

A ZOMBIE'S SURVIVAL GUIDE IN A CHANGING DATABASE WORLD

Is the database administrator
dead? Or alive and
preparing for the future?

By Pini Dibask, Product Manager for Database Performance Monitoring, Quest Software, and Dave B. Awesome, Database-Administrator-at-Large

Quest®



Table of Contents

Introduction	3
Trend 1: Yesterday's DBA is today's DevOps engineer	6
Trend 2: The rise of multi-platform environments	12
Trend 3: The cloud and Database-as-a-Service: Your ticket to influence	16
Conclusion	20

Introduction

Now that we have your attention ...

Is the DBA dead?

Or are your dreams of that steady role as a DBA dying as you struggle to keep up with how quickly the landscape of database administration is changing?

What if that futurist at Cisco was right, and 127 new things really are connecting to the internet every second?¹ That works out to 328 million things per month: drinking cups, tennis rackets, hoodies, thermometers, railroad cars, cows, pigs, vehicles, diapers (yecch!), sky-scrappers, door locks and garbage cans, not to mention whatever products and services your company sells. Some or all of those things are going to start sending data into your organization, and just the thought of keeping abreast of what it takes to monitor and protect those new kinds of data can be enough to suck the soul right out of you.

Or maybe you feel like one of the undead or near-dead when you think about how much the database management landscape has changed in the past decade and how much

¹ Evans, David, "Introducing the Wireless Cow," Politico, June 2015, <https://www.politico.com/agenda/story/2015/06/internet-of-things-growth-challenges-000098>



Quest



it's going to change in the next decade. Sure, you've got your Oracle and SQL Server certs so you'll always have work. But then you read a few articles and see that search engines like Elasticsearch and Splunk and new platforms like MongoDB and PostgreSQL are eating the world.

"Ugh," you think, "look what all the cool kids get to work on."

Or are you a workplace zombie, hunched over your keyboard, staring at three monitors and working on database performance tuning all day long? Let's see ... what's on your task list again?

- Fix problems users have reported.
- Find problems and fix them before users report them.
- Repeat.

And you've only been doing that for like 900 years. I don't blame you for wondering if you're dead. Or undead, even.

So Pini Dibask asked me to tell the story that will help you get past this zombie stage of database administration. We've put together this Zombie's Survival Guide in a Changing Database World, based on his webcast with

Quest

Peter O'Connell titled “Is the DBA Dead? ... or alive and preparing for the future.”²

OK, so there’s some doom and gloom out there about the future of the DBA. But in this e-book, you’ll learn about the top three trends — DevOps, multi-platform databases and cloud-based databases — that affect DBAs like us. You’ll get some real-world perspectives on the evolving role of the DBA.

If you can shake off the DBA-pocalypse, you’ll see how to adapt, evolve, survive and even thrive in a database world that is now and will forever be in flux.

You have nothing to lose but your zombie-hood.

² MSSQLTips, April 2018, <https://www.quest.com/webcast-on-demand/is-the-dba-dead-or-alive-and-preparing-for-the-future8131529/>



Trend 1: Yesterday's DBA is today's DevOps engineer

We went through a reorg not long ago, and when they sent out the new org chart, guess what? I got a title change from DBA to DevOps engineer. What do I do that's different from before? Not much. Most of what I do hasn't changed at all. But my cubicle is ground zero for DevOps now, so it's the first trend I want to emphasize.

It turns out that, as DBAs, we've always been part of DevOps. That's because most of us already do Dev — like writing SQL, tuning performance, doing Object Analysis and reporting — and we already do Ops — like configuring servers, running backups and restores, and tuning the OS, network and storage.

The main change is that DevOps often pits application developers against infrastructure teams over stability and performance issues. Dev and Ops have different priorities that require a fine balancing act.



THE DEV WAY

The Dev side of the house is measured on how quickly and reliably they can get changes into production. Their goal is to develop code to spec as fast as possible, get it into production and keep those changes coming. Our job as DevOps engineers is to think “continuous:” continuous deployment, continuous integration, continuous delivery.

For applications, that works just fine. Even though it’s continuous, it’s sequential, and you’ve got relatively stable parts to that process as you put it into production. Application code becomes like boxes on a conveyor belt, as shown in Figure 1.

And here’s what’s most important for us: Say you release version 5.4.3 of your application and then find a bug in it. You can release version 5.4.4 with the fix or you can go back to 5.4.2, which didn’t have the bug. Life is great when you have those options.

“Hurry it up,” says Dev to Ops. “We’re kicking agile and taking names, and you’re slower than Netflix over dial-up. When are you going to get your automation-and-collaboration act together so our whole organization can accelerate delivery and leave our competitors in the dust?”

