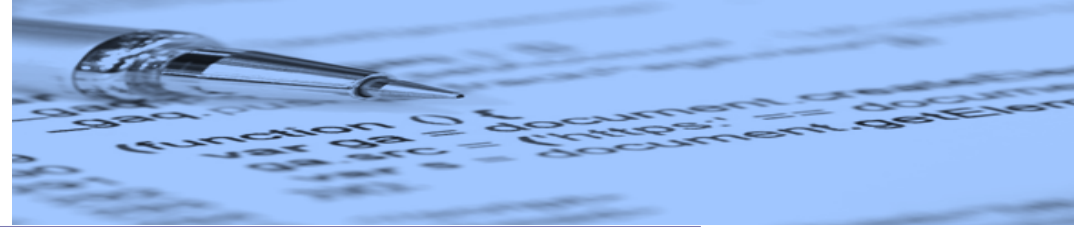


Business-driven application release automation



Enabling a more responsive business through the use of automation in the DevOps process

August 2015

The age of cascade development and long drawn out software projects is in decline. Businesses have a need for continuous delivery of new functions, and this requires a far more streamlined approach from IT.

Systems of engagement and systems of record need to be abstracted; approaches need to change. Automating functional release in a secure and manageable manner enables IT to better facilitate rapidly changing business process needs. This requires robust tools to work alongside development, test and operational resources.

Report authors

Clive Longbottom

Tel: +44 118 948 3360

Email: Clive.Longbottom@Quocirca.com

Bernt Ostergaard

Tel: +45 45 50 51 00

Email: Bernt.Ostergaard@Quocirca.com

Sponsor

AutomicTM

Business-driven application release automation

Providing fast, dynamic support to the business through the provision of continuous delivery of incremental functionality should be the top priority for any head of IT. However, just speeding up the flow of code from development-to-operations is not enough: it has to be fully managed and audited, ensuring that new apps, enhancements and updates are ready to do business. A properly implemented DevOps strategy, led top-down by the business, can enable organisations to be more competitive by delivering richer services to their customers faster.

Traditional cascade approaches to development are no longer valid	Creating new applications over periods of time, measured in months, fails to support rapidly changing business needs. At best, such an approach is trying to deal with problems the business had, but may no longer have. At worst, it can prevent the business from reacting rapidly to competitive pressures and can place the business itself in jeopardy. A new approach is needed – one that enables IT to react promptly yet effectively to what the business requires.
Existing applications need to be updated to give a better user experience	Enterprise applications, while still providing the backbone to most organisations, are beginning to show their age. The majority of end users are accustomed to rapid, but incremental changes in how they deal with their own technology. Yet they see little change over long periods of time in how they interact with an enterprise application, followed by a huge change requiring re-training and ongoing issues. This presents a massive problem to IT: does it go for faster forklift upgrades, or look to a different approach?
DevOps is not a silver bullet	DevOps needs some checks and balances if it is to be effective. Feedback loops can be used to ensure that developers see what the IT and business priorities are, with more of a 'BizDevOps' focus. The ability to roll back to a previous known state quickly where necessary is a key part of this. While process automation is key, this cannot be done blindly: intelligent automation is required to ensure that DevOps is successful.
Systems of record and engagement can leverage the benefits of DevOps	By breaking down how an organisation utilises its existing applications into systems of record (how data is stored and manipulated) and systems of engagement (how the user interacts with the data), a far more responsive overall environment can be built up. While the systems of record continue to be data stores and execute the business logic, there is a continued need to extend the use of enterprise applications, as well as roll-out updates. Meantime building new apps and enhancing the existing systems of engagement faster will have a direct positive business impact. Both systems can benefit from the extra speed and agility DevOps can bring.
A closed-loop approach to DevOps provides greater business value	Open-loop approaches to application development, delivery and management have led to isolation of many systems with no feedback of problems in a manner conducive to issues being dealt with effectively. A closed-loop approach ensures that information is captured early and is fed back to the right people at the right time to enable effective resolution of any issues. A closed-loop approach also ensures that the forward movement of code can be automated: the underlying platform needs for the code can be guaranteed to be available.
Packaging of code in a suitable manner is key	There is much more than code that goes in to the constituting of a new app. Making sure all the pieces are collected, correct version levels being used and other artefacts are packaged before being pushed out into the operational environment, is a time consuming, error-prone process. A key part of any DevOps strategy has to be the automation of such packaging, ensuring that all code can be pushed out into operations in the knowledge that it will work as developed.

