

Automating the Digital Transformation Journey

Having to shut down an application for maintenance should no longer happen. Moving to an optimised, highly available IT environment should be the aim for every organisation.

Key Points:

'Digital Transformation' may well be one of IT's most abused terms, yet those that have planned their journey well are realising solid business benefits from it. A large part of this is gained by making the organisation more agile through the use of DevOps and continuous delivery.

However, as with all journeys, the route to the destination is dictated by where you start from. For most organisations, the journey begins with a mix of different on-premise enterprise and departmental applications, often alongside a proportion of public cloud-based functionality and services.

Therefore, a more flexible way of creating, monitoring, managing and changing business process flows is required that provides greater visibility of not just all an enterprise's data, but also through automating the data insights and process flows.

Report Authors

Clive Longbottom

Tel: +44 118 948 3360

Email: Clive.Longbottom@Quocirca.com

Marcus Austin

Tel: +44 7973 511 045

Email: Marcus.Austin@Quocirca.com



Commissioned By:

Automic™

Digital Transformation

Digital transformation is one of those terms that may seem to be nothing more than a sound bite. However, it is true that many organisations are facing a greater change in how they use IT as a platform than ever before. Sure, there have been previous big changes – the move from the mainframe to distributed computing, the use of client/server, the advent of widespread Internet use and the web and so on - but the ‘perfect storm’ of cloud computing, containerisation and continuous delivery means that organisations must now review how they use IT.



Figure 1: The Path to Digital Transformation

Digital transformation is predicated not only on forklift upgrades, where everything is replaced as rapidly as possible, but also on creating the means of moving at a measured rate from where an organisation is now to where it needs to be. And, as is ever the case with technology, that ‘place it needs to

be’ is ever changing – so the tools put in place to support the journey must be capable of embracing change as it happens.

One of the biggest issues when commentators discuss digital transformation is that it is assumed that the future is all about composite hybrid cloud-based applications built for the cloud using microservices. It sounds good – and it makes creating a compelling message easier for the commentator. However, for an average organisation, this is not the case – they will have invested large amounts of capital and operating expense in implementing existing monolithic enterprise applications, such as Siebel, SAP and so on.

Forklift upgrades are not only something that IT doesn’t want to undertake, but also the business has become wary of the approach due to the amount of business operational disruption that is involved in moving over from one platform to another.

The key to a controlled digital transformation is to focus on two areas – the data and the technical functionality – that facilitate business processes. By taking such an approach, a digital transformation can be managed effectively.

Things ain’t what they used to be

Ever heard of mainframes? Or client/server computing? Don’t forget that it’s only an IT generation since web-enabled applications were all the rage. However, every change has led to the cries of ‘everything that has gone on before is dead’ – and yet your organisation is probably still running several applications that are dependent on these platforms.

But the future is cloud computing. Almost everyone accepts this – so it must be a case of getting rid of all of these legacy systems and going cloud-native, surely?

No, this is the wrong approach.

Although there is a move toward cloud computing, the real key is in organisations doing so at a speed that best suits them: one that fits in with their corporate risk profile and budget. The future is hybrid for a long time – not just hybrid in the sense of private/public cloud, but also in the sense of older-style application and platform architectures alongside cloud. By putting in place systems that enable orchestration of business processes and the data flows that they are dependent on while enabling the movement of functionality from older systems to new platforms, the digital transformation journey can be streamlined.

But what does this mean for the IT team and the business?

There is a ‘perfect storm’ of issues that IT and the business are struggling to deal with – not only with cloud, but also big data, the internet of things/everything (IoT/E). This is leading to a need for organisations to review where they are and where they are going – but also requires thought on how their applications and functions are written and provisioned.

The perfect storm is therefore made worse by the addition of other areas, such as DevOps, agile computing and continuous delivery (CD). Again, the received wisdom seems to be to throw out everything that has been used before around code control and release management and move instead to a new mix of software delivery tools such as Jenkins and Chef. Again, this suggests organisations having to completely move away from cascade project approaches – which may not be that easy unless managed correctly.