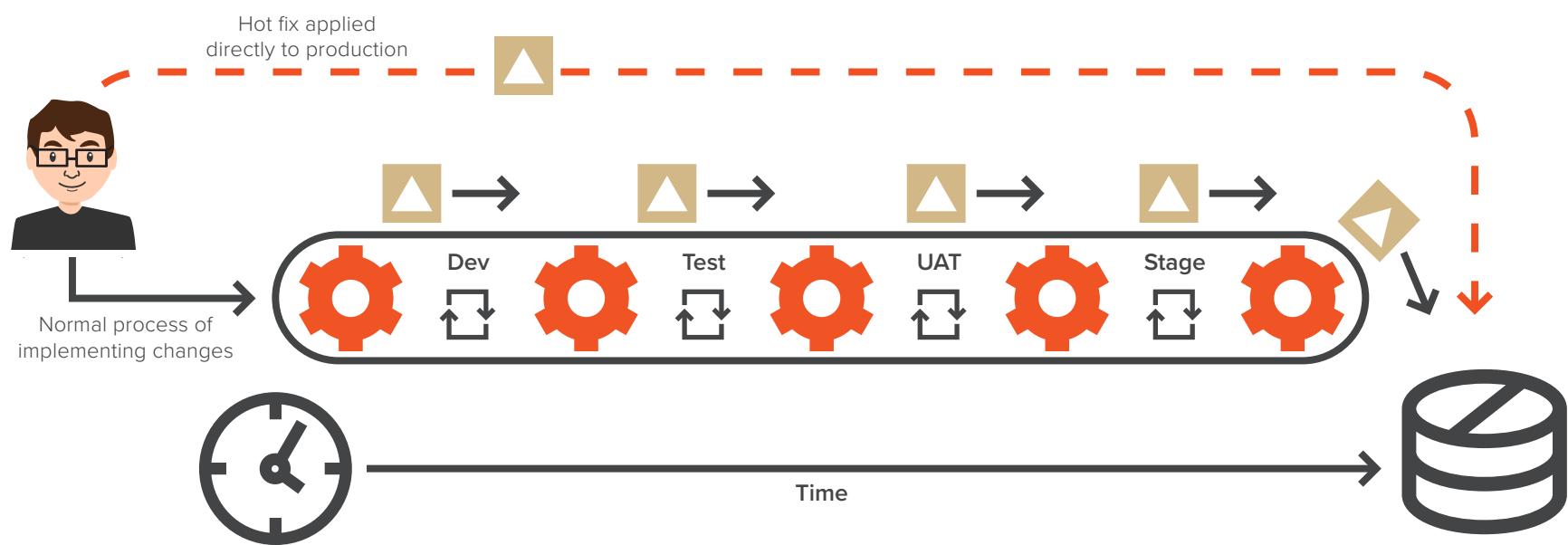


Figure 1: The sequential life in Dev

THE OPS WAY

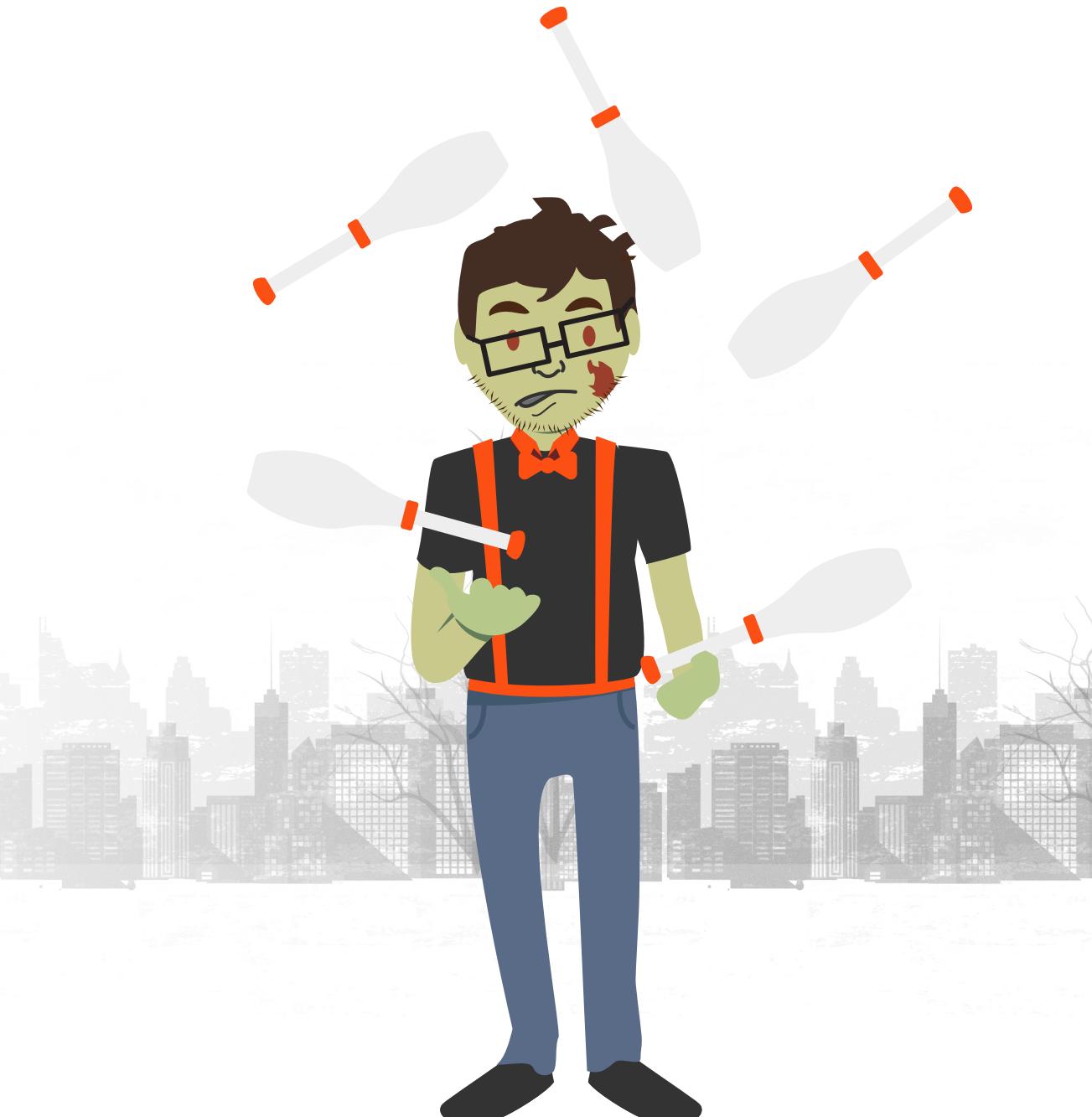
As for Ops — well, not so much.

Ops is concerned with making sure that things don't break when you take your eye off them. It's about all the fundamentals that keep the lights on and the data moving. Fast and continuous change is not a welcome visitor on the operations side. In Ops, our job is to maintain stability through diagnostics, tuning and administration. The only thing we do continuously is to monitor.