Conclusions

DevOps promises much, and an effectively implemented DevOps strategy can have a powerful positive impact on business performance. Simply speeding up the movement of code from development through testing to operations will result in more errors and downtime. Putting in place a set of automated processes based on solid policies ensures that code flows as it should do. Abstracting systems of engagement away from existing systems of record provides a means of optimising continuous delivery to the business and its users of IT. For many organisations, an approach of 'refurbishment' and continuous improvement of the system of engagement will provide the greatest return on investment, leading to a measured migration to a new world of composite applications.

The problem with applications

Historically, organisations have bought or written monolithic applications that they hoped would deal with a large area of their business problems. As such, whole application markets have grown up around areas such as enterprise resource planning (ERP), customer relationship management (CRM) and supply chain management (SCM). However, as time has progressed, it has become apparent to many organisations that trying to adapt these monolithic applications to changing business circumstances is costly, complex and – eventually – self-defeating.

This all came to high visibility as the global economic collapse happened. Up until then, many organisations had sufficient cash flows that poor systems and processes could remain hidden. As the recession bit, cash flow became far more important, and the problems with highly prescriptive applications became more telling. A business needed to respond far more rapidly to market conditions, yet it would take IT many months to change or layer code over the monolithic application to meet the business' needs. This also led to the issue that by the time the 'solution' was in place, it was trying to solve a problem that the business had many months ago – and that it had either found an alternative way to work around, or had already lost a large amount of time and money over. IT was rapidly becoming part of the problem, rather than part of the solution.

In the majority of cases, the business has a simple issue: it needs to be able to sell more of what it already sells at the same or greater margin and/or it needs to bring a new product or service to market with sufficient margin. While IT should have been helping this to happen, in many cases it was preventing it.

There had, however, been several different movements within the technology industry that were converging to give hope to organisations. Virtualisation and cloud computing were offering faster ways to provision applications; cloud computing was offering the capability for functional components to be brought together to create a composite application that could be more flexible than a monolithic one. Agile development approaches were replacing cascade in many places – and now we have DevOps (a philosophy aimed at bringing the development and operational aspects of IT more closely together) and automation layered over the virtual/cloud platform to offer a faster and more streamlined means of dealing with the IT code creation, testing and implementation lifecycle.

And yet organisations still struggle. DevOps can only work where the right checks and balances are in place

What the business needs



Figure 1: What the business needs



to ensure that developers don't just push anything out that they believe is ready for operational use. It also needs to be driven not by IT, but by the business. Tools are required to oversee the overall code process and application lifecycle.

Where automation can be used, it should be, so that human error can be avoided and full monitoring and issue resolution can be carried out quickly and effectively where needed. While part of a long tail where most organisations will still be wanting to ensure that they extract the maximum value from their existing enterprise applications, monolithic applications should be deprecated as they reach the end of their useful life. New, flexible composite applications based on the continuous delivery of incremental change to the business is what is now required. DevOps can still help with that long tail – a strategy of continuous delivery of new functionality abstracted from the core system can still be implemented and used throughout that period of time.

Quocirca sees many organisations that regard continuous delivery as an issue: they doubt that their users will be able to cope with lots of change; that the help desk will struggle to support so much change and