Cloud has also changed the environment. Again, moving to cloud is not a binary option for organisations. Many are in the process of moving some workloads from a purely physical world to private cloud, and others to public cloud through use of infrastructure, platform or software as a service (I/P/SaaS) model. Maintaining visibility across such a hybrid platform is important – but should not require a mix of tools that are each dedicated to looking at a specific type of platform.



Figure 2: The Perfect Storm

The world does not operate in such binary stages: instead, the need to plan and move from the position an organisation is at now toward that required state has to be managed.

Meanwhile, the actual users are no longer scared of using technology, nor are they aware of how many different shades of grey they are dealing with in such technology usage. In their consumer lives, they have embraced the advent of the ‘app economy’. Here, incremental change is not only accepted, it is demanded. Alongside this acceptance of change is a demand for ‘always available’: it only needs a glance to social media to see how users respond when there is even a small interruption to their enjoyment of an app.

The time has come for organisations to better balance the demands of their own users with those of the organisation. Not by throwing everything out and

starting again; not by changing the way everything is done so as to meet the individual's need at the cost of the organisation's. Now, what is required is a means of ensuring that quality of experience and outcome are maintained while introducing more agile and flexible concepts that enable both continuous delivery of function while enabling faster, more flexible and effective business process flows to be facilitated.

Bridging the islands

One of the continuing issues around IT has been 'islands of data'. The enterprise resource planning (ERP) system stores its data here; the customer relationship management (CRM) system has a different database over there; whilst the IT help desk has its data in a different place, as does the human resources (HR) system and so on.



Figure 3: Bridging the Islands

This then leads on to business constraints of 'islands of automation' – it is difficult to automate processes across such disparate data stores. Attempting to pull everything together has led to the growth of markets such as enterprise application integration (EAI) via enterprise service busses (ESBs); extract, transform and load (ETL) tools for creating massive data warehouses and so on

– yet none of these seem to have solved the problem. As it is so difficult, many do not bother, instead either using suboptimal processes that reside within a single 'enterprise' application, or trying to automate end-to-end processes across multiple applications that do not interact well with each other, leading to issues around whether the output from such processes has any real business value.

Instead, why not make the process views intelligent, providing a single means of working across all the data assets, so enabling analytics to be run directly across such a unified view?

Through such an approach, business processes can be simplified, with the complexities of dealing with disparate data stores hidden from the end user. Imagine being able to carry out full end-to-end HR processes – everything from advertising a post through to on-boarding, performance reviews and rewards through a single interface. Imagine being able to have access to all the required information so that financial reports can be created simply and effectively within the timescales that are demanded by governance, risk and compliance (GRC) needs.

Being able to have analytics visibility across the end-to-end process flows and total available data assets of an organisation leads to faster and better decisions being made. It also leads to better performance: processes can be more easily adapted to respond to changes in market conditions; initiatives by competitors can be more easily tracked and responded to.

Sure – this needs more than just a simple layer of technology. It requires a change of thought process to one where change is the only constant; one where the business and IT work together with a single aim.

This needs a more agile approach to IT – a move away from cascade-style projects to one which provides direct support through to the business on a more continuous basis.