That's what databases need because changes aren't sequential, they're all over the place. In many ways it's the juggling act you are expected to perform every day.

Undoing changes in a production database is like changing the DNA of a rabbit. One does not simply revert to an earlier state as in application development because data has changed in a zillion different places and the organization depends on the changes. There's no undoing the changes and going back to how things were three days ago.

"You don't have to be as careful and monitor as much as we do," Ops replies to Dev. "If we were able to use all the automation tools you use, we could catch up to you no sweat. But if we start automating the deployment of database changes, things are liable to go missing. Starting with your paychecks."



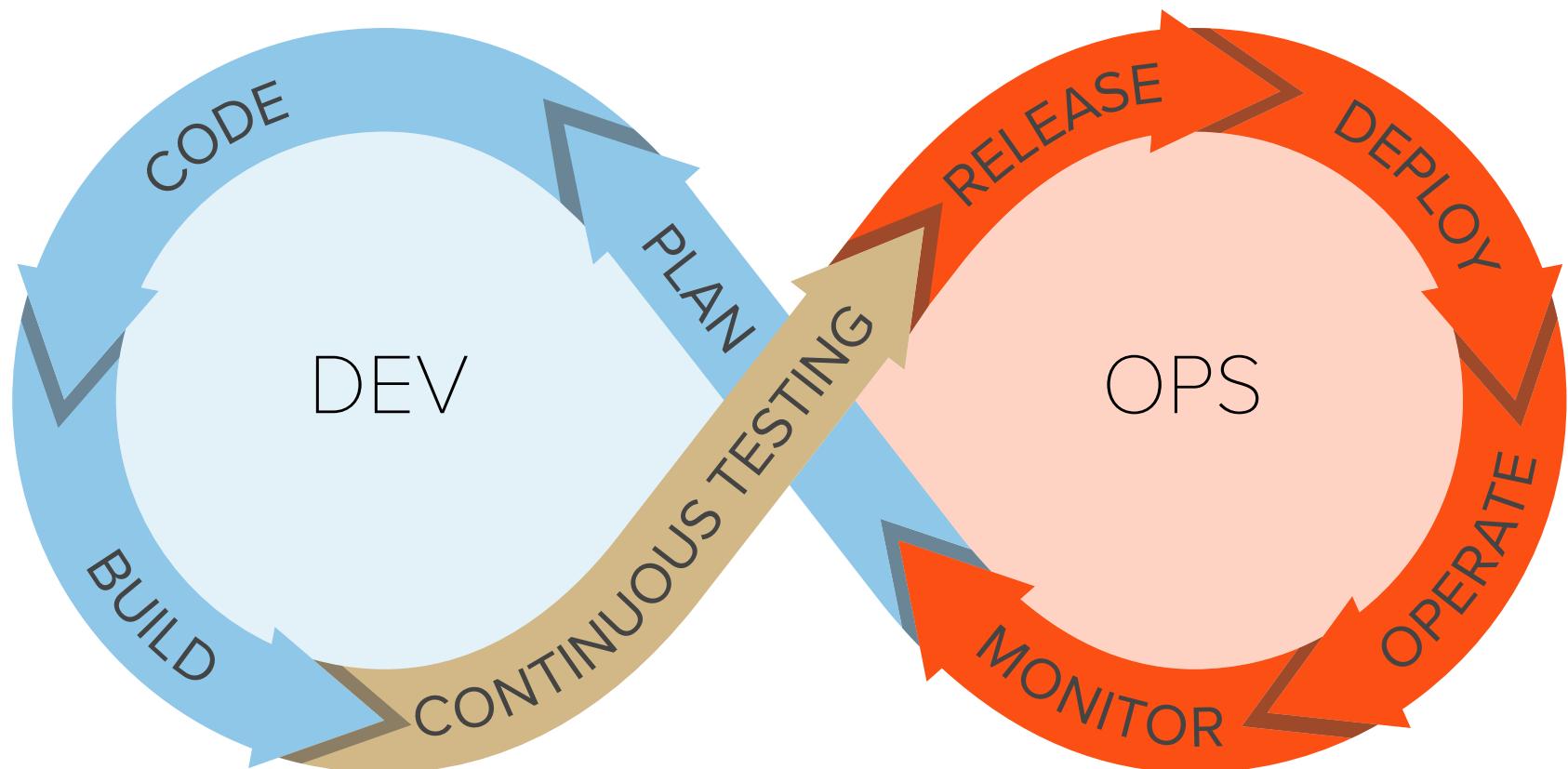


Figure 2: The DevOps loop

DEVOPS: CONTINUOUS DELIVERY, CONTINUOUS INTEGRATION, CONTINUOUS DEPLOYMENT

Much of DevOps is based on the motto, “Fail often, fail fast.” Make small, incremental changes and push them into production. If they cause a problem or if they’re not quite what customers wanted, then roll them back quickly, repair them and get them back into production pronto. As shown in Figure 2, change and improvement are continuous; they’re not reserved for just the testing phase.

At the other end from continuous delivery is the continuous monitoring that we DBAs specialize in. Are we maintaining the same performance benchmarks? Does everything work as intended? When something goes wrong, can we fix it quickly? Can we adapt to changing app scenarios? Can we map a change observed in production against some schema change?

The loop runs continuously back into Dev to design and build differently.

ON BEING THE DEVOPS ENGINEER

That's why DBAs have a big role to play in reconciling Dev and Ops. In fact, it's a big opportunity for DBAs to grow into the role of DevOps engineer.

It's not always easy to understand the root cause of a performance or stability problem. It takes digging to connect the dots between a change somebody made to the application or the infrastructure and the impact it has on the database workload and application performance. We'd love to automate the integration of database changes, then say, "Never mind!" and roll back to an earlier version when something goes haywire, but we can't. Besides, there are a lot more automated test tools for application development than for database development. So, we monitor continuously.

