The Practical Blueprint to Continuous Delivery

ca technologies
A Broadcom Company

Executive Summary

CA Technologies, A Broadcom Company, proposes a four-stage blueprint to continuous delivery to assist enterprises at any stage of their DevOps journey. This plan can take a company of any maturity level all the way up to enterprise-scale continuous delivery using a combination of Automic® Continuous Delivery Automation, 40-plus years of business automation experience, and the proven tools and practices the company is already leveraging.

As the only vendor that automates within the toolchain (steps one and two) and across the toolchain (steps three and four), CA Technologies is in a unique position to help enterprises at every stage of their progression. Our tool-agnostic, cross-platform and multi-stack technology means developers can continue using the tools they are most comfortable with, while DevOps teams can collaboratively build automated, production-ready deployment processes that make operations comfortable.

CA Technologies offers a DevOps Maturity Assessment that shows on which of the four steps an organization sits, and compares its current state with that of its competitors, its industry and its competition globally. Using the assessment findings will give companies a blueprint to guide them up to the next level on the road to enterprise agility as quickly and safely as possible.

Step 1: Manual and Scripting

Participation in our DevOps Maturity Assessment will deliver a continuous delivery roadmap that contains a blueprint for DevOps throughout the enterprise. Customers will be able to compare their current state with their competitors, their industry and globally.

Manual steps will be identified and documented, then replaced with automated tasks and workflows. The role and relevance of existing scripts are considered for inclusion in a deployment automation pipeline. Additionally, niche or job-specific tools will be reviewed for their role in producing a coherent deployment automation pipeline.

Step 2: Automate Deployments

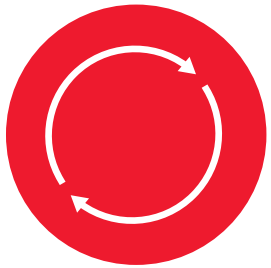
CA Technologies now defines and maps practices, people steps and scripted tasks to model-based automation workflows within Automic Continuous Delivery Automation, creating a reusable automated deployment pipeline. Environment configuration information is modeled separately to aid the portability of application deployments. Role-based access control (RBAC) separates information for application components, workflows and environments.

Automated processes are repeatable and can assist with consistency and auditability. Automated workflows replace manual processes and will standardize environment change, reduce human error and relieve staff of performing time-consuming monotonous, repetitive tasks. An automated deployment pipeline that allows deployment portability across any environment will also promote automated rollbacks of versioned components.

Step 3: Orchestrate Toolchain

The automated deployment pipeline is expanded to include orchestration—running workflows on cross-platform, multi-stack hosts and against adjacent systems. Application-dependent systems such as virtual machines (VMs), Docker containers and continuous configuration tools are on-boarded along with elements of the IT service management (ITSM) stack. Environment lifespans are provisioned and de-provisioned on-demand or by timed lease to reduce virtual sprawl and provide self-service accessibility to developers, testers and other delivery personnel.

The entire application lifecycle is both automated and orchestrated—deployment, provisioning, automated QA testing, configuration, ITSM tools and notifications are all included in a single coherent deployment pipeline. Applications can progress from development environments to production and production-like environments with zero or near-zero downtime. Application updates, be they in-house-developed code or commercial off-the-shelf (COTS) updates for core backend business apps, are continually flowing to production.



Step 4: Continuous Delivery

The last stage is automated release management across the entire application portfolio and delivery chain. Applications are automatically executed in production if the versioned package passes automated tests and predefined criteria. This may not be suitable for every kind of app, but you now have this vital capability at your disposal.

Your automated deployment pipeline will provide a safe and consistent framework to quickly onboard additional applications across the enterprise portfolio—even core back-end business applications. Response times to line of business (LOB) requests by IT is shortened, allowing LOB to more quickly realize actionable responses to competitive pressures.

Introduction

Continuous delivery can seem daunting, if not impossible, for the enterprise where applications are monolithic, containing numerous dependencies, and where each deployment is seemingly unique. Add the fact that these dependencies often map to core back-end business applications whose deployments must also be managed and accounted for, and a tangled picture emerges for enterprise-scale deployments and production support.

The question is, where to start? The prevalent and ambient response is DevOps. The challenge here is that DevOps is a philosophy, not a practice like ITIL, a framework like COBIT or a standard like Sarbanes-Oxley. The philosophy embodies both culture and technology but leaves specifics for each organization to develop.