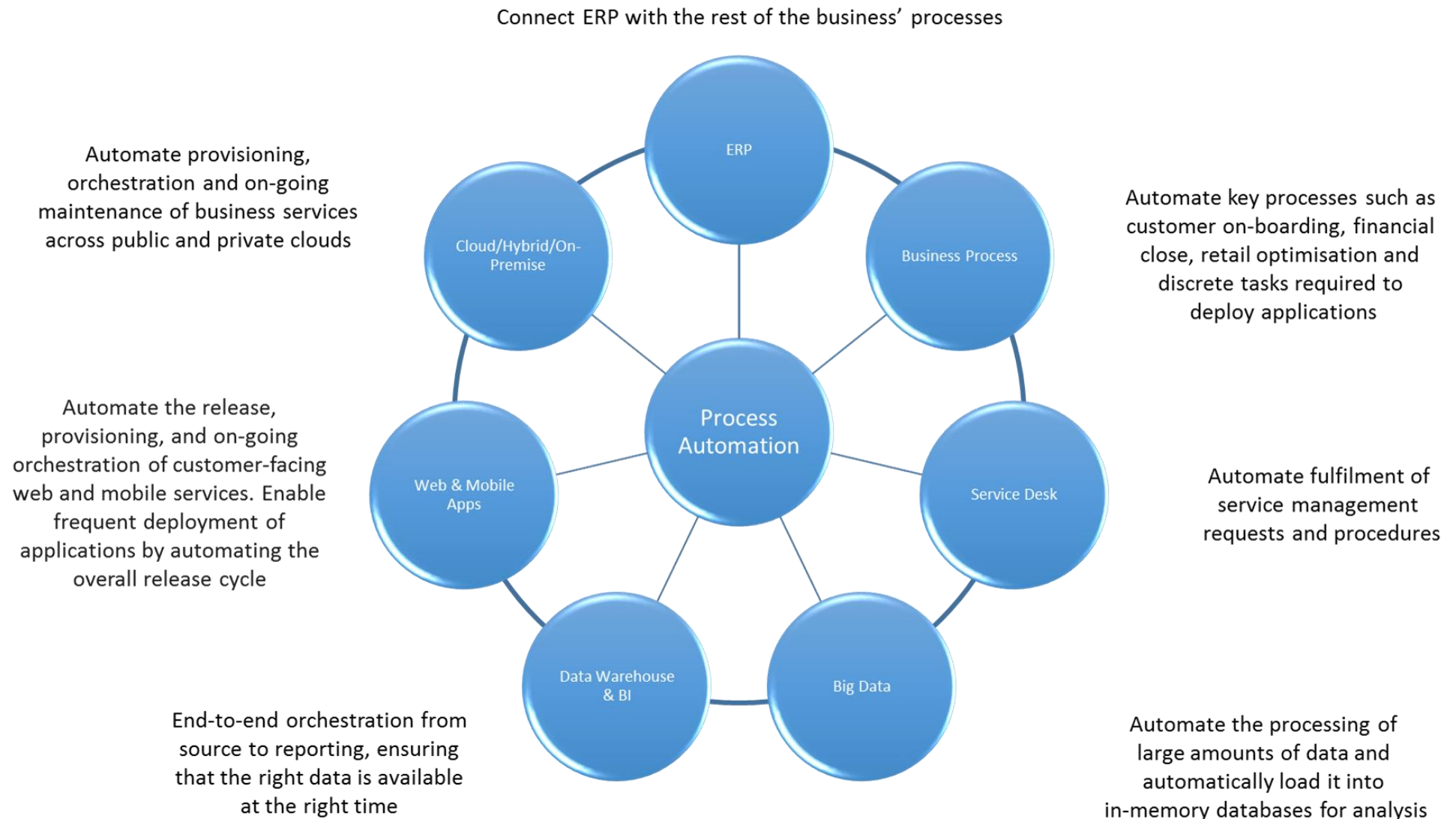


Figure 4: Driving intelligent insight across islands of automation/data

DevOps, Agile and Continuous Delivery

Old cascade-style projects were predicated on having a version 1 release of software followed by timed releases of other sub- or full-versions on a known, semi-manageable basis. For example, if V1.0 of the software was released on March 1st, V1.1 (fixing issues and adding small amounts of extra functionality) would be planned right from the outset as being, say, due for release on September 1st. V2.0 of the software would be timed for a 12- or 18-month schedule.

“Continuous Delivery (CD) has moved from being a niche approach in commercial environments to being one demanded by users, now that they have seen the power of CD in the consumer environment, where they now expect to see small changes provided on a constant basis to things like Facebook, Instagram and SnapChat.”

24x7, highly available systems were assumed to be there. However, this rarely was the case. Although money could be thrown at problems to avoid unplanned downtime, planned downtime still plagued the markets. Large functional updates and patches were carefully timed to fit in with the organisation’s business operations. However, there was all too often no ‘suitable’ time available.

Those using major commercial off-the-shelf (COTS) enterprise applications such as SAP, Oracle eBusiness Suite or Siebel are all too aware of the problems

that this can introduce. That ‘off-the-shelf’ description is meant to point to such systems being unpacked, installed and run without any need for major changes. However, in-house integrations, modifications and additions meant that there is no such thing as a basic COTS deployment. Alongside the need to deal with the software vendor’s updates and patches, there is also the problem of dealing with all those in-house needs – and whether a change in one area can then lead to a problem in another.

