TATA CONSULTANCY SERVICES

Getting IT Infrastructure Right by Implementing a DevOps Framework

Abstract

With improved internet bandwidth and increased mobile phone penetration, businesses are facing intense competition, resulting in rapidly shrinking profit margins, and surprisingly crunched product launch timeframes.

The DevOps framework helps improve an organization's time-to-market significantly by bringing together the twin functions of development and operations in the software application space. However, by implementing DevOps, organizations must incur radical changes to their technology, process, and support culture.
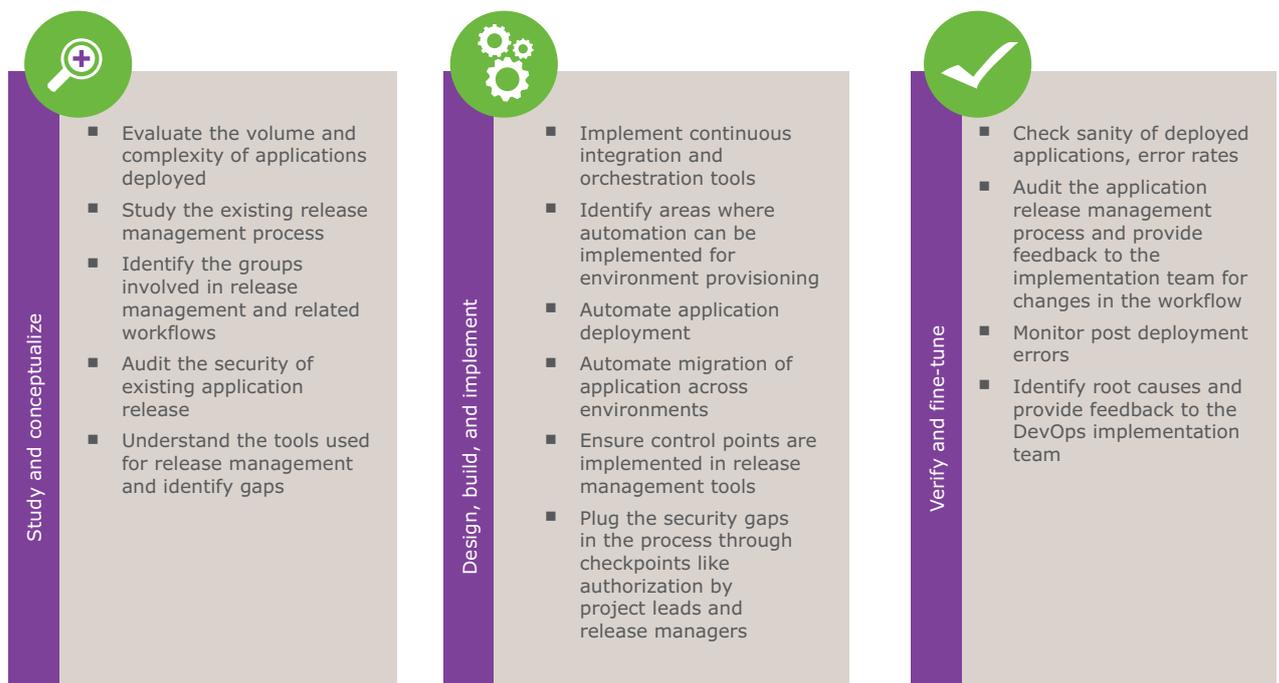
The Evolution of DevOps

Several factors have contributed to the evolution of the DevOps framework. These include virtualization and automation opportunities, the ability to automate environment provisioning using scripts, transitions in software development processes from the waterfall model to the Agile model, automation of application testing activities, introduction of release orchestration systems for enhanced collaboration among teams.

Stages in Setting Up a DevOps Environment

A DevOps framework is ideal for organizations requiring a significantly high number of application releases, new features implementation, and application upgrades in a short span of time.

Organizations often deploy complex, multi-tiered applications to ensure system robustness and scalability, risking delays in taking their products and services to market. The DevOps framework can address these issues. However, organizations must first evaluate and establish the need for implementing the framework by identifying the desired number of application builds or releases per week.

While DevOps is jointly implemented by development, testing, and IT operations teams, the effort required from an IT infrastructure and operations perspective is significant, and often overlooked. Implementing a DevOps based environment is envisaged as a three-stage process involving stakeholders, from both, application development and testing teams.