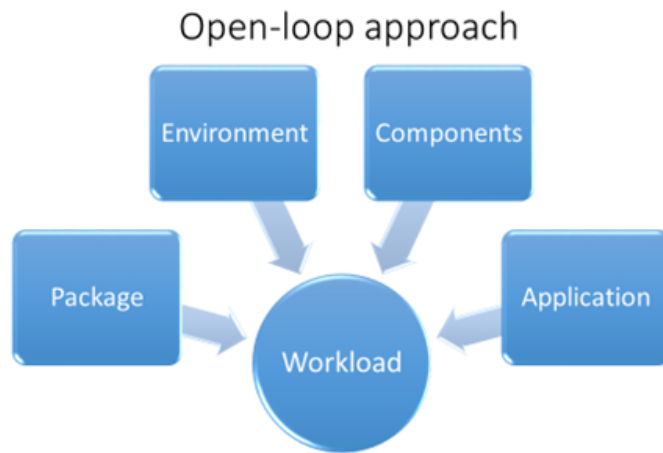


Figure 2: Open-loop approach

The need for Application Release Automation

“We are within the eye of a storm: the economic situation is looking better, but purse strings will not suddenly be opened up to pay for massive changes in an organisation’s approach to IT. Forklift upgrades, where everything to do with an environment is replaced in one go, are still not on the cards for the majority of organisations. IT investments have to be carefully considered. Only changes that have a tangible positive impact on how well the business can operate will gain the requisite funding. Application release automation should be one of these.”

that their operational systems will be unable to cope with the changes in workload. However, users are now comfortable with incremental change – over the air updates to apps they use on a daily basis on tablets and smartphones no longer phase them. The help desk is in the same position – as long as they know what has changed and what this means to how a user should be using the system, they do not need major re-training for each change that takes place. For the operational environment, intelligent tools can ensure that new code is only placed on the right platform with the right technology resources allocated to it so as to ensure that the overall IT platform maintains an optimum operations capability.

We are within the eye of a storm: the economic situation is looking better, but purse strings will not suddenly be opened up to pay for massive changes in an organisation’s approach to IT. Forklift upgrades, where everything to do with an environment is replaced in one go, are still not on the cards for the majority of organisations. IT investments have to be carefully considered. Only changes that have a tangible positive impact on how well the business can operate will gain the requisite funding. Application release automation should be one of these.

The rise – and fall – of DevOps

The key to enabling continuous improvement to the business is in providing a set of optimised IT processes that make the entire development/test/operations system operate more flexibly. However, this also needs to be done in a manner that is secure, audited and can be easily rolled back if problems are seen downstream. Indeed, the whole approach should not be seen as a one-way flow: operations must be able to provide feedback into development, along with supporting data from the help desk so

that developers can better prioritise what issues they should be working on. It also has to be driven top down from the business – a bottom up approach from IT will perpetuate the problems perceived by the business in how IT is not particularly responsive to its needs. In many ways, the rise of the term ‘DevOps’ is wrong – it really needs to be ‘BizDevOps’, reminding everyone involved that the business really does come first.

Historically, there has been an approach to systems development that is ‘open-loop’ based. Development groups realise that there is a need for different roles within the DevOps process, but put these in place as discrete functions. Therefore, the development team will create the application, with any dependent components being identified and collated by the test team. The environment requirements will be decided by the systems analyst team, and the whole package will then be put together and implemented by the operations team. Issues can occur anywhere – and finger pointing will happen as teams and individuals try to transfer the blame.

Even now, many DevOps products try to shoe-horn in this old style approach: a set of point solutions attempting to pull disparate functions together. This is a sub-optimal approach that will only lead to further problems down the line.

So, DevOps needs to be a controlled process that enables functionality to be developed, tested and implemented in a secure and managed manner within an organisation, so as to ensure that the business aims of improving profit through IT’s adequate support of the necessary business processes occurs.

However, the implementation of DevOps in many organisations has not been all that successful. All too often, it has been implemented mainly as a means of

The system of engagement is how an individual engages with the process. Generally speaking, within an IT system, this can be seen as being the user interface – but for areas such as retail, where multi-channel comes in to play, the system of engagement can include such things as telephone, face-to-face, social media and so on.

Most organisations will have existing systems of records that they want to extract ongoing value from – part of the long tail of the move to composite applications discussed earlier. Return on investments in areas such as CRM and ERP need to be maximised, and few organisations have the appetite for complete replacement of systems unless there is a very solid reason to do so, as the impact on the business itself can be major. Review points for such core enterprise systems are few and far between – any DevOps approach needs to deal with existing systems, composite apps and all the stages in between.