This is made worse by the adoption of DevOps, Agile projects and CD against old-style applications. Even with new composite applications, this approach is struggling: taking down even one small part of a composite application can lead to the whole application being unavailable.

DevOps has been adopted to help streamline the process of functional delivery through development, test and production by automating as many steps in the process as possible.

Agile projects are based on being able to provide software against the needs of the organisation as a set of rapidly developed ideas, concepts and proof of concepts, followed by fast development, testing and provisioning, generally through a DevOps approach.

CD has moved from being a niche approach in commercial environments to being one demanded by users, now that they have seen the power of CD in the consumer environment, where they now expect to see small changes provided on a constant basis to things like Facebook, Instagram and SnapChat.

CD is aimed at getting around the issues of storing up functionality until a release/update window is available. Instead, it is about responding to needs in the fastest time possible. The organisation needs some new functionality? Great – use Agile and DevOps, followed by CD to get the functionality out there within hours or days.

However, attempting to introduce 'true' CD in a world where maintenance windows are still required is both contradictory and can introduce inefficiencies that cause excess process interruptions. Continuous interruptions to the users' capabilities to access applications will not only alienate the users, but will also have a major impact on an organisation's performance.

"It is all very well having the upstream side sorted out – ensuring that the IT processes are all optimised and working. Without the same optimisation and analysis of the downstream side, it will all have been of little use."

Everything needs orchestrating effectively and efficiently: automation can remove the complex dependencies that lead to the need for downtime.

What does this all mean?

Let's assume for the moment that everything discussed above is possible. An effective agile DevOps environment is put in place that results in solid continuous delivery. The business is very happy, as although they are unaware of all the complexity of tooling that has had to be put in place, they now have a reliable basis for creating and using new processes and for getting new services and products out into the market.

It is all very well having the upstream side sorted out – ensuring that the IT processes are all optimised and working. Without the same optimisation and analysis of the downstream business side, it will all have been of little use.

A platform is required that can provide a single overall view across all the different systems, all the process flows and all the data, so enabling analytics to be carried out 'in the round'. This is shown at a high level in figure 4 – all data stores can be viewed from a single point; all the data can be analysed in place; the results can drive optimised business processes that result in solid additional business value.

Customer, supplier and other stakeholder activity and performance can be more easily monitored, measured and acted upon. New means of cross- and up-selling can be tried out with customers. Using the power of full advanced analytics across all available data and basing new process automation flows on the results opens up a raft of new possibilities – ones that any organisation should welcome.

The improvements in efficiency and effectiveness could also result in extra time being freed up for the workforce. This opens up the opportunities for the business to try out new things: for example, the introduction of new products or services, new ways of doing things, investigating new markets. With an effective platform monitoring and reporting on all aspects of data and process flows while enabling a truly effective DevOps/CD model, it is easier (and more cost effective) for organisations to try out such new things – feedback will be more direct and ideas that are obviously not working can be shut down rapidly.

Most users will enjoy working in an environment where such activity is encouraged – as long as there is also some form of reward at the end for those who do come up with an idea that makes a significant positive difference.

For the business itself, freeing up such time also enables it to plan activities far more effectively as it finds the need for less overtime, less requirement for cover staff during periods of, for example, school holidays and so on.

The complex tooling environment

The DevOps/Agile/CD approach is not something put together in a planned manner by incumbent software vendors and then introduced to their customers. What happened was that developers and operations staff were looking for new ways of dealing with the demands being placed upon them for faster delivery of functionality, and so a new breed of open source software tools happened to come together and be used by the different parts of the development/test/operations environment to enable more effective process flows within their own teams. This then led to a need for each team to provide a means of automating the process flows between them - and somewhat serendipitously, DevOps was born.

So, tools like Ruby, Python and so on were being used by developers, with code being managed by other open source version control systems, such as Git or Subversion. The movement of software along the development process was managed with continuous integration tools such as Jenkins, providing certain low-level capabilities, such as bringing together different distributed copies of developer code into a single mainline working copy on a regular basis.