We propose a four-stage stairway to end-to-end continuous delivery as the most efficient and chronological way for the enterprise to achieve full agility. It allows organizations to establish just where they sit in the stairway and how to progress toward the goal of enterprise-scaled continuous delivery. It is a blueprint to achieving continuous delivery success that encompasses the entire enterprise application portfolio and cross-platform, multi-stack environments, whether they be private, public or hybrid clouds; the traditional data center; or legacy systems, such as the mainframe.

While ultramodern, cloud-native companies dominate the headlines for entirely agile app development and deployment, the reality is that most enterprises must modernize the delivery mechanics of legacy systems and alter monolithic architectures before they can compete with cloud-native companies.

Another truth is that if they do not adapt soon and embrace agile development and continuous delivery practices before the end of this decade, their business model may no longer be sustainable. The pressure is on CIOs to deliver a bold, courageous digital vision quickly, so a defined, tried-and-tested game plan is crucial.

Step-by-Step

Whether you move forward just one step or go all the way up to scaled continuous delivery, the CA Technologies stairway to continuous delivery will help your company make significant strides toward becoming a fully agile enterprise. It will equip your IT department with different methods of software delivery to tackle fluctuating and unpredictable demands from competitors, market pressures or LOB initiatives.

This model is adaptable to core back-end business apps, such as Siebel, and for modern digital apps that use the latest development frameworks and platforms. It is principle-based and criteria-guided, so both your present state and desired state are part of the maturity and growth plan.

The objective is to gain the agility needed to constantly adapt and improve, while maintaining control and stability of mission-critical apps as you make updates on a continual basis. The CA Technologies stairway to continuous delivery will provide your enterprise with a starting point and an incremental methodology for improving service delivery and feedback response. Applying continuous delivery best practices and tooling to building an automated deployment pipeline of both digital apps and core back-end business apps will then be accomplished.

While continuous delivery may seem like an unobtainable dream, CA provides a tried-and-tested route toward it, based on extensive experience. The first two steps allow any business to automate within the toolchain, and the last two allow you to orchestrate across the toolchain for full control and scaled agility.

Why CA Technologies Automation Solutions?

CA Technologies invites you to take full advantage of our considerable heritage: over 20 years in production automation, and a proven track record of helping companies achieve rapid and effective digital transformation. We are a trusted partner, giving CIOs the guidance to affirm their vision for the company's digital evolution.

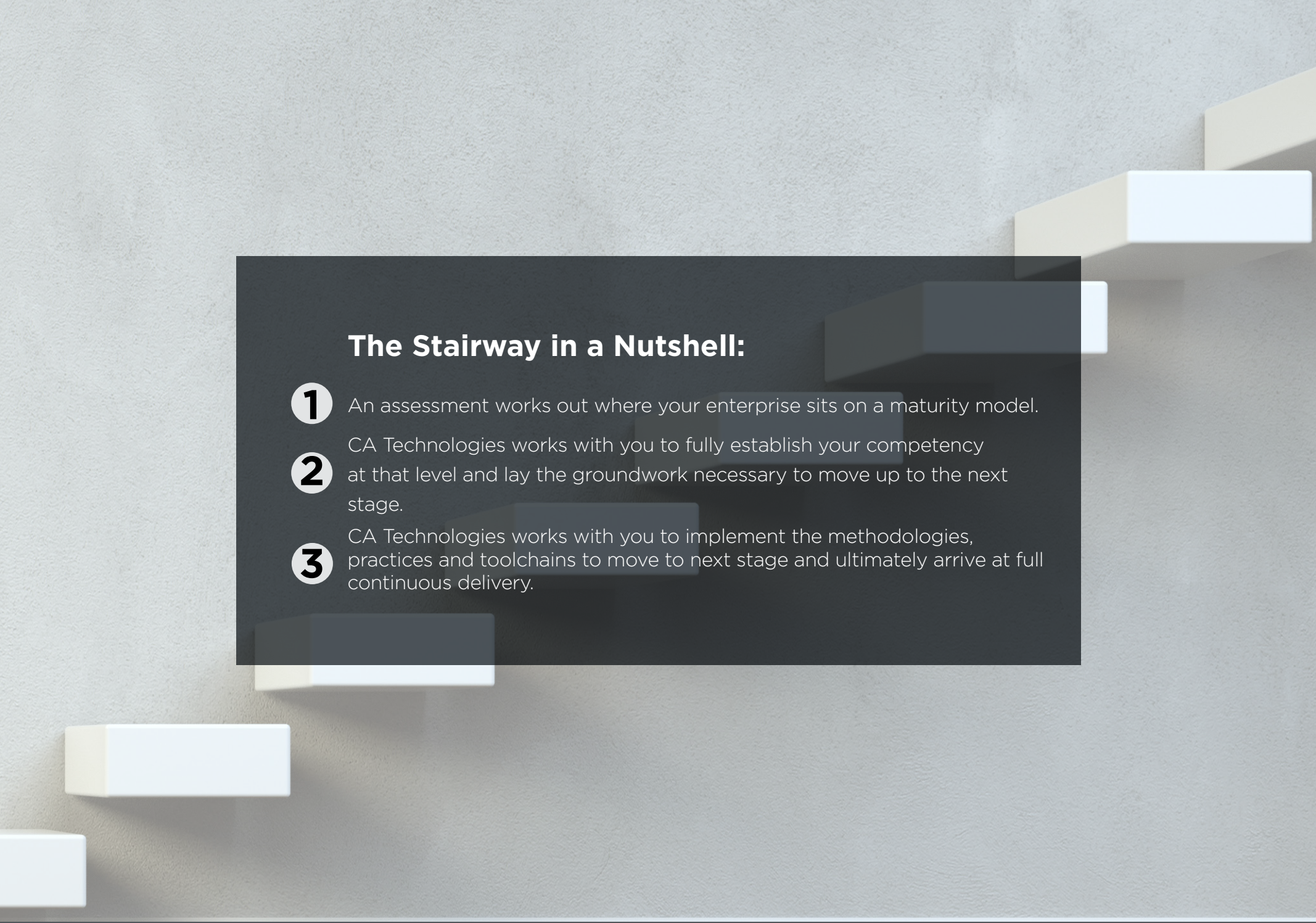
CA Technologies provides unrivaled, native cross-platform, multi-stack support, which allows the enterprise to include a broader technology stack in its deployment pipeline. Whether deployment environments are virtual, public/private/hybrid clouds, Docker or even mainframe, CA Technologies has you covered. Additionally, we are the only vendor that provides a multi-tenant product to give enterprises the ability to comply with separation of duties, create isolated sandboxes, secure production workload or host the product in the cloud.