The first step for many organisations will be the realisation that the systems of engagement in use with existing applications are often becoming a constraint on the business. Users more familiar with the ease of use of mobile-based Apps in their home life look at client/server front ends from only a few years back and shudder. Systems that seemed OK now look aged and are slow to use. In many cases, the user will find other ways to carry out tasks, avoiding the use of the system if they can – and so creating data that is not in the correct system of record.

The obvious outcome here is that the business starts to operate against incomplete data, leading to errors and poor decision making. To avoid such issues, the system of engagement needs to be intuitive and attractive: it has to offer the user more value in what they get back than the effort they perceive they put in.

By looking at the systems of engagement as an abstraction from the system of record, a far more dynamic approach to how business applications are created and operated can be carried out. In most cases, the pace of change at the system of record level will be far slower than the need for change at the system of engagement. A well-architected system of record will be capable of storing the data that the system of engagement creates, but the system of engagement may not meet the needs of the user or the business as



their expectations change. Indeed, a major consideration of DevOps should be how existing systems of record can be utilised with a different, more effective system of engagement. For many organisations, such an approach of ‘refurbishment’ as opposed to ‘new build’ will provide them with the least risk and cost, while giving the biggest opportunity for additional value add.

The need is for continuous delivery to be available. The DevOps system must allow for both the systems of

record and engagement to be fully abstracted, such that functional changes to the user interface can be made rapidly to meet the requirements of the user and business, while more fundamental change to the system can be carried out to the back end without the need to change the front end.

Such an approach of abstracting the two systems also enables changes to be made that will not have impact on the other system. New ideas can be tried out – and then dropped if they do not add the value expected. Parallel developments can be carried out, with specific groups using the new functionality against live data, only bringing in the rest of the user base once the changes have been proven.

Existing organisations also need to look at what is happening with new start-ups. Green-field organisations are using DevOps concepts to avoid the need for much IT. By using cloud computing functionality, companies like Uber and Airbnb minimise the need for a self-owned IT platform, instead pulling together cloud functions to facilitate their business processes. The impact on existing organisations cannot be overemphasised. These ‘new world’ digital native organisations can move fast and can change their IT far faster than an organisation that is dependent on typical cascade style development projects. It is therefore a necessity that organisations bring in the capabilities of a DevOps approach in order to ensure that they can respond to new threats from these green-field organisations.

The release automation cycle

The release automation cycle has to be understood so that the right checks and balances can be put in place. The overall aim is to ensure that work packages can be promoted through the various stages of the cycle, with continuous feedback loops to ensure that remedial action on any issues can be carried out at the earliest possibility.

Just what are the aspects of a full DevOps strategy that are required, then? Quocirca believes that the following steps apply:

