

„DevOps ist die nächste Stufe der Agilität!“

T. SCHISLER

U. BAUMANN



Thomas Schissler

ist Bereichsleiter der Softwareentwicklung bei der artiso solutions GmbH und Microsoft MVP für Visual Studio Application Lifecycle Management (ALM). Er zählt deutschlandweit zu den Top-Experten für .NET-Technologien, Team Foundation Server, moderne Architekturkonzepte und Prozessdesign. Als zertifizierter Professional Scrum Trainer und Professional Scrum-Master nutzt er dabei seine umfassende Erfahrung aus der firmeninternen Scrum-Praxis und aus zahlreichen Trainings und Coachings bei Kunden.

Uwe Baumann

ist ausgewiesener Experte für Team Foundation Server (TFS), Application Lifecycle Management (ALM) und .NET. Er war unter anderem als Technologieberater und Produktmanager für ALM-Tools bei der Microsoft Deutschland GmbH tätig und ist Fachautor und Sprecher auf Konferenzen. Als ALM Consultant, Scrum-Trainer und strategischer Technologieberater bei der artiso solutions GmbH unterstützt er mit seiner umfassenden Erfahrung Kunden dabei, die Herausforderungen der modernen Softwareentwicklung zu meistern.

10 Releases am Tag: Eine Geschichte über DevOps

Freitag, früher Nachmittag. Der Rollout steht an. Die Anspannung ist beim ganzen Team greifbar. Alle sind da, Manager und Entwickler. Ich nippe an meinem Energy-Drink. Jeder wartet darauf, dass der letzte Build durchläuft. Die Admins bereiten sich darauf vor, die finale Version in die Produktionsumgebung einzuspielen. Denn am Montag muss alles funktionieren. Dann ist der große Tag ... unser Release geht live.

Der Vertrieb hat unsere Kunden informiert, am Montag ist die große Launch-Veranstaltung. Im Anschluss wird die neue Produktversion live geschaltet. Deren Erfolg oder Misserfolg wird über die Zukunft unserer Firma entscheiden. Es muss einfach klappen.

Aber wir sind zuversichtlich. Die Arbeit von fast zwei Jahren wird sich auszahlen. Damals wurde uns klar: Wir sind mit unserem Produkt weit hinter unseren Mitbewerbern. Der Vorstand war alarmiert, Gelder wurden genehmigt, das Projekt „Aufholjagd“ gestartet. Für mich bedeutete dies zahllose Überstunden und in den letzten Wochen hat unser Team an den Wochenenden praktisch durchgearbeitet. Die harte Arbeit hat sich jedoch gelohnt: Wir haben die benötigten Funktionen implementiert.

Damit stehen wir sogar besser da als unsere Mitbewerber.

Risiko Big Bang Release

Noch ist es jedoch nicht ganz so weit. Wir sind nervös. Alle sagen sich: Es wird schon klappen. Wir haben ausgiebig getestet, und außer ein paar kleineren unkritischen Bugs, die wir schon kennen, sollte eigentlich alles rund laufen. Jetzt können wir also nur noch warten – auf Montag, den großen „Big Bang“ Release-Tag.

Wahrscheinlich wird alles gut gehen. Aber vielleicht auch nicht? Schwer zu sagen, wenn man nur alle eineinhalb Jahre ein neues Release präsentieren kann. Wir haben damit einfach keine Erfahrung!

Und es könnte so viel schief gehen. Immer wieder gab es im Team Diskussionen: Wie werden die Kunden auf unsere neuen Features reagieren? Wird der Server auch unter Last gut funktionieren? Einige der neuen Funktionen belasten die Datenbank doch mehr als bisher. Und es ist nicht zu leugnen, dass einige der Features auch heute schon wieder ein wenig angestaubt sind – der Markt hat sich in der langen Zeit, seit die Anforderungen geschrieben wurden, doch ziemlich verändert...