In fact, a survey by DBTA polled professionals in application development/deployment and IT operations about DevOps, asking what was most difficult in integrating database changes to DevOps.³

As you see in Figure 3, automated regression testing is no slam-dunk; almost half of us

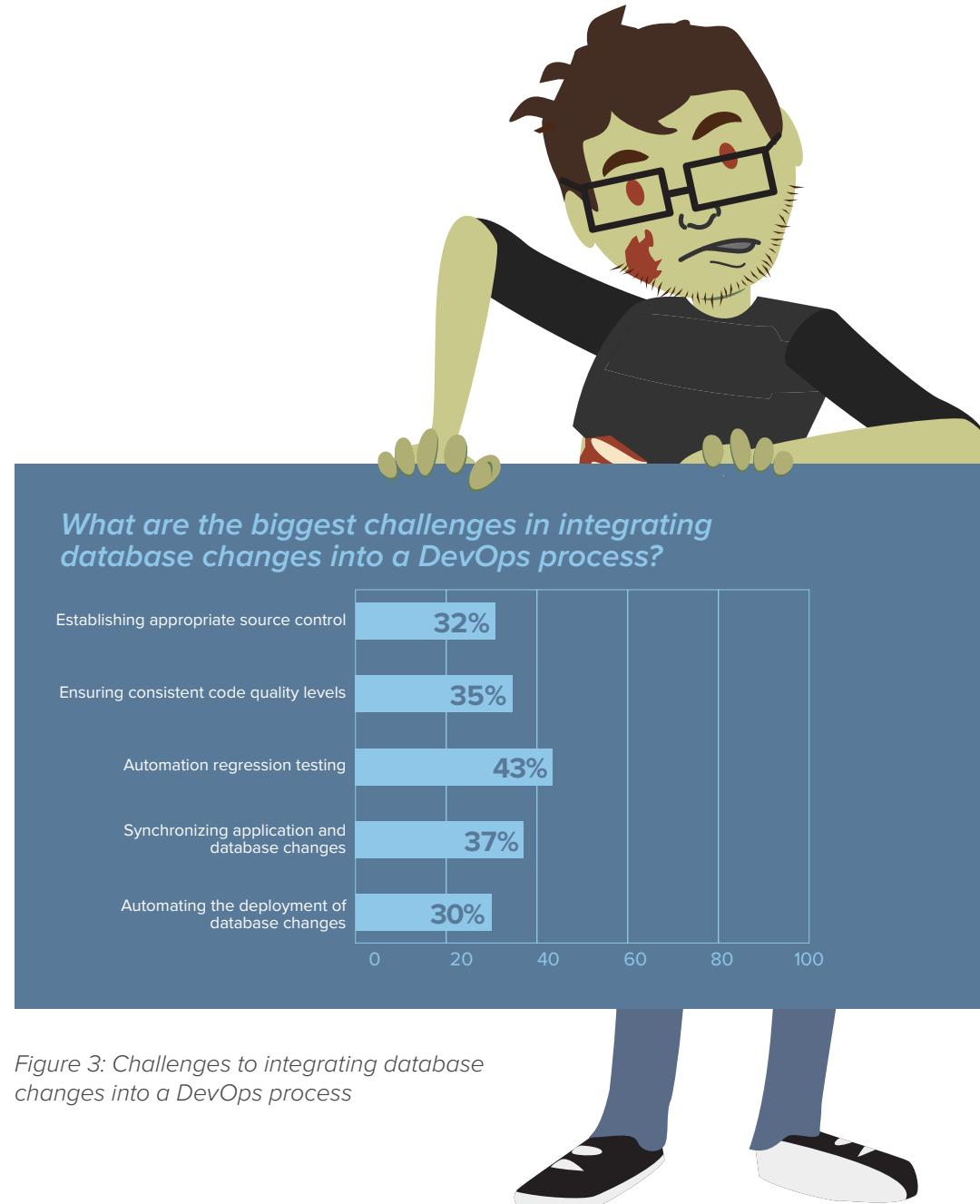


Figure 3: Challenges to integrating database changes into a DevOps process

³ King, Elliot, "The Current State and Adoption of DevOps," Unisphere Research sponsored by Quest Software, September 2016, <https://www.quest.com/whitepaper/the-current-state-and-adoption-of-devops81f6848/>



are trying to figure out how to make it work. Of course, synchronizing application and database changes is the crux of DevOps, and we can't take our eye off of consistent code quality.

It's all still greenfield, so once you awaken from your zombie state and realize you're not dead yet, that bright future as a DevOps engineer will look pretty good to you. We advise our customers to start with small, bite-sized projects, one process at a time, instead of trying a deep dive into something like rapid deployment right off the bat. Internally at Quest, we've been at it about five years, and only now are we ready to undertake continuous deployment.

Fail often, fail fast and continuously improve.



Figure 4: It takes all kinds of database platforms including Oracle, SQL Server, SAP ASE, DB2, PostgreSQL, MySQL, MongoDB and Cassandra.

Trend 2: The rise of multi-platform environments

The second trend coming straight for you is the rising overall number of database platforms and technologies. Accomplishing anything worthwhile takes a lot of them.

And they're not waiting for you to get un-dead.

THE DATABASE LANDSCAPE

The enterprise war horses like Oracle and SQL Server are still around, and you'll always have work if you know them. But who knew MySQL had so much staying power? Why didn't you spend your evenings learning it when you had the chance, instead of playing Minecraft all night?

And then you look at DB-Engines or a job posting (just accidentally, of course) and see that the database landscape is filled with categories like wide-column stores, search engines, key-value stores, graph database management systems and document stores.