While many vendors will sell complete toolchains, CA Technologies allows its customers to utilize any best-of-breed tool along the deployment automation pipeline. The Automic® solution automates within and orchestrates across the DevOps toolchain.

Our toolchain, cross-platform and multi-stack agnosticism allows the enterprise to define a deployment automation pipeline shared by all delivery contributors: developers, testers and operations. The same automated deployment process used in development is also used in test and production.

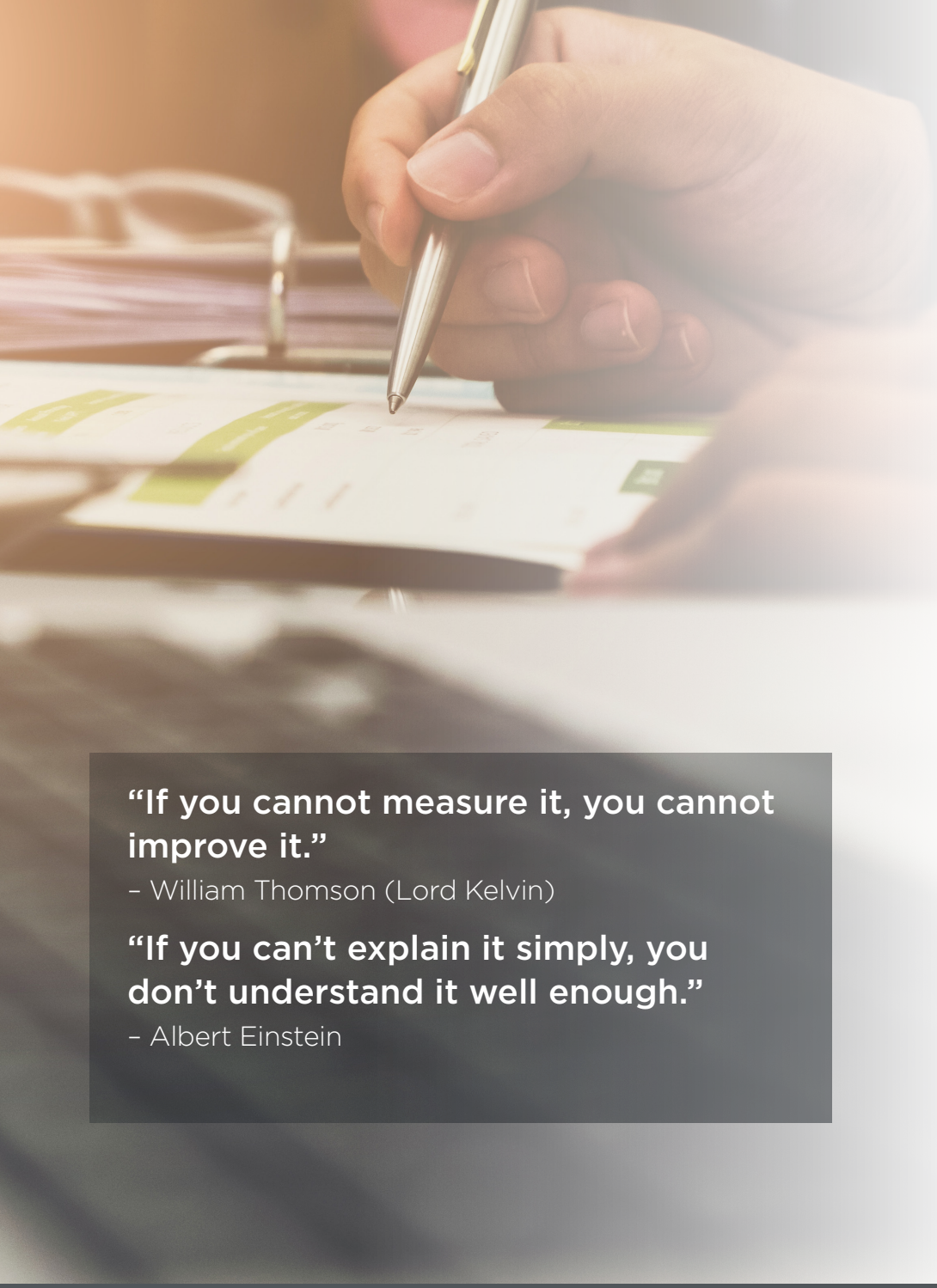
Using the same automated deployment process across environments is a critical enabler on the road to continuous delivery, because the only way to have full confidence in an automated production deployment is to have tested that same process dozens, if not hundreds, of times in lower environments before using it in production. These deployment iterations come from the fact that developers deploy all the time, QA less often and production less often still. The deployment process itself has been tested and vetted before being used in production.