Klar, unsere Firma entwickelt Software noch nach dem klassischen Wasserfall-Modell. Anforderungen möglichst komplett erfassen, dann eine lange Implementierungsphase, die Tests und nötige Bugfixes, dann das Release. So war das schon immer bei uns.

„DevOps ist die nächste Stufe der Agilität!“

T. SCHISLER

U. BAUMANN

Agilität mit Hindernissen

Meine Gedanken schweifen ab, und ich denke an das Meeting unserer .NET Usergroup am letzten Mittwoch.

Viele Kollegen, die ich dort treffe, arbeiten in ihren Projekten nach agilen Methoden. Startup-Style. Kurze Zyklen von zwei Wochen. Das gibt ihnen die Möglichkeit, die Anforderungen ständig neu zu priorisieren und schnell auf Veränderungen im Markt zu reagieren.

Am Ende jedes Zwei-Wochen-Sprints liefern die agilen Teams funktionierende, getestete Software aus. So können sie den Nutzern die neuen Features früher in die Hände geben, erhalten früher echtes Feedback und können darauf

Big Bang Releases

Das Team arbeitet monate- oder jahrelang an einem Projekt - mit der Intention, das entstandene Produkt mit einem „Großen Knall“ als Ganzes live zu stellen. Big Bang Releases werden oft mit klassischer Wasserfall-Entwicklung in Verbindung gebracht, aber auch Scrum Teams können sich dazu gezwungen sehen.

auch schnell reagieren. Und sie sehen auch gleich, ob die Software im realen Betrieb stabil und performant läuft. Eigentlich schon eine prima Sache.

In der Theorie.

Die Realität sieht aber oft anders aus. Beim Bier nach dem Usergroup-Vortrag erzählte mein Freund Peter - selbst Mitglied eines agilen Teams - von seinem momentanen Hauptproblem: „Es macht schon Spaß, in Sprints zu arbeiten und selbstbestimmt im Team die Arbeit zu

organisieren. Cool ist auch, dass man regelmäßig Resultate in Form von fertiger Software sieht - man weiß, was man geleistet hat. Aber dann der Frust mit den Jungs aus der IT...“

Peter seufzte.

Schwachstelle agile Release-Pipeline

„Wir liefern alle zwei Wochen Software an unseren Product Owner“ fuhr er fort. „Der ist auch meistens glücklich. Aber dann geht erstmal wenig weiter. Denn unsere Admins müssen ja nun die Software auch auf die Server ausrollen - und das kann dauern. Schließlich haben wir immer noch den guten alten Stufenprozess für das Deployment.

Du kennst das: Die Bits müssen erst von unserem Development-Server auf den Staging-Server kopiert werden, dabei ist auch noch jede Menge Konfigurationsarbeit nötig. Wenn im Staging alles funktioniert, kann die Software auf dem Produktionsserver installiert werden. Und jedes dieser Deployments muss zwischendrin von verschiedenen Verantwortlichen getestet und freigegeben werden. Die Tests sind zeitaufwändig. Und wehe, da ist mal jemand im Urlaub - dann kommt der ganze Prozess einfach zum Stillstand.

Zudem machen die Admins vieles beim Deployment noch manuell. Sie verlangen von uns genaue Change Logs und detaillierte Deployment-Anleitungen, damit sie wissen, welche Files sie wie einspielen müssen und was konfiguriert werden muss. Aber oft passieren trotzdem noch Fehler, dann rufen Sie bei uns an und wir müssen helfen.

Peter nippte nachdenklich an seinem Bier. „Im Endeffekt dauert es dann doch wieder ein halbes Jahr, bis die Nutzer unsere Software wirklich verwenden können. Denn die IT sammelt erst mal die Ergebnisse von ein paar Sprints, bis sie die Software als Endkundenrelease ausrollen. Sie sagen, das schaffen sie nicht

alle zwei Wochen. Und selbst wenn das Deployment schneller ginge. Unsere QA Abteilung braucht allein 3 Wochen für einen kompletten Testdurchlauf. Wenn die dann einen Bug finden, wird auf eine neue Version gewartet, die den Bug-Fix beinhaltet, aber auch die neuesten Funktionen. Und dann wird diese Version getestet. Es ist fast nicht vorhersagbar, wann wir eine Version haben werden, die unsere Test-Suite erfolgreich durchläuft.“