Study and conceptualize

- Evaluate the volume and complexity of applications deployed
- Study the existing release management process
- Identify the groups involved in release management and related workflows
- Audit the security of existing application release
- Understand the tools used for release management and identify gaps

Design, build, and implement

- Implement continuous integration and orchestration tools
- Identify areas where automation can be implemented for environment provisioning
- Automate application deployment
- Automate migration of application across environments
- Ensure control points are implemented in release management tools
- Plug the security gaps in the process through checkpoints like authorization by project leads and release managers

Verify and fine-tune

- Check sanity of deployed applications, error rates
- Audit the application release management process and provide feedback to the implementation team for changes in the workflow
- Monitor post deployment errors
- Identify root causes and provide feedback to the DevOps implementation team

Key Considerations for DevOps Infrastructure Implementation

Adopt an agile infrastructure

- The Agile software development methodology employs an iterative approach to develop and test software codes. This requires frequent deployment of codes and immediate testing of results.

- In terms of infrastructure support, moving to the Agile methodology requires the implementation of continuous integration (CI) software, which enables uninterrupted delivery across the software development chain. As the entire code repository stays integrated to CI systems, it is imperative to safeguard these from malicious access.

- As complexities related to code and environment increase, infrastructure teams will need to maintain CI systems effectively.

Move from manual to automated infrastructure environment build

Agile software development requires infrastructure that can be created dynamically, with minimal instructions. The infrastructure team plays a vital role in automating environment provisioning, which is a critical activity in DevOps implementation, while also supporting application deployment in software containers. Automation can be done through customized scripts or through commercial products available in the market.

Configuration management tools can create and replicate infrastructure based on codes. This has introduced a new concept called Infrastructure as a Code (IaaC), where the majority of infrastructure components are provisioned through codes. Most enterprises are still trying to virtualize physical systems and are contemplating moving to a cloud based service delivery model.

However, containerization of the environment is growing in popularity because it:

- Helps developers deploy the same application on different platforms, without rework

- Reduces application environment provisioning from several hours of effort to a few minutes

- Extracts configured web and application servers from production servers, and replicates them across systems

To rapidly adopt a DevOps framework, organizations are increasingly moving to appliance based middleware, which promises a 'cloud in a box' environment with in-built automation and orchestration.

Containerization may eventually replace virtualization as it substantially reduces the effort involved in application environment provisioning.

The IT infrastructure team must implement orchestration tools with high availability features, customize workflows with security access controls, and integrate service catalogs in the IT operations ticketing systems.

The DevOps teams should be equipped with the knowledge of release, service and change management, environment provisioning, application deployment, as well as automation and tools.

Leverage release management orchestration tools

It becomes necessary to integrate the release management methodology through existing service management tools such as IT operations ticketing and change management systems.

For instance:

- Releasing complex software in an agile development environment using orchestration tools such as CA Release Automation and BMC Release Lifecycle Management.

- Releasing new applications by configuring changes in the non-production and production environments.

- Orchestration of automated activities for code release across of delivery pipelines

- Integration of Service management tools with Release Orchestration tools

Enable organizational changes and encourage adoption of new skills

During the initial phases of the DevOps implementation process, a strong collaboration between developers and operations teams is a must. Developers' participation is critical during the environment build stage, while the support of the operations team is required during the application design, build, and release processes.

Addressing Challenges in DevOps Implementation

Implementing security controls

To reap the full potential of Agile software project delivery, organizations may have to bypass several security controls in the software development process while implementing the DevOps framework. However, the framework must be aligned with fundamental security measures such as team segregation, role based access to codes and systems, and ITIL service management processes

Implementing DevOps controls will go through four stages:

- **Align** the framework with fundamental security measures such as team segregation, role based access to codes and systems, and ITIL service management processes.

- **Opt** to bypass several security controls in the software development process.

- **Make** configuration items accessible to the right stakeholders and compliant with industry standard security frameworks like Common Objectives for Information and Related Technology (COBIT).

- **Administer** access restrictions (responsibility of the infrastructure team) and conduct internal and third party audits at defined intervals.

Introduce checkpoints in orchestration tools

Automating tasks such as building applications, testing the quality assurance (QA) environment, and production deployment may require some additional precautions, including manual check points that are essential when tool designers implement the workflow across application development and deployment in the QA or production environment.