Over 2,600 customers rely on the Automic solution to automate crucial business processes that run and grow their business. As the largest automation pure-play ARA vendor on the market, we at CA Technologies are the automation and DevOps experts.



The Stairway in a Nutshell:

1. An assessment works out where your enterprise sits on a maturity model.
2. CA Technologies works with you to fully establish your competency at that level and lay the groundwork necessary to move up to the next stage.
3. CA Technologies works with you to implement the methodologies, practices and toolchains to move to next stage and ultimately arrive at full continuous delivery.



“If you cannot measure it, you cannot improve it.”

– William Thomson (Lord Kelvin)

“If you can’t explain it simply, you don’t understand it well enough.”

– Albert Einstein

Step 1: Manual and Scripting

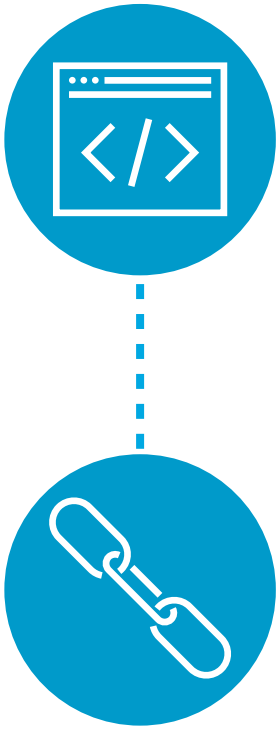
The first step toward continuous delivery is to complete the CA DevOps Maturity Assessment. The assessment collects the metrics and task definitions of existing practices and processes as well as defining the goals of your IT department and LOB. Delivery contributors will identify any currently used tools for inclusion or exclusion in producing a coherent deployment automation pipeline.

This data will be used as a baseline for measuring improvement, success and accountability. It will benchmark your enterprise against critical dimensions of the maturity model to work out if and where gaps exist regarding skills, tools and process.

CA Technologies will grade your company on:

1. Culture and strategy
2. Automation capabilities
3. Structure and processes
4. Collaboration and sharing

Your results can be compared to the CA Technologies standard to see where your company sits on our DevOps competency scale. You can also be graded against your peers, industry and region or ranked globally.



Manual Steps and Scripts

Current manual steps will need to be replaced by automated processes. Ideally, no environment will be touched by human hands for deployment tasks. Using automated mechanics for every environment reduces human error and configuration drift and makes error resolution far more predictable. Accurately documenting all manual steps, procedures and processes is essential to translating them into automation workflows.