Herausforderung Produktivbetrieb

„Wenn die Software dann produktiv läuft, hören wir oft erst wieder was von den Admins, wenn es Fehler gibt. Neulich hatten wir große Probleme, da war die Performance unserer App plötzlich extrem schlecht. Verärgerte User hingen schon bei unserer Hotline in der Warteschleife. In einer solchen Situation klingelt dann bei uns das Telefon - und wir Entwickler sollen den Fehler sofort beheben. Also haben wir unsere reguläre Arbeit unterbrochen und hektisch versucht, die Anwendung im Produktionsbetrieb zu debuggen - und Ihr wisst ja, wie schwierig das ist. Da ist es dann aus mit der agilen Theorie.“

Ich nickte stumm - das Problem kam mir auch sehr bekannt vor. Peter schweig, schaute auf sein fast leeres Bierglas und winkte der Bedienung, um Nachschub zu beschaffen. Mein Blick fiel auf mein Smartphone.

Zehn Releases am Tag?

„Tja, Facebook, Amazon oder Flickr - da müsste man arbeiten“, meinte ich. „Die können angeblich zehn Mal pro Tag oder häufiger neue Funktionen an die End User ausliefern¹. Das ist cool!“

Peter war wenig beeindruckt: „Mit diesen Megakonzernen können wir uns doch nicht vergleichen. Bei uns ist so etwas einfach nicht drin.“

Ich wurde nachdenklich. „Eigentlich geht es doch gar nicht darum, wirklich

„DevOps ist die nächste Stufe der Agilität!“

T. SCHISLER

U. BAUMANN

zehn Mal am Tag neue Funktionen bereitzustellen. Das brauchen vielleicht echt nur die Internet-Giganten. Aber es wäre schon hilfreich, wenn wir in der Lage wären, sozusagen auf Knopfdruck sehr schnell eine neue Funktionalität in Produktion zu bringen.“

Marie - unser Usergroup-Urgestein und leidenschaftliche Web-Programmiererin - hatte sich bis jetzt neben uns etwas müde und lustlos an ihrer Cola festgehalten. Jetzt wurde sie aber plötzlich wach. „Erzählt mir was Neues!“, sagte sie resigniert. Peter und ich schauten sie an und Marie klagte uns ihr Leid.

Paradebeispiel Hotfix

„Es hat seinen Grund, dass ich heute Abend so spät zu unserem Usergroup-Treffen gekommen bin. Wir hatten in der Firma nämlich den Super-GAU. Ein Kunde hatte durch Zufall ein Sicherheitsloch gefunden - direkt in unserer Kundenverwaltung!

Und das, nachdem die Nachrichten voll sind von Datenklau-Skandalen bei allen möglichen Firmen. Unser Entwicklungsleiter wurde weiß wie Schnee...Wir haben alles liegen lassen und sofort einen Hotfix zusammengeschustert.

Als der fertig war, erhielten wir ausnahmsweise die Erlaubnis, für das Deployment eine Abkürzung zu nehmen - normal haben wir natürlich auch einen Genehmigungsprozess wie bei Peter. Kurz und gut - wir haben eine DLL mit dem Hotfix direkt auf die Produktionsserver kopiert. Es musste ja alles sehr schnell gehen.

Damit fingen die Probleme aber erst richtig an - die DLL war nämlich nicht kompatibel zur bisherigen Version! Jemand hatte inzwischen ein paar Interfaces geändert - und jetzt stand das ganze System! Hätten wir nur unsere Testsuite mal drüber laufen lassen. Naja, ich mach's kurz, nach vier Stunden liefen die Server wieder. Das war vielleicht ein Alptraum!“

Marie verstummte und sah Michael an, unseren immer gut gelaunten Usergroup-Sunnyboy. „Was gibt's denn da zu grinsen? Solche Probleme kennst Du wohl nicht, oder?“

DevOps: Nurein neuer Hype?

