Infosys cobalt

WHITE PAPER

ARCHITECTING DIGITAL TRANSFORMATION RIGHT WITH DEVSECOPS

Infosys®
Navigate your next

Two out of every three adults worldwide now make or receive a digital payment, according to the World Bank's 2021 Global Findex database[1]. In addition to financial inclusion, the COVID-19 