To provision software into the operations environment had previously been carried out through the use of hand-written scripts. Tools such as Chef and Puppet emerged to automate such scripting, and have grown in functionality to include capabilities for understanding the dependencies between different parts (such as database, storage pointers, etc.) of an application as it gets provisioned.

The problem is that as the DevOps approach grew, it was through individual technical staff finding the best tools to suit them. By the time DevOps had become an accepted term and the big vendors threw their hats in the ring, it was, essentially, too late.

Trying to force technical staff into accepting something that is a replacement for something that they have chosen themselves is like herding cats, only less successful.



Figure 5: Avoiding DevOps Chaos

An organisation may try and insist that its technical staff all move onto Large Co's Premium DevOps End2End Enterprise Edition (v8.3a), but if the technical person has found their choice of tool to do what they as an individual want, an audit 12 months down the line will show that Large Co's product is still shelfware, while the IT department is running 28 different open source systems. Alongside this could well be the human factor of the development team not only choosing to use a different set of tools, but pretty much blankly refusing to work with the operations team on a point of principal. And herein lies chaos. If one operations person is using Chef and another Puppet, neither is seeing what the other is doing. If the upstream developers are using Jenkins,

unless there is a solid way to bridge what Jenkins does to that Chef and/or Puppet environment, the chances of putting in place a workable CD process are almost certainly zero. Tools that were meant to introduce efficiencies suddenly cause problems where different tools require a lot more manual intervention to decode scripts and move the processes along and so on.

And when push comes to shove, all these tools still run up against the same brick wall: application and code updates require a maintenance window.

Order needs to be brought to chaos: automation needs to be made easy across the chosen systems. Rather than forcing everyone into using the same systems that just happen to allow better automation, it is better to choose an automation tool that enables the continued use of the tools and the skills that have already been built up across the organisation.

Aiming for 'absolute zero' downtime

There are two main types of downtime to consider: the downtime required to update the software that users within an organisation rely on, and the downtime required when using tools that are there to carry out the tasks of updating the environment.

The first is fairly self-explanatory: users do not like downtime; it gets in the way of their capabilities to do their job. Therefore, avoiding downtime while maintaining applications and functions should be a primary aim.

However, the second one is harder to deal with. Let's assume that an organisation has managed to implement a set of tools that manage the updating process to the applications and functions such that downtime is avoided. This is great – everyone is happy. Except for when these tools need updating themselves, and downtime is required to carry out the update. This need for downtime means that no updates can be carried out to the applications and functions the users are dependent upon during that time.

With the IT group, downtime can be minimised, but not necessarily eliminated, through the use of common tools and agreed processes. However, as previously discussed, it is not really possible to insist that all IT staff move over onto a common set of tools. Even where COTS software is involved and the vendors have tried to ensure that the maintenance window required is minimised, there will still be a need for downtime. This may just be because of the vendor's upgrade; it may also have to be to deal with all the dependencies that are in place due to custom code and integrations. Where custom code is involved, there could well be a long (and often unsuccessful) import and transformation process before the code can be output successfully into the operational environment.

Indeed, many software vendors have moved to the adoption of containers as a means of getting packaged code out into the customer's operational environment quickly and efficiently, with maintenance windows being minimised through such technology. However, this still does not address any issues around custom code and integrations; a 1 second maintenance window is still a maintenance window. The system is still not available, and unless other means of capturing and managing state, as well as capturing and managing transactions that are in mid-flow are in place, data, information and value will still be lost.

So – the search for zero downtime goes on.

The key is in being able to provide an underlying automation layer that is fully data and environment aware. In this way, the various aspects of the whole DevOps environment can be pulled together, and code can be provisioned in a manner that does not require any downtime. Such orchestration is complex, but when carried out successfully can make a DevOps/Agile/CD approach work well for an organisation.

Each constituent in the process can continue using their favourite tools, the underlying automation system pulls the various pieces together to ensure that

everything is maintained and monitored through a 'single pane of glass'. Roll out and roll back can be maintained through such automation: system contextualities and dependencies can be maintained throughout.