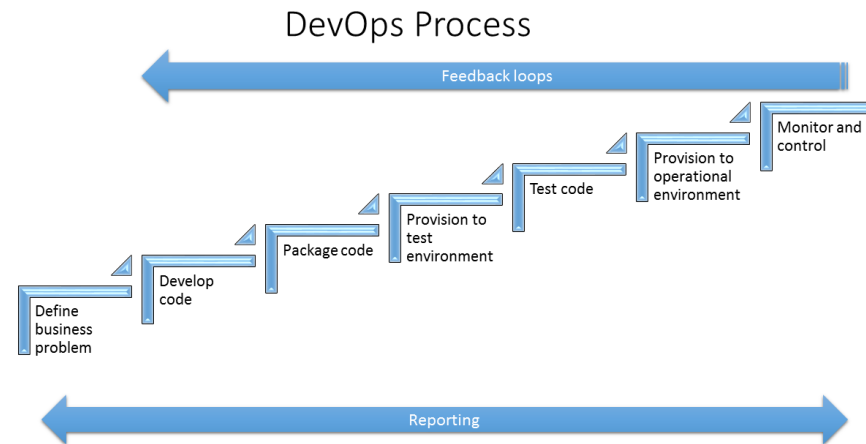


Figure 4: The DevOps Process

- » **Defining the business problem.** Sounds simple enough, but this is a step that is all too commonly left out by the IT department, as it tries to push for an interesting or elegant technology 'solution' to a problem the business doesn't actually have. Desired outcomes and timescales are needed – but not just as a hard document. These needs are dynamic, and should be revisited and reviewed throughout the DevOps process. Only through this approach can DevOps assure that it will deliver the right results.
- » **Code development.** This may be a complete system (a full system of record and engagement), but will increasingly be a continual delivery of incremental changes to the system of engagement, with occasional changes to the system of record. It will also involve integrations across new and existing systems, and the pulling together of diverse cloud-based functionality into a workable system that supports the business' process needs.
- » **Code packaging.** This could be as a set of discrete code items that need provisioning to specific parts of the infrastructure, but will also increasingly be based around the concept of a container. The right tools are required to ensure that the right packaging is used for the right code.
- » **Test provisioning.** Once the packaging has been completed, the code needs to be provisioned in the right environment for testing.
- » **Testing.** Code created by developers that works in their environment may not work once 'out in the wild'. Testing is still required in the operational environment, and tools are required to ensure that all dependencies are covered and allowed for, and that information can be fed back to the developers as required.
- » **Operational provisioning.** Once the test phase has been passed, the code will need provisioning directly into the operational environment. Dependencies have to be checked again, and full roll-back allowed for should there be any problems.
- » **Monitoring and control.** Once the new code is up and running in the operational environment, it needs to be fully monitored, such that any problems can be quickly and effectively fed back to the development team for rectification.
- » **Reporting.** Throughout the DevOps process, full reporting must be possible. This should be flexible enough to be used by all interested parties – developers, testers, sysadmin and business people, so that the status of the process and of the code can be fully understood at all times.

At each stage, the right controls need to be in place to ensure that everything is fully audited and that only those with the right level of authority can move the process on. However, for the full benefits of a DevOps approach, this needs to be as seamless and efficient as possible, allowing such authority to be given by an individual within a group based on full knowledge of all that has gone on to that point of the process. Also, roll back at every stage is required, along with feedback loops to ensure that any issues can be dealt with at the earliest possible point in the process.

The aim has to be for everyone working within the DevOps process to be part of an overall team, so avoiding the dangers of the open-loop approach.

There is also a need for any chosen tool to work with what you already have to avoid forklift upgrades and replacements. For example, development systems, help desk and trouble ticketing systems and systems management tools must all be fully supported and included in the overall process.

Conclusions

DevOps has been put forward as the ‘next big thing’ for organisations to use when dealing with their development needs. However, poorly designed and implemented DevOps strategies can lead to less stable applications, greater unplanned downtime and less user satisfaction.

Organisations need to make sure that they utilise tools that enable DevOps to be put in place that aims to provide what the business needs: maximising the value of existing investments, enabling continuous delivery of new functionality alongside enabling business applications to be migrated from one technology architecture to another as and when this makes sense.

A system that brings together business owners and the disparate groups involved in development, test and operations is required, that also allows for business input at any stage. Such a closed-loop approach can ensure that code and information is moved along the process effectively while allowing for feedback loops that focus development on the issues that matter – those that are impacting the users and the business itself.

Breaking down the approach to development to one dealing with abstracted systems of engagement and record provides a means for an organisation to move at two speeds. At the systems of record level, existing databases and server-side business logic can continue to be utilised, as the required pace of change at this level tends to be relatively low. Meanwhile, the systems of engagement can be altered at will: users can gain incremental functionality on a continuous basis to meet their and the business’ requirements. As the systems of record age and review points are hit, the systems of record can be more easily swapped out if required, with existing systems of engagement maintained to minimise the need for user and help desk agent retraining.

Use Case Scenario

A large third party administrator (TPA)

Business issue: The company has tens of thousands of customers, covering hundreds of thousands of individuals needing access to its IT systems, with an average concurrency of 40,000 people on its systems at any one time. The IT platform is a mixed Windows/Linux environment, mainly on-premise, but also using some software as a service (SaaS) systems as well as a small AWS EC2 private cloud. Due to the company's historical use of large, cascade projects and an open-loop approach, code releases could only be carried out at weekends, due to the need for considerable downtime. The number of updates were constrained, due to the numbers of people involved and the impact the roll-outs had on the business. The overall value of the updates were perceived as relatively low by the business, and the overall quality of the resulting operational code was low, due to too many manual steps. The overall process lacked automation, and was only familiar to a few people. The business was demanding faster technical capabilities to meet its market needs. For IT, the troubles with managing Java and PHP releases were overwhelming its capability to support the business' needs.