Any scripts that are currently used to deploy applications in any and all environments will be identified and reviewed, and their role assessed. Environment-specific details such as credentials and configuration information are removed from scripted logic to make the logic portable. Assigning role-based access control (RBAC) to environment-specific details will also be identified to ensure controlled collaboration throughout the deployment pipeline.

Automating the DevOps toolchain

One of the most common challenges for enterprises looking to automate their deployment pipeline is operations' lack of confidence in using developer-specific tools in production environments. Automic Continuous Delivery Automation is a production-proven technology that allows developers to use the tools they are familiar with while providing operations with the controls, auditability and security they require. They can use the same process to deploy and run their applications in development environments that operations will be comfortable with using in production and production-like environments. Islands of automation are now aligned along a digital assembly line that all delivery teams can use.

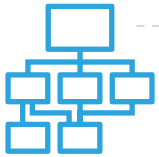
Once a strategy for the digital assembly line is in place, additional tools, if needed, may be properly assessed in the context of building a coherent deployment pipeline. It is important that each member of the DevOps team understands the capabilities and role of each tool in the deployment pipeline and how it will aid overall throughput. An automated and yet collaborative deployment pipeline provides a foundational base for the journey toward continuous delivery. It will allow you to move on to step two and continue your progression.

Next Steps

1. IT and LOB shareholders will receive assessment findings containing recommendations and a go-forward strategy.

2. A blueprint for automating and standardizing people steps, scripts and tasks will be provided.

3. Metrics and performance data will be uncovered, reviewed and used as baselines to give shareholders the ability to measure success and accountability.



4. A repeatable best practice will be designed into automated workflows to provide more predictive, consistent and auditable processes. Objects, artifacts and/or items currently not automated or just semi-automated will be identified and earmarked for full automation.

Step 2: Automate Deployments

The aim of this step is to automate the deployment pipeline. Now you are ready to map defined practices, manual tasks, and people steps and tasks to automation workflows within Automic Continuous Delivery Automation.

Environment Model

Environment configuration data, properties and/or items will be modelled separately from the workflows and application models so that automation deployments are portable. The configuration information will be broken down by delivery contributors and by environment and assigned RBAC to ensure that only the right contributors own and control the values of their domain-specific information. Modeling information and access this way provides controlled collaboration, with the flexibility to scale the deployment automation pipeline across the modern enterprise.

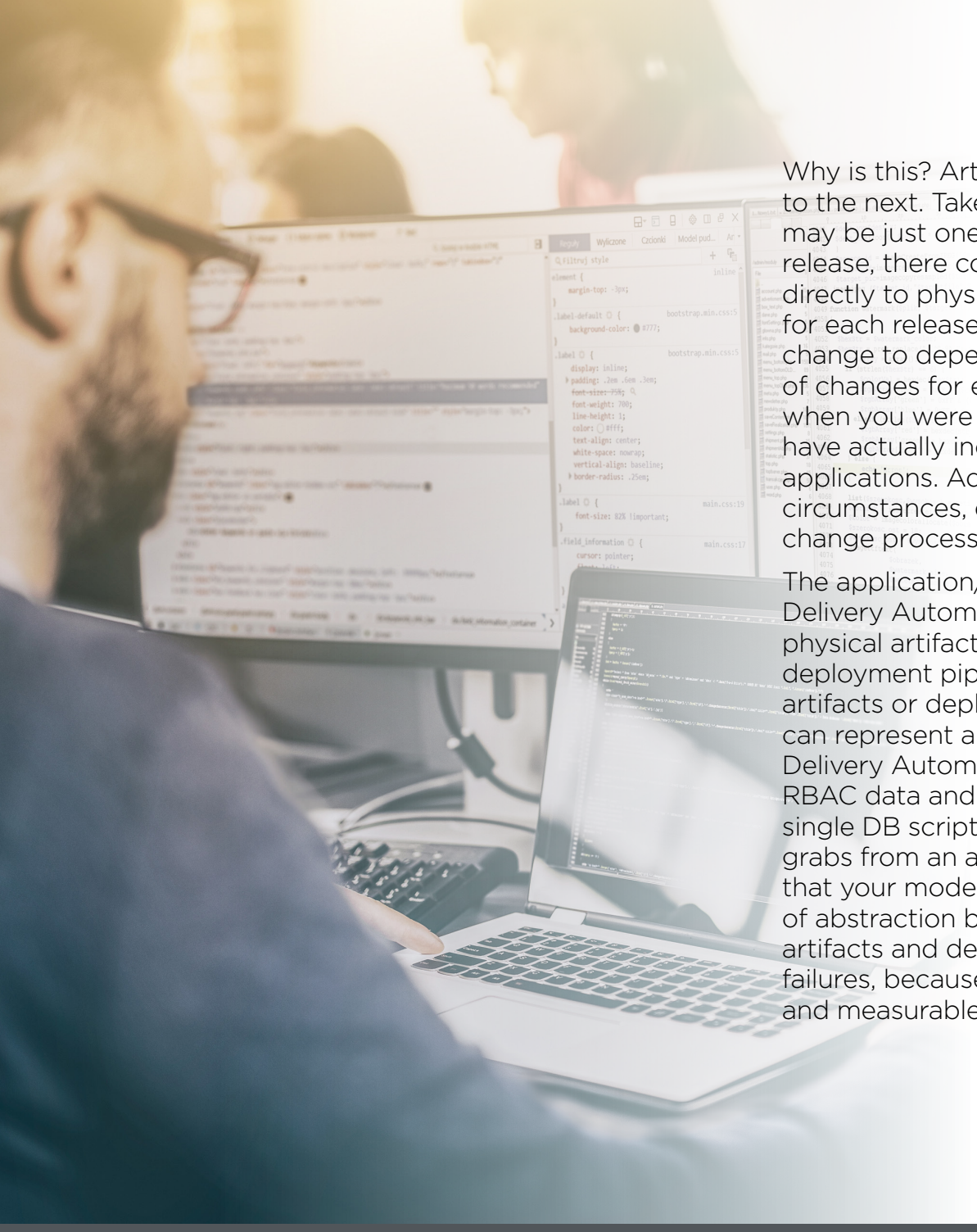
Environment promotion state flows and gated criteria will also be set up to allow applications to be promoted from one environment to the next along the deployment automation pipeline. Promotion or termination of versioned deployment packages in any environment occurs should they pass or not pass defined criteria or approval scrutiny.