„Oh doch!“, gab Michael zurück. „Das kenne ich sehr wohl. Ich wollte auch nicht arrogant wirken. Es ist nur so: Bei unserem aktuellen Projekt haben wir in

meinem Team mal versucht, mehr auf DevOps zu setzen. Seitdem ist alles anders.“

„Ja, klar, noch ein hippestes Buzzword, das braucht die Welt!“, mischte sich Peter ein. Inzwischen war aus unserem Gespräch eine kleine Diskussionsrunde geworden und es hatten sich sogar Leute von den anderen Tischen dazugesellt.

„Sicher wäre es gut, wenn Developer und Admins besser zusammenarbeiten würden. Um das zu verstehen brauche ich keinen neuen Hype.“

Michael nickte: „Ja, ich dachte auch erst, es geht einfach nur um die Verbesserung der Zusammenarbeit von Entwicklung und Betrieb. Aber als wir uns im Team damit auseinandergesetzt haben, wurde schnell klar: DevOps hat zwar auch mit guter Zusammenarbeit von „Devs“ und „Ops“ zu tun, ist viel aber mehr als das!

Große Worte, ich weiß. Wir haben damit schrittweise angefangen und zuerst unseren gesamten Release-Prozess mit Visual Studio Release Management² automatisiert. Wenn wir einen Release Build durchführen, wird der Build automatisch mit vorgefertigten Skripten erst auf die Dev-Umgebung, und dann auf die Staging- und schließlich auf die Produktionsumgebung aufgespielt. Das heißt, wir testen damit nicht nur unsere Software, sondern auch unseren Deployment-Prozess.

Mit Release Management können wir sogar den Genehmigungsprozess für die einzelnen Stufen definieren. Für einige zentrale Vorgänge gibt es einen Verantwortlichen, der den jeweiligen Deploymentsschritt genehmigen und prüfen muss. Das gilt bei uns zum Beispiel für die endgültige Produktionsfreigabe. Die entsprechende Person bekommt vom System eine E-Mail und kann die Freigabe auf einer speziellen Website erteilen oder zurückweisen. Andere Schritte sind automatisiert, werden aber auch

Grundprinzipien von DevOps

DevOps beschäftigt sich mit dem Thema, wie Software nicht nur besser gebaut, sondern auch bereitgestellt und betrieben wird. Der Begriff ist relativ neu und wird oft nur mit der kontinuierlichen Bereitstellung gleichgesetzt. Doch diese Definition greift zu kurz. In der Praxis erfordert der DevOps-Ansatz, dass die gesamte IT-Organisation in einen vollständigen Lifecycle (ALM) investiert. Ziele dabei sind:

- » Die Durchlaufzeiten zu minimieren (d.h. den Zeitraum vom Beginn der Implementierung eines Features bis zur tatsächlichen Verfügbarkeit für den Anwender zu verkürzen)
- » Die Ressourcen durch automatisiertes Ausrollen von Systemen und Updates zu optimieren
- » Qualität und Verfügbarkeit der Anwendungen mit durchgängigen Tests und Überwachung sicherzustellen

„DevOps ist die nächste Stufe der Agilität!“

T. SCHISLER

U. BAUMANN

automatisch protokolliert. Man sieht jederzeit, welches Release von wem angestoßen wurde, und wer wann was freigeben oder testen muss.“

Release-Automatisierung als Basis

„Die Skripte für den Deploymentprozess haben wir in Zusammenarbeit mit unseren Admins entwickelt – zusammen mit Rollback-Skripten. Diese versetzen den Server in den ursprünglichen Zustand, falls bei einem Deployment irgendetwas schief geht. So ähnlich wie bei Datenbank-Transaktionen. Im Build laufen natürlich Unit Tests, Integrationstest und auch die Coded UI Tests, die bei uns die wichtigsten Funktionalitäten abbilden. So haben wir eine hohe Sicherheit, dass ein neues Deployment keinen Schaden anrichtet.“