Rank	Jun 2018		DBMS	Database Model	Score		
	May 2018	Jun 2017			Jun 2018	May 2018	Jun 2017
1.	1.	1.	Oracle +	Relational DBMS	1311.25	+20.84	-40.51
2.	2.	2.	MySQL +	Relational DBMS	1233.69	+10.35	-111.62
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1087.73	+1.89	-111.23
4.	4.	4.	PostgreSQL +	Relational DBMS	410.67	+9.77	+42.13
5.	5.	5.	MongoDB +	Document store	343.79	+1.67	+8.79
6.	6.	6.	DB2 +	Relational DBMS	185.64	+0.03	-1.86
7.	7.	↑ 9.	Redis +	Key-value store	136.30	+0.95	+17.42
8.	↑ 9.	↑ 11.	Elasticsearch +	Search engine	131.04	+0.60	+19.48
9.	↓ 8.	↓ 7.	Microsoft Access	Relational DBMS	130.99	-2.12	+4.44
10.	10.	↓ 8.	Cassandra +	Wide column store	119.21	+1.38	-4.91
11.	11.	↓ 10.	SQLite +	Relational DBMS	114.26	-1.19	-2.44
12.	12.	12.	Teradata	Relational DBMS	75.77	+1.36	-1.55
13.	↑ 14.	↑ 18.	MariaDB +	Relational DBMS	65.85	+0.85	+12.95
14.	↓ 13.	↑ 16.	Splunk	Search engine	65.78	+0.68	+8.26
15.	15.	↓ 14.	Solr	Search engine	62.06	+0.55	-1.55
16.	16.	↓ 13.	SAP Adaptive Server +	Relational DBMS	61.49	-0.02	-6.04
17.	17.	↓ 15.	HBase +	Wide column store	59.70	-0.25	-2.17
18.	18.	↑ 20.	Hive +	Relational DBMS	57.33	+0.36	+12.95
19.	19.	↓ 17.	FileMaker	Relational DBMS	56.18	+1.51	-0.90
20.	20.	↓ 19.	SAP HANA +	Relational DBMS	49.35	+0.97	+1.85
21.	21.	↑ 22.	Amazon DynamoDB +	Multi-model ↗	45.79	+1.60	+11.78
22.	22.	↓ 21.	Neo4j +	Graph DBMS	41.97	+1.39	+4.10
23.	23.	↑ 24.	Memcached	Key-value store	33.80	+0.25	+5.07
24.	24.	↓ 23.	Couchbase +	Document store	32.45	+0.04	+0.54
25.	25.	25.	Informix	Relational DBMS	26.55	+0.76	-1.28
26.	26.	↑ 27.	Microsoft Azure SQL Database +	Relational DBMS	26.29	+1.09	+4.97
27.	27.	↑ 28.	Vertica +	Relational DBMS	21.16	+0.07	+0.25
28.	↑ 29.	↑ 30.	Firebird	Relational DBMS	20.39	+1.40	+1.24

Figure 5: DB-Engines.com database ranking

At Quest we're all about the multi-platform. We're building a SaaS application using an Azure SQL database, table storage, Redis and Document DB. We plan to then migrate some of that into Azure Cosmos DB as well and use its REST API. The choice of technology depends on factors like what we're doing, the consistency of the data and the performance that users need.

Why is multi-platform important? Because one size database no longer fits all. Oracle and SQL Server are still popular for online transaction processing (OLTP), ERP, CRM, data warehousing and online ratings. For web-based applications, it's MySQL, MongoDB and Cassandra. For distributed, multi-model databases covering multiple regions and their individual data regulations, look at Cosmos DB from Microsoft. As shown in Figure 5, it takes all kinds.

And one size DBA no longer fits all. As more platforms enter the landscape, DBAs are being asked to manage them. The same DBAs who used to manage just the relational databases now need to be flexible and be cross-trained to support other databases.

Put down that controller. You've got some homework to catch up on.

MULTIPLE PLATFORMS: HOW MANY AND WHY?

Without your even asking, your organization is nudging you down the career path of handling more categories and platforms (not to mention more instances of each). In another survey conducted by DBTA, about 50 percent of respondents reported that their DBAs are responsible for database management systems from two different vendors, with 26 percent saying their DBAs manage platforms from three to five vendors.⁴

That's the "how many." Another question in that same survey, summarized in Figure 6, asked the "why:" What motivated their organizations to use multiple database platforms?

You see that cost savings are part of the reason for keeping your eggs in more than one basket, but the biggest reasons for implementing multiple databases are to support multiple applications, workloads and groups of users.

That means that if we're going to stay smart and marketable, DBAs like you and me have to keep an eye on how tasks get executed in the business and monitor database performance to keep up with applications, workloads and user needs.

⁴ King, Elliot, "DBAs Face New Challenges: Trend in Database Administration," Figure 17, Unisphere Research, December 2017, <https://www.quest.com/whitepaper/dbas-face-new-challenges-trend-in-database-administration8130482/>