Application Model

Applications and their components will need to be defined and mapped to a model that represents the logical deployment architecture, not necessarily the design architecture. Application and environment models are separate to ensure deployments are portable across any platform or stack.

When identifying applications and their components, it must be remembered that many applications are composite applications—a large "application" consisting of a collection of smaller, separately versioned applications. Sometimes composite applications are referred to as platforms. In any case, we need to identify the smallest unit of logical, architected or versioned functionality than can be considered an application in its own right, remembering that applications must also include one or more smaller deployable entities called components (Web, storage, DB, biz logic, etc.). Going through the exercise of splitting up apps and their deployable components will help simplify the deployment process and is vital to maintaining an application that is always deploy-ready.

Physical artifacts such as JARs, DLLs, DB scripts and so on should not be mapped one-for-one to logically modeled app components. Many application release automation vendors do this, but it increases the cost of maintaining your automated workflows and inhibits the benefits of using a standardized best practice.



Why is this? Artifact counts can change in number from one release to the next. Take, for example, DB scripts. In one release there may be just one DB script that needs to be executed. In the next release, there could be 20. If your modeled app components map directly to physical artifacts, you will have to change your model for each release. Changing the model will most likely also require a change to dependent workflows. Once you're making these kinds of changes for every release, you are not much better off than when you were manually executing tailored scripts. Indeed, you have actually increased the overhead and friction for deploying applications. Additionally, should a deployment fail under these circumstances, everything is brought into question, including the change process itself, in an effort to discover what went wrong.

The application/component models built in Automic Continuous Delivery Automation are a logical representation of the kinds of physical artifacts to be deployed. This ensures the automated deployment pipeline is unchanged, though the number of artifacts or deployment targets may change. For example, you can represent a DB app component within Automic Continuous Delivery Automation's model. This component can be assigned RBAC data and automation steps. It can then execute against a single DB script of thousands of DB scripts, depending on what it grabs from an artifact repository at run time. You can rest assured that your models and workflows stay consistent. Keeping a layer of abstraction between the model and workflow and the physical artifacts and deployment target reduces troubleshooting efforts for failures, because the apps are consistently released in a predictable and measurable way.

Packaging

Each release requires exact versions to match a requested deployment. Binary artifacts, configuration files, middleware scripts, OS packages/installers and database scripts can all be part of every deployment. Automic Continuous Delivery Automation's packaging mechanism assigns the appropriate collection of modeled app components to a versioned deployment package. The underlying automation engine will intelligently pass over app components that are not part of the deployment package. Automic Continuous Delivery Automation deployment packages are tracked through the promotion path to production and across the deployment pipeline.

Validations and Rollbacks

One of the principles of continuous delivery is always to leave an environment in a stable state. Ensuring environment stability implies two things: (1) that validation checks need to be performed in order to confirm pre-deploy assumptions, and (2) that rollback procedures are in play so failed deployments can be partly or wholly undone.