„Macht Sinn! Hätten wir dieses automatisierte Deployment auch bei uns gehabt, wäre mein Tag anders verlaufen.“ meldete sich Marie. „Zum Beispiel hätten die Integrationstests wegen der inkompatiblen DLL Alarm geschlagen – und wir hätten durch die Skripte trotz des Zeitdrucks wahrscheinlich keine Fehler beim Ausrollen des Fixes gemacht.“

Peters Miene erhellte sich: „Also braucht Ihr die Admins nicht mehr? Alles wird per Skript installiert?“

„Nein, das würde ich nicht sagen. Die Rolle unserer Admins hat sich verändert. Wir sind jetzt ein Team und arbeiten eng zusammen. Denn das Ganze ist auch eine Kulturfrage. Wir tauschen uns viel mehr und viel früher darüber aus, wie sich eine Applikation installieren, warten und überwachen lässt.“

Kleine Schritte machen

Er hielt kurz inne und überlegte. „Aber es stimmt schon, die Automatisierung hilft und ist ein sehr wichtiger Aspekt. Dadurch, dass unser Deployment jetzt so rund läuft, ergeben sich viele andere Vorteile. Nicht nur beim Einspielen von

Bugfixes. Wir liefern jetzt wirklich viel öfter an die Endkunden aus, dafür kleinere Päckchen. Wir können so auch neue Features in der echten Welt testen. Und bei kleinen Schritten ist es einfach unwahrscheinlicher, dass wir einen fatalen Fehler einbauen. Wir haben auch ganz einfach mehr Übung beim Deployment! Das ist für uns kein großes Ding mehr.“

Build, Measure, Learn

Michael blickte in die Runde. Er hatte unsere volle Aufmerksamkeit. „Aber das reicht uns nicht.“ fuhr er fort. „Wir wollen viel genauer verstehen, welche Features wie oft genutzt werden, und wie sich unsere Anwendung im realen Betrieb verhält. Mit IntelliTrace in Production³ können wir Fehler im Produktivbetrieb sehr schnell analysieren.“

Außerdem testen wir gerade Application Insights⁴. Mit diesem Framework können wir unsere Webanwendung mit wenig Aufwand instrumentieren. Wir erhalten im laufenden Betrieb Echtzeitdaten über alle möglichen Aspekte unserer Applikation – von der Performance über die Häufigkeit, mit der bestimmte Funktionen genutzt werden bis zu auftretenden Exceptions. Allerdings gibt es im Team noch einige Diskussionen, denn diese Telemetriedaten laufen über den Microsoft-Server. Das möchte nicht jeder. Zudem ist die Technologie momentan erst als Preview verfügbar. Mal sehen, wie sich das entwickelt – ansonsten gibt es auch von anderen Herstellern leistungsfähige Alternativen.“

Peter hatte jetzt auch Feuer gefangen: „Wow – also könnt Ihr im wörtlichen Sinn morgen sehen, was Eure Änderungen von heute bewirken – wie gut sie angenommen werden, aber auch wie gut der Code wirklich läuft!“

„Ja“, ergänzte Michael „eben nach dem Motto ‚Build, Measure, Learn‘. Etwas bauen, dann messen was passiert, um daraus wiederum zu lernen. Das ergibt einen richtigen Kreislauf der Verbesserung!“

Lohnende Herausforderungen

„Okay, aber ist es nicht eine dumme Idee, neue Features gleich an alle Kunden auszuliefern und diese sozusagen als Versuchskaninchen zu benutzen?“ konterte Marie. „Ziemlich mutig, oder eher unverantwortlich, würde ich sagen.“