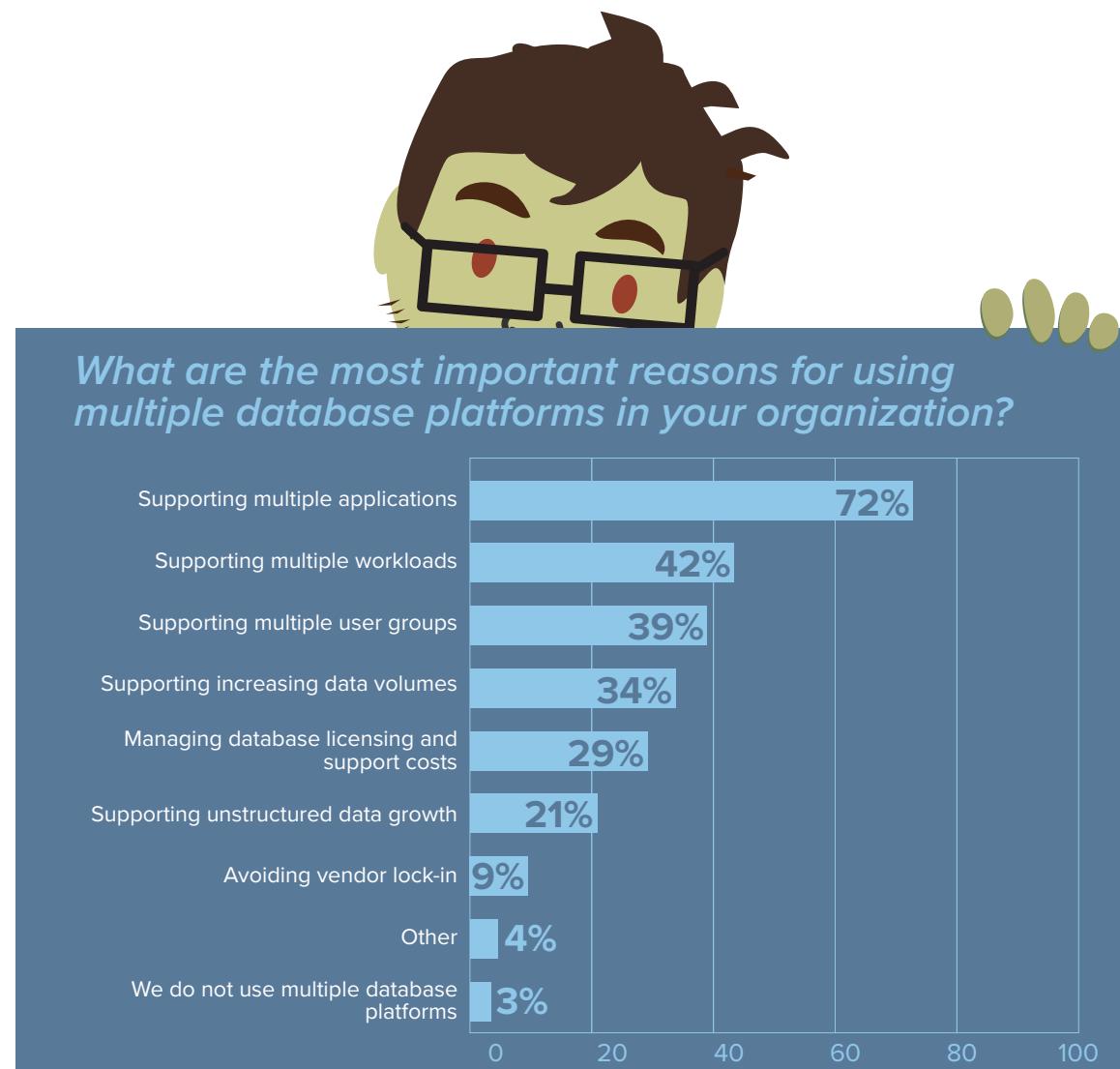


Figure 6: Reasons for using multiple database platforms

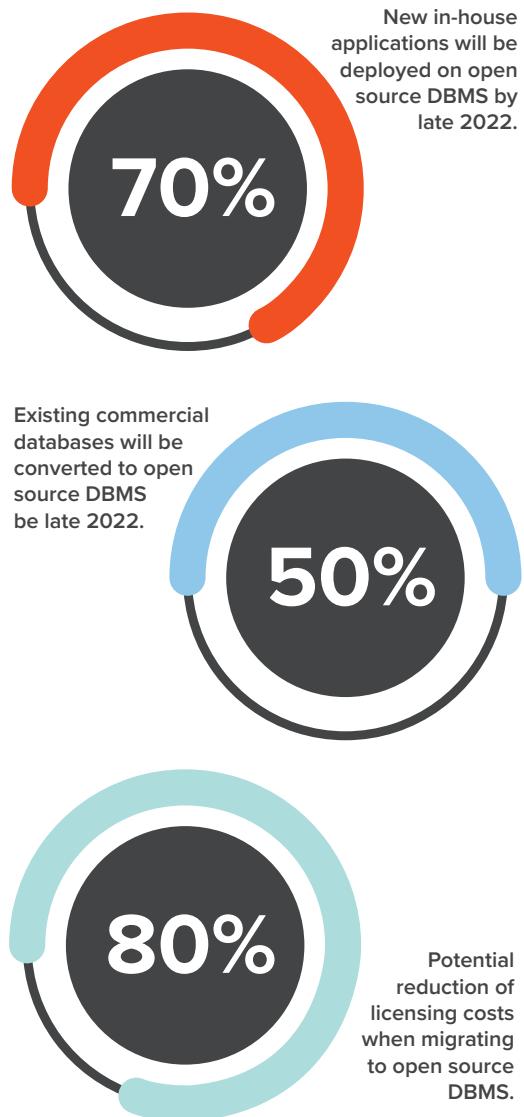


Figure 7: Advantages of open source database management systems

OPEN SOURCE DATABASES: “FREE” AS IN “FREE SPEECH,” NOT “FREE BEER”

When you start talking about multi-platform databases, before long you’re talking about open source.

Gartner believes that by 2022, “more than 70 percent of new in-house applications will be developed on an open-source database management system (OSDBMS), and 50 percent of existing commercial relational database management system (RDBMS) instances will have been converted or will be in process of converting.”⁵

That’s a lot of momentum toward open source.

Sometimes it’s about money. Companies that have used databases like Oracle and SQL Server have started letting open source databases in the door as a way of lowering licensing costs. Newer companies and younger DBAs dive straight into open source databases like MySQL, PostgreSQL and MongoDB. They’ve never used traditional databases and never even thought twice about it. In the same Gartner report, respondents estimated savings of “up to 80 percent of the cost of DBMS software” from using open source.

There’s a lot to love about open source, including that it’s free in some regards. But not like free beer.

In many cases, open source means that you are allowed to view, use and even improve the source code. But the vendors have to make money somehow, so they charge for technical support, security patches and updates, and for features like backup, high availability and disaster recovery.