Thought process: A move had to be made away from a standard, large development project approach to one of continuous delivery. This required the adoption of a true DevOps approach, with full application release automation. A repeatable process was required that minimised human intervention, while eliminating or significantly shortening any associated down time. The frequency of releases had to be improved, along with the quality of the code provisioned to the operational environment. Controls had to be strengthened, yet the chosen toolset had to be flexible and proven in the marketplace.

Chosen solution: The company chose Automic Release Automation to meet its needs. This provided the required framework for a DevOps, continuous delivery approach, along with all the audit controls and reporting required to ensure that everything was carried out efficiently and effectively.

Business impact: The release cycle for major Java-based applications requiring down time was reduced to 45 minutes, with those not requiring downtime brought to a continuous delivery level. Hot fixes and patching were also moved to being delivered on an 'as required' basis with no negative impact on the business.

For PHP code, the release cycle became fully automated, with all the requisite roles fully trained in how to manage the promotion of each phase of the process as required.

The company also created automated processes for provisioning of AWS EC2 environments, with automated environment refresh and deployment of all required components for a full test system. This has led to the capability for any development or quality assurance individual to rapidly and effectively provision their own test environment as required.

The improvement in reporting means that anyone within the company has visibility on the state of any project. This not only covers the stage within the process any code is, but also the accountability of available environments, ensuring that no individual can overspend on or overuse available resources.

The company is now looking at expanding the approach to cover off-the shelf environments such as its CRM and workflow systems, and to enhance its usage of Automic in automating AWS EC2 and in providing enhanced high availability to the IT systems and applications for the business.

About Automic

Automic, a 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 2,600 customers including Bosch, Netflix, eBay, AMC Theatres, Carphone Warehouse, ExxonMobil, BT Global Services, Société Générale, NHS SBS, General Electric and Swisscom.

More information can be found at www.automic.com.

About Quocirca

Quocirca is a primary research and analysis company specialising in the business impact of information technology and communications (ITC). With world-wide, native language reach, Quocirca provides in-depth insights into the views of buyers and influencers in large, mid-sized and small organisations. Its analyst team is made up of real-world practitioners with first-hand experience of ITC delivery who continuously research and track the industry and its real usage in the markets.

Through researching perceptions, Quocirca uncovers the real hurdles to technology adoption – the personal and political aspects of an organisation's environment and the pressures of the need for demonstrable business value in any implementation. This capability to uncover and report back on the end-user perceptions in the market enables Quocirca to provide advice on the realities of technology adoption, not the promises.

Quocirca research is always pragmatic, business orientated and conducted in the context of the bigger picture. ITC has the ability to transform businesses and the processes that drive them, but often fails to do so. Quocirca's mission is to help organisations improve their success rate in process enablement through better levels of understanding and the adoption of the correct technologies at the correct time.

Quocirca works with global and local providers of ITC products and services to help them deliver on the promise that ITC holds for business. Quocirca's clients include Oracle, IBM, CA, O2, T-Mobile, HP, Xerox, Ricoh and Symantec, along with other large and medium sized vendors, service providers and more specialist firms.

Details of Quocirca's work and the services it offers can be found at www.quocirca.com

Disclaimer:

This report has been written independently by Quocirca Ltd. During the preparation of this report, Quocirca may have used a number of sources for the information and views provided. Although Quocirca has attempted wherever possible to validate the information received from each vendor, Quocirca cannot be held responsible for any errors in information received in this manner.

Although Quocirca has taken what steps it can to ensure that the information provided in this report is true and reflects real market conditions, Quocirca cannot take any responsibility for the ultimate reliability of the details presented. Therefore, Quocirca expressly disclaims all warranties and claims as to the validity of the data presented here, including any and all consequential losses incurred by any organisation or individual taking any action based on such data and advice.

All brand and product names are recognised and acknowledged as trademarks or service marks of their respective holders.