The infrastructure team is primarily responsible for ensuring that automation bears the desired results and significantly reduces the amount of effort involved in entry-level tasks. Though administrators skilled in troubleshooting technical issues in the production environment will still be required, especially post framework implementation, using the DevOps framework reduces inter-team coordination.
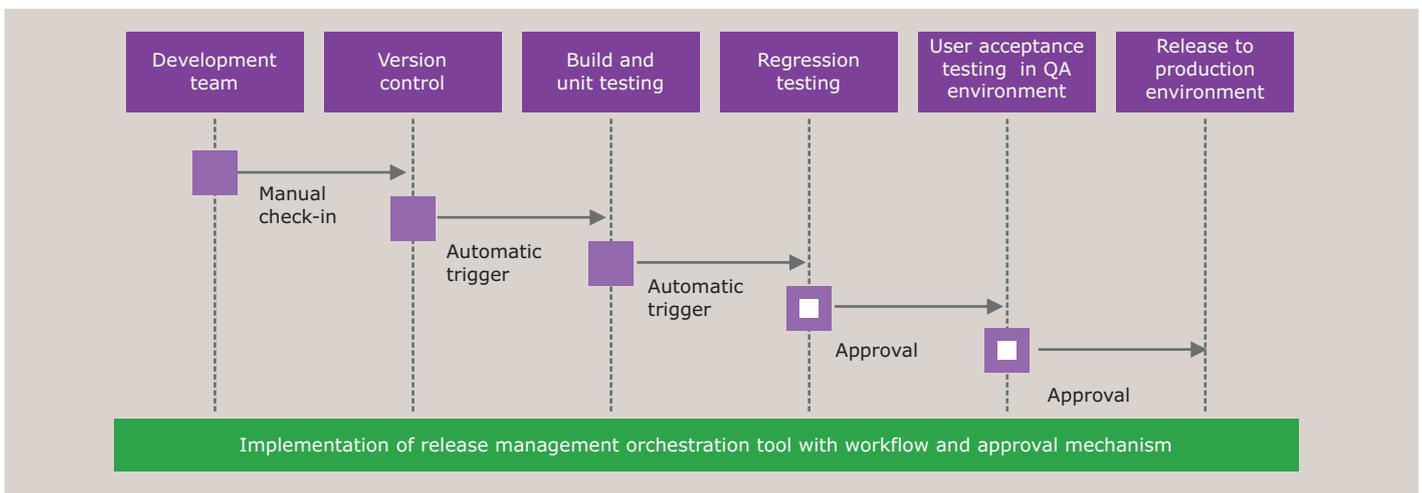

Figure 6: Implementing Controls in Orchestration Tools (Source: TCS Internal)

Rewiring Infrastructure Services to Make the Most of DevOps

Through automation of environment build and application release management activities, the DevOps implementation will shakeup the existing support structure within operations. A new DevOps team will be created to maintain the automation and orchestration framework with minimal resources

Software version management tools and customization of release workflows will be other major activities within the scope of IT infrastructure services.

A majority of the IT support processes are built on ITIL V3, which does not include the DevOps framework. However, the ITIL v4 is expected to include application release management process that takes into account automation and controls that will be implemented in the environment

Conclusion

DevOps is going to gain prominence as infrastructure virtualization and automation increase over the next few years. The factory model of application release will soon become a reality. With environment provisioning requiring minimal effort, an organization's infrastructure operations team should focus on developing skills to architect new environment configurations for complex web applications.

Implementing the DevOps Framework will introduce automation, orchestration of operational activities, collaboration among development, test and operation teams, and revision of control mechanisms across the enterprise. This will ensure faster delivery of higher quality software products online, offering increase customer satisfaction

About The Author

R. Nanda Kishore

Contact

Visit TCS' IT Infrastructure Services unit page for more information

Email: itis.presales@tcs.com

Subscribe to TCS White Papers

TCS.com RSS: http://www.tcs.com/rss_feeds/Pages/feed.aspx?f=w
Feedburner: http://feeds2.feedburner.com/tcswhitepapers

About Tata Consultancy Services Ltd (TCS)

Tata Consultancy Services is an IT services, consulting and business solutions organization that delivers real results to global business, ensuring a level of certainty no other firm can match. TCS offers a consulting-led, integrated portfolio of IT and IT-enabled, infrastructure, engineering and assurance services. This is delivered through its unique Global Network Delivery Model™, recognized as the benchmark of excellence in software development. A part of the Tata Group, India's largest industrial conglomerate, TCS has a global footprint and is listed on the National Stock Exchange and Bombay Stock Exchange in India.

For more information, visit us at www.tcs.com

Experience certainty.   IT Services
                         Business Solutions
                         Consulting

TCS Design Services IM I 12 I 16