Or, in the case of MySQL (an Oracle product), you may use the Community Edition for free, but you pay licensing fees for the Standard Edition and Enterprise Edition.

Do you need the open source database for commercial use? Some vendors include a no-cost license to use the product in your dev or test environment, but if you need the database in production, then they charge for the license. Who could blame them?

Still, open source databases are real. Once they had proved their resilience in non-mission-critical applications like dev and test, they came into their own in commercial applications. The opportunity is for us DBAs is to be the experts and to advise the business knowledgeably about how much open source costs and when our organizations have to pay.

⁵ Adrian, Merv and Feinberg, Donald, “State of the Open-Source DBMS Market, 2018,” Gartner, February 28, 2018, <https://www.gartner.com/doc/3863764/state-opensource-dbms-market>.

Trend 3: The cloud and Database-as-a-Service: Your ticket to influence

The third trend affecting us DBAs is the cloud, especially with the evolution of DBaaS.

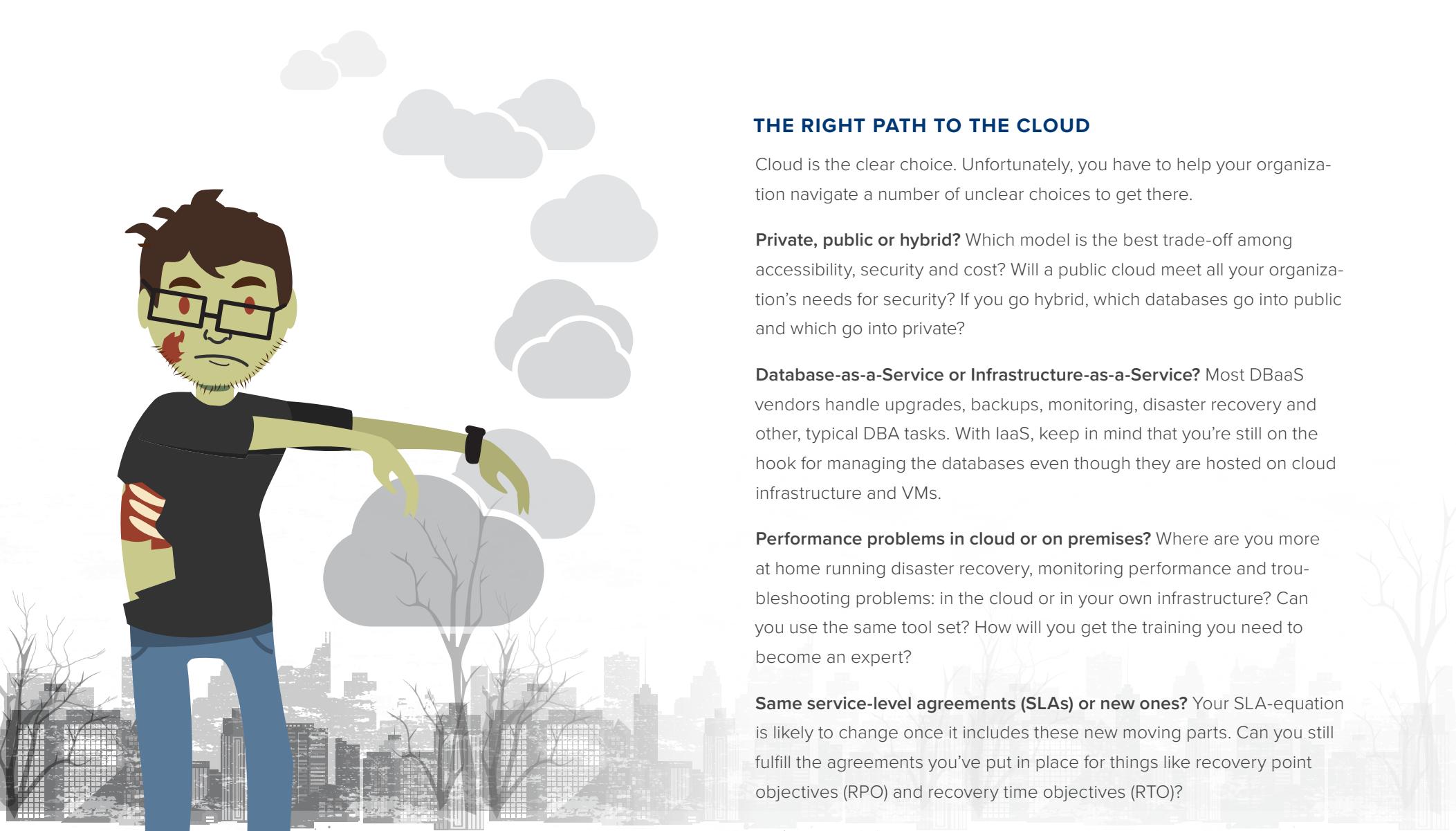
As I said about traditional databases, on-premises deployments are in no danger of going away; IDC predicts that about 80 percent of worldwide RDBMS revenue in 2020 will come from on-premises databases. But IDC also predicts that the other 20 percent of revenue will come from public cloud, on a growth trajectory of 41 percent per year.⁶ That's gonna be a whole lotta public cloud.

And with great cloud comes great responsibility. Besides the uptime, monitoring, maintenance and global coverage we keep an eye on with on-premises databases, we have to manage relationships with cloud vendors and watch out for cost overruns. Is that starting to feel oppressive to you?

In the end, though, if you can shake off the DBA zombie blues, you'll see that cloud and DBaaS are your ticket to new kinds of influence in the organization. If you play your cards right, you can get away from grinding out your workdays on the boring stuff and switch your focus to innovation and the business needs.



⁶ Olofson, Carl, "Worldwide Relational Database Management Systems Forecast, 2017–2021," IDC, June 2017, <https://www.idc.com/getdoc.jsp?containerId=US42750117>



THE RIGHT PATH TO THE CLOUD

Cloud is the clear choice. Unfortunately, you have to help your organization navigate a number of unclear choices to get there.