Validation Checks

Some checks should be considered, such as ensuring that the environment meets the physical specs needed by the app, that there is enough storage, or that the correct version of the app we are about to replace is in fact currently installed. There are more criteria to be sure of, and identifying these must be done so the deployment automation workflow can perform prescribed changes safely.

Rollbacks

Rollback criteria and automated procedures will be set up to ensure any deployment environment is left in a stable state should a deployment fail. The execution of rollbacks can be user-initiated or automatically executed if a deployment or portion of a deployment fails. Rollbacks can occur at the app component level or higher up, at the application level, where performing the removal of all deployment components is done.

Many ARA vendors suppose that simply redeploying the previous version of a successfully deployed application is a best practice, but this approach is error-prone. Each versioned release of an application may touch different configuration and system file/settings. A configuration setting altered during a failed deployment may not be in the change manifest for the previous release. Configuration residue left behind can destabilize the previous version of the app, leaving support staff perplexed and left to attempt a needle-in-a-haystack troubleshooting exercise.

Workflows

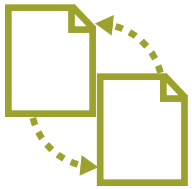
If deployment procedures and processes are not automated, they are not repeatable. Under a manual methodology, each deployment will be different, error-prone, and will lack auditability. Automated workflows will be collaboratively designed by delivery contributors to incorporate the information and definitions discovered in the maturity assessment. These workflows will replace manual tasks and will provide a repeatable, consistent and auditable process to install, upgrade, patch or remove an application in any environment (cross-platform or multi-stack).

Automic Continuous Delivery Automation workflows represent your organization's standardized, pre-approved change process for carrying out deployments in any environment along the deployment pipeline. Pre-approved does not imply exemption of approval gates or authorizations for execution of these workflows within environments. It means that delivery contributors collectively and collaboratively define a best practice that will be the only means of executing deployment changes in any environment, should that deployment be authorized.

Automic Continuous Delivery Automation separates workflows from application and environment models. This gives organizations the ability to reuse established best practices as a foundation for additional onboarding of applications and prevents the common IT practice of continually reinventing the wheel.

Pre-Build Functionality

Automic Continuous Delivery Automation provides a continually expanding set of pre-built integrations called “actions.” These can be downloaded at any time from the CA Automation Marketplace. CA Technologies keeps its marketplace up-to-date with content developed by both CA Technologies and customer contributions. What makes the CA Automation Marketplace unique is that you are not limited to downloading singular integration points; you can download full best practice workflows.



Workflow Versioning

Automic Continuous Delivery Automation automatically versions changes made to workflows, indicating when updates are made and by whom. If a change introduces a problem, it can quickly be reverted to a previous version. RBAC prevents workflows from being edited by unauthorized personnel.

Security

Production and production-like environments should be completely locked down so that changes in those environments, like application deployments, are only made with automation mechanisms such as Automic Continuous Delivery Automation. This provides a reliable way to audit changes, troubleshoot problems and recover from failures more predictably. It also ensures that standardized best practices, in the form of automated workflows, are the only change mechanisms.

Automic Continuous Delivery Automation provides authentication, encryption and advanced security measures to prevent hackers or unauthorized personnel from reading step/workflow details, executing steps/workflows on remote hosts, or hijacking those hosts in an attempt to pretend to be a valid endpoint.

Step 3: Orchestrate Toolchain

The automated deployment pipeline or digital assembly line built in the previous step will expand to include orchestration (running workflows on different hosts and separate supportive systems). Systems that are supportive of and dependent on your applications, such as provisioning and continuous configuration tools, will be onboarded along with the full ITSM stack.

Supportive systems, platforms or stacks may include, but are not limited to:

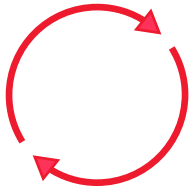
1. Infrastructure provisioning and configuration
2. Cloud/Container environment and sandbox management
3. IaaS, PaaS, SaaS
4. Change processes
5. Service management
6. Automated QA Testing
7. Approval policies
8. Corporate policies
9. Regulatory, compliance or legal procedures
10. Notification systems

We will now include deployment environment lifespans as part of your automated deployment pipeline(s). Provisioning environments prior to deploying the application and de-provisioning as soon as they are no longer needed will save money and increase application stability. Environments become on-demand, reducing or removing virtual sprawl while easing the ability to troubleshoot versioned instances.