Michael grinste: „Ja, da hast Du recht. Aber wir sind dabei, unseren Code so umzuschreiben, dass wir neue Funktionen per Konfiguration ein- und ausschalten können. Mit Feature Flags – da gibt es übrigens schon einige Frameworks dafür bei NuGet. Der Plan ist, dass wir die neuen Features erst für einige wenige Kunden aktivieren. Falls alles gut läuft, schalten wir immer mehr Nutzer frei. Das sehen wir ja durch die Telemetrie. Das nennt sich Test in Production. Noch haben wir da ein Stück Arbeit vor uns, denn wir müssen dafür die Architektur unserer Anwendung erst noch fit machen. Aber ich denke der Aufwand lohnt sich. Kommerzielle Services wie Visual Studio Online gehen schon länger so vor.“

Marie war noch immer skeptisch: „Denkt Ihr nicht, dass die Kunden verstört reagieren, wenn ständig neue Features einfach so dazukommen, ohne dass man sich wie bisher bei einem großen Release darauf einstellen kann?“

„Ja, das stimmt – die Leute müssen umdenken“ stimme Michael zu. „Aber diese Art, neue Funktionalität auszuliefern, ist für die Nutzer eigentlich bekannt – schließlich hat so gut wie jeder ein Smartphone. Für die meisten Apps gibt es sehr häufig neue Versionen, die meist ohne Nachfrage automatisch installiert werden. Ich denke sogar, dass diese schnellen Releasezyklen von den Nutzern in Zukunft bei jeder Art von Software erwartet werden – Stichwort Consumerization of IT.“

Peter grinste: „Genau! Wäre ja noch schöner, wenn man ein CD-Laufwerk

„DevOps ist die nächste Stufe der Agilität!“

T. SCHISLER

U. BAUMANN

ans Handy anschließen müsste und die Software alle Schaltjahre auf CD geliefert würde. Aber bei vielen klassischen Anwendungen ist das im übertragenen Sinn immer noch so! Bei Webanwendungen oder bei lokalen Anwendern ist das Problem, die Software auszuliefern ja auch nicht so groß.“

Michael nickte zustimmend. „Dabei gibt es schon lange Möglichkeiten, auch normale Client-Software unter Windows einfach und sicher regelmäßig upzudaten. ClickOnce ist der Klassiker unter diesen Frameworks. Aber es gibt mittlerweile auch genug andere Möglichkeiten, die Funktionalität an die User zu liefern, z.B. per Push-Verteilung über NuGet. Für diese Aufgaben gibt es Open Source-Bibliotheken. Oder man löst das mit eigenem Code.“

Wir haben zum Beispiel als Konzeptstudie einen kleinen Prototyp geschrieben, der das Managed Extensibility Framework und Windows Communication Foundation-Services nutzt und Softwarekomponenten im laufenden Betrieb austauscht.“

„Könnte man damit sogar Software, die im 24/7-Dauerbetrieb im Einsatz ist, im laufenden Betrieb updaten?“ fragte Peter.

„Das würde gehen! Auch das machen die großen Internetkonzerne ja heute schon. Schließlich können die bei diesen häufigen Update-Zyklen nicht immer das System offline nehmen. Aber es gibt schon einige Herausforderungen, wenn man den DevOps-Ansatz konsequent

bis auf dieses Niveau verfolgt. Security ist ein Thema – es darf auf keinen Fall möglich sein, dass Angreifer von außen Software in den Releaseprozess einschleusen können. Und wie ich vorher schon gesagt habe: Um wirklich voranzukommen ist es nötig, ganzheitlich zu denken – von der Software-Architektur über die richtigen Tools und Frameworks muss alles passen.“

Aber die Chancen, die sich durch DevOps ergeben, sind so groß, dass es sich lohnt, konsequent an den Herausforderungen zu arbeiten. Dann könnte man viele Vorgehensweisen, die in der Welt der Smartphones schon Standard sind, auf andere Applikationen und Anwendungsfälle ausweiten.“