Private, public or hybrid? Which model is the best trade-off among accessibility, security and cost? Will a public cloud meet all your organization's needs for security? If you go hybrid, which databases go into public and which go into private?

Database-as-a-Service or Infrastructure-as-a-Service? Most DBaaS vendors handle upgrades, backups, monitoring, disaster recovery and other, typical DBA tasks. With IaaS, keep in mind that you're still on the hook for managing the databases even though they are hosted on cloud infrastructure and VMs.

Performance problems in cloud or on premises? Where are you more at home running disaster recovery, monitoring performance and troubleshooting problems: in the cloud or in your own infrastructure? Can you use the same tool set? How will you get the training you need to become an expert?

Same service-level agreements (SLAs) or new ones? Your SLA-equation is likely to change once it includes these new moving parts. Can you still fulfill the agreements you've put in place for things like recovery point objectives (RPO) and recovery time objectives (RTO)?

Which cloud? On top of all those choices comes your choice of vendor. Which criteria are most important to your organization? Reputation? Market share? Consistency with your existing infrastructure?

Whatever path your company may take to the cloud, be ready to research and document your recommendations. You're the expert.

YOU PUT THE LOAD, PUT THE LOAD, RIGHT ON ME

The fun doesn't stop with all of those choices. Once you've charted a path to the cloud, next come the responsibilities.

Earlier I mentioned that most of us DBAs have long been a part of DevOps. The Dev part is SQL tuning, creating reports and writing SQL statements, and the Ops part is infrastructure, with tasks like backup, monitoring and disaster recovery. With cloud and DBaaS, get ready for those responsibilities to extend to 24/7 coverage of global databases, paving the way for continuous deployment.

Also, who's minding the store? You may have hired cloud vendors, but you didn't hand them the keys and go on an extended vacation. You still have to answer for any problems with performance, capabilities, functions and SLAs, whether they're in the cloud or on your premises.

Speaking of the store, performance management in the cloud is closely tied to cost management, and now that your organization is being charged by the month, monitoring and controlling cost really matters. For example, if a poorly written SQL query is running 50 percent as fast as it should, then it's costing you 50 percent more because you pay directly for the resources it consumes.

The result is one of the biggest responsibilities of all: Your CFO now knows your name. Beyond management, performance, monitoring, disaster recovery and everything else a DBA does, now you're also responsible for costs. If you haven't done your homework, chosen the right path to cloud and followed it properly, then you're costing the organization money.

And that can strangle your dreams of that steady role as a DBA.



In which parts of the process of deciding to maintain databases in the cloud are DBAs in your organization involved?



Figure 8: DBA Involvement in cloud decisions

BE THE CLOUD WHISPERER

But like DevOps and multi-platform databases, cloud has a big, potential upside for your career as a DBA.

First, it isn't really a bad thing that the CFO knows your name and where the money is going. Anything that ensures DBAs a seat at the table helps lift your profile and heighten your reputation as the valuable member of the organization that you are, particularly in the age of cloud.

Next, have a look at Figure 8, summarizing responses to a pivotal question in a DBTA survey:⁷

It's no surprise that DBAs have influence in researching, implementing and maintaining cloud and DBaaS solutions. It also makes sense that they have influence when it comes to training.

So what's up with deciding on solutions and initiating projects? Why aren't more organizations relying on DBAs for two of the tasks in which they have the most experience and are most knowledgeable? As we saw above, DBAs chart the right path to the cloud through

the research they perform and the answers they collect; after that, what's the deal? Do the grownups with the dark glasses come along and say, "Good work, Slick, we'll take it from here"?

Making decisions about the cloud and initiating cloud projects are two areas where we DBAs have a lot of influence to exert and room to grow.

Finally, that ties in to the big WIIFM⁸ of cloud and DBaaS for DBAs: to spend less time on the mechanics of database administration like deploying new databases, performing backups, monitoring performance, testing disaster recovery and installing upgrades. The more you move to the cloud, the more those tasks should go from your to-do list to your cloud vendor's to-do list.

Naturally, you'll keep an eye on your vendors, but at the same time you'll have gotten a lot of the boring duties off your plate so you can focus on innovating and adding more value to the business.

⁷ King, "DBAs Face New Challenges: Trend in Database Administration," Figure 27.

⁸ What's in it for me?

Conclusion

As long as databases continue to evolve, so too will our role as a DBA.

There's nothing wrong with plugging away at the same DBA duties you've known all these years. But eventually trends like DevOps, multi-platform databases and the cloud will cause those duties to change. The sooner you can identify and pursue the opportunities each trend brings, the sooner you can move past the zombie stage of database administration and on to the high-value tasks that turn DBAs into true partners in the business.

Find out more about DevOps from the tech brief Pini wrote: [The Importance of Continuous Monitoring in DevOps Pipeline](#).

Find out more about multi-platform databases from the tech brief Pini wrote: [Managing Cross-Platform Database Environments](#).

Find out about the challenges your DBA peers are facing every day in this report from DBTA: [DBAs Face New Challenges: Trend in Database Administration](#).



ABOUT QUEST

At Quest, our purpose is to solve complex problems with simple solutions. We accomplish this with a philosophy focused on great products, great service and an overall goal of being simple to do business with. Our vision is to deliver technology that eliminates the need to choose between efficiency and effectiveness, which means you and your organization can spend less time on IT administration and more time on business innovation.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.

Attn: LEGAL Dept

4 Polaris Way

Aliso Viejo, CA 92656

Refer to our website (www.quest.com) for regional and international office information.

© 2018 Quest Software Inc. ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at www.quest.com/legal

Trademarks

Quest and the Quest logo are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit www.quest.com/legal/trademark-information.aspx. All other trademarks are property of their respective owners.