

Ergänzungen und Berichtigungen

Programmierung; Die großen Fünf;
Die Neuerungen in C++20; iX 8/2019, S. 46

Überraschend wurden Contracts beim ISO C++ Meeting in Köln (15. bis 20. Juli) aus dem kommenden C++20-Standard entfernt, da es zu dem Zeitpunkt zu viele unterschiedliche Ansichten hinsichtlich des exakten Designs gab. Zum Belegen der Meinungsverschiedenheiten bildete man die neue Study Group für Contracts. Damit dürfte das Konzept wohl eines der großen Features des C++23-Standards werden.

Die iX-Redaktion behält sich Kürzungen und auszugsweise Wiedergabe der Leserbriefe vor. Die abgedruckten Zuschriften geben ausschließlich die Meinung des Einsenders wieder, nicht die der Redaktion.