Neue Chancen und Geschäftsmodelle

Denkt mal an einen Werkzeugmaschinen-Hersteller. Dessen Software läuft auf Maschinen, die auf der ganzen Welt verteilt sind. Dort wäre es ein fantastischer Vorteil, wenn diese Software automatisch und ferngesteuert ausgetauscht werden könnte – womöglich ohne die Produktion unterbrechen zu müssen! Da würden sich ganz neue Geschäftsmodelle ergeben, denn die Firma könnte das als zusätzliche Vertragsoption zur Hardware anbieten. Wir haben einige Kunden, die von dieser Idee total begeistert sind! Natürlich ist das bei Industriemaschinen nicht so einfach wie beim Handy, da gibt es neben den erhöhten Security-Anforderungen noch ganz andere Themen – zum Beispiel müssen dort formale Freigabeprozesse eingehalten werden und vieles mehr.“

Die nächste Stufe der Agilität

Die Diskussion war lange noch nicht zu Ende, und wir wurden schließlich vom Wirt der Gaststätte um halb eins herauskomplimentiert.

Marie brachte es schließlich auf den Punkt: „Jetzt verstehe ich auch, was ich neulich gelesen habe! Jemand meinte, DevOps wäre die nächste Stufe der Agilität. Eigentlich ist es ja eine logische Fortsetzung der agilen Prinzipien: Kurze Zyklen, Fokus auf die Realität, immer dazulernen.“

„Tja, apropos Lernen“ meinte Michael noch im Hinausgehen – ich hab da noch einen Lesetipp für Euch: „The Phoenix Project“⁵ – ein spannendes Buch über DevOps, das die ganzen Prinzipien erklärt, und zwar in Form einer fesselnden Geschichte.“

Tja, und ich musste zurück in die Realität, zurück in mein Team. Denn am Montag ist der große Release-Tag. Big Bang.

Hoffentlich geht alles gut! Und wenn nicht: In Michaels Team ist noch eine Stelle frei...

[1] „10+ Deploys Per Day: Dev and Ops Cooperation at Flickr“: <http://tinyurl.com/k3xfb32>

[2] Visual Studio Release Management Produktwebsite: <http://tinyurl.com/ol9afzl>

[3] Using IntelliTrace in Visual Studio 2015: <http://tinyurl.com/nv4zf2p>

[4] „Get started with Azure Application Insights“: <http://tinyurl.com/nf4jehq>

[5] Gene Kim, Kevin Behr, George Spafford: „The Phoenix Project. A Novel About IT, DevOps, and Helping Your Business Win“: IT Revolution Press (2013). ISBN 978-0988262591



artiso solutions GmbH

Oberer Wiesenweg 25
89134 Blaustein

E-Mail: service@artiso.com
Telefon: +49 (0)73 04 / 803-0
Fax: +49 (0)73 04 / 803-300
Web: www.artiso.com

Agilität ist unsere Leidenschaft und bestimmt unser Tun und Handeln. Die gesamte Unternehmensphilosophie der artiso solutions GmbH basiert auf dem Bestreben, qualitativ hochwertige Software-Produkte mithilfe selbst-organisierter Teams und moderner Entwicklungsmethoden und Technologien zu entwickeln.

Profitieren auch Sie von unserem hocheffizienten Vorgehensmodell, indem Sie Ihre Software-Lösung durch unsere Teams erstellen lassen und wir Sie dabei unterstützen, Ihre Organisation und Ihre Teams durch Agilität noch effizienter und erfolgreicher zu machen!

Für artiso spricht:

- » Seit 1991 Erfahrung in der Entwicklung von IT-Lösungen in der Microsoft .NET Welt
- » Führendes ALM-Integrationsunternehmen; erfahrener Berater und Begleiter bei der Einführung des TFS und dessen optimaler Nutzung
- » Microsoft Gold Partner ALM und Application Development
- » Mitglied im Microsoft Developer Quality Board
- » Anerkannte Experten (Microsoft MVP für Visual Studio ALM, zertifizierte Professional Scrum Trainer, Autoren, gefragte Sprecher auf den etablierten Fachkonferenzen)
- » Bekannte Unternehmen wie die Daimler AG, Sartorius AG, AD-AC, ratiopharm und die Uzin Utz AG vertrauen artiso