A suitable tool will have blueprints or templates that can be created from existing environments, so creating a model-based, declarative system where a new application can be provisioned based on the existing application instance.

Such tools must reflect what they provide: if they need downtime for updates to their systems, then the end result is the same. The business has to wait until the tools have been updated – opportunities can be missed and revenues impacted.

The need for such 'absolute zero' downtime should not be underestimated.



Conclusions

The rapid pace of change in both the business and technology spaces cannot be overstated. The need for organisations to be able to respond to market forces in near-real time means that many traditional approaches to IT and business processes are no longer viable. Cascade projects with prescriptive tools; large application, data and processes silos forcing islands of automation; ‘in the rear view mirror’ data analysis are all now threats to how well an organisation can perform – indeed, without a move to near-real time operations they are threats to the very survival of many organisations.

Now, the drive is toward far more flexible and effective business and IT processes that work together. The aim is toward DevOps providing continuous delivery directly to the business, with the business being able to fully use the data resulting from the new functionality for analytics and business process needs.

However, throwing the baby out with the bath water is not an option; nor is major forklift upgrading. Organisations must look to the creation of an effective automation platform that embraces everything that is already in place – yet adds appreciable value to it all.

For example, the tools that development and operations have grown accustomed to using – Jenkins, Chefs and Puppets and so on – need to be orchestrated, so that each group can focus on getting its work done without the need for major re-training. However, any system that is put in place needs to make these tools work better together – enabling streamlined, secure and manageable processing of the whole DevOps environment.

Likewise, within the business environment, the monitoring of workflows that support existing enterprise applications needs to be made more visible across the whole organisation. Only through full visibility can advanced analytics take place that have true value. Analysing a sub-set of these workflows can lead to inexact and often misleading results: providing the capability to analyse everything will result in better decisions that will have more of a positive impact on the business.

The platform must also enable full workflow automation – both within itself and across the broader IT environment. By providing broad data and process flow visibility, such a system can provide timely information and knowledge to those in the business that need it – but only if it can ensure that the information gets to the person at the right time.

And finally, the platform must eliminate downtime. Continuous delivery built around a need for each incremental implementation to require a system downtime is not feasible. Ensuring that systems can be updated, patched and replaced without downtime has to be the top of the priority list for any organisation. This also needs the solution chosen to follow its own mantra – if it requires downtime for its own updates, look elsewhere.

About Automic

Automic, the leader in business automation, helps enterprises drive competitive advantage by automating their IT and business systems - from on-premise to the Cloud, Big Data and the Internet of Things. With offices worldwide, Automic powers over 2,700 customers including Bosch, Netflix, eBay, AMC Theatres, Carphone Warehouse, ExxonMobil, Vodafone, Société Générale, NHS SBS, General Electric and Swisscom. More information can be found at www.automic.com.

About Quocirca

Quocirca is a research and analysis company with a primary focus on the European market. Quocirca produces free to market content aimed at IT decision makers and those that influence them in business of all sizes and public sector organisations. Much of the content Quocirca produces is based on its own primary research. For this primary research, Quocirca has native language telephone interviewing capabilities across Europe and is also able to cover North America and the Asia Pacific region. Research is conducted one-to-one with individuals in target job roles to ensure the right questions are being asked of the right people. Comparative results are reported by geography, industry, size of business, job role and other parameters as required. The research is sponsored by a broad spectrum of IT vendors, service providers and channel organisations. However, all Quocirca content is written from an independent standpoint and addresses the issues with regard to the use of IT within the context of an organisation, rather than specific products. Therefore, Quocirca's advice is free from vendor bias and is based purely on the insight gained through research, combined with the broad knowledge and analytical capabilities of Quocirca's analysts who focus on the "big picture". Quocirca is widely regarded as one of the most influential analyst companies in Europe. Through its close relationships with the media, Quocirca articles and reports reach millions of influencers and decision makers. Quocirca reports are made available through many [media](#) partners.

To see more about Quocirca's analysts, click [here](#)

To see a list of some of Quocirca's customers, click [here](#)

To contact Quocirca, please click [here](#).