Your processes will now enjoy the benefits of:

1. Zero or near-zero downtime deployments
2. Automated pre-deployment validation checks
3. Automated post-deployment smoke tests
4. Multi-stack provisioning
5. Cross-platform continuous configuration
6. Inclusion of full ITSM stack within your deployments pipeline

Apps are progressed from development to production. This frees up the capacity of delivery contributors to work on more strategic efforts, while enabling faster feedback loops and a continuous flow of iterative updates with little or no downtime.



Step 4: Continuous Delivery

Despite the implications and connotations of continuous delivery, its defined purpose is to make deployments available for production and to execute within production when requested. The progressive (perhaps aggressive) next step is continuous deployment, which, by definition, automatically executes deployments in production if the deployment package passes automated tests and predefined criteria.

Your defined deployment pipeline's last stage is an automated, hands-free deployment into production. The path from development to production is fully automated and occurs without human intervention. Obviously, continuous deployment isn't for everyone, or for every app or circumstance, but this should be the ideal to strive for—an ability you should have at your disposal should you want to use it.

At this final stage, all apps within the enterprise application portfolio can be onboarded to the deployment pipeline framework. Any application compatible with continuous deployment is enabled. Inclusion, consideration and respect for regulatory, compliance, audit, and ITSM tools and practices are all part of the deployment pipeline framework to ensure software deployments occur safely, reliably and within predictable timeframes.

Above and beyond the benefits of safety, predictability, reliability, separation of duties, and even the automation itself (though it is the backbone for all this), the larger benefit will be a more efficient continuous improvement cycle. LOB experience a decrease in the amount of time it takes to react to market or competitive pressures, while application quality and relevancy exponentially increase because the feedback loop to developers is faster and more frequent. Success begets more success.

At this pinnacle, you are now able to control the delivery process for any app. Multiple deployment pipelines will be defined, monitored and reported on, but you have a deployment pipeline framework from which to grow and build. When software delivery has been industrialized within your organization, you will realize the following benefits:

Employee Empowerment

One of the key tenets of an automated deployment pipeline is a self-service system allowing developers, testers, operations or support personnel to pick the version of the application they want and designate or provision the intended deployment environment. This ability puts delivery contributors in better control of their work, which increases the quality of the application and the level of constructive collaboration between team members.

Error Reduction

Following a repeatable process makes errors outliers, and it is ideally the only way to push changes into production. Should an error occur, the mean time to repair is far more predictable, perhaps automated by a rollback or a redeploy of the previous version.

Employee Morale

Less stress, working fewer evenings and weekends, focusing less on repetitive tasks and more on innovative or strategic tasks. This change in morale not only moves the enterprise forward, but it also increases job satisfaction and employee retention. Everyone wants to feel like they are more than a robot mindlessly performing repetitive tasks. Employees want to feel they are contributing to something that matters. Automation provides employees with the time to be more strategic in their thinking and more constructive in building solutions. When employees have time to work on more valuable tasks, they feel more valued, and when people feel appreciated, they go the extra mile.

Revenue Growth

Enterprise Management Associates' (EMA) 2015 DevOps/Continuous Delivery research showed that companies that increased their deployment frequency by 10 percent or more were 2.5 times more likely to experience double-digit revenue growth than companies that increased deployment frequency by less than 10 percent, or that flatlined or decreased.

Conclusion

DevOps is a philosophy that embraces culture and technology. Continuous delivery binds culture and technology, and the Automic solution provides a tried-and-true approach to reaching peak continuous-delivery potential.

Automation is the backbone for both DevOps and continuous delivery. However, implementing islands of automation specific to tasks and domain expertise is not enough to experience transformative benefits. Indeed, merely adding a tool here and a tool there to automate this and that may contribute to a degradation in service level, increase availability problems and decrease personnel performance.

Automic Continuous Delivery Automation is the core or backbone to delivering agility to the enterprise, automating within and across the DevOps toolchain, and includes supportive systems and platform support into the deployment pipeline. Transforming boardroom inspiration into marketable modern digital apps takes a fraction of the time it used to take. Meanwhile, core back-end business apps can have a degree of agility that was once considered unobtainable.

Go from being disrupted to being the disruptor and invite CA Technologies to conduct a DevOps maturity assessment at your organization.

References

Humble, J., Molesky, J., & O'Reilly, B. (2015). Adopt Lean Engineering Practices. Lean Enterprise (p. 156). Sebastopol, CA: O'Reilly Media.

For more information or a product demonstration, please visit ca.com/automation.

For product information and a complete list of distributors, visit our website at: broadcom.com

CS200-368898-0619 June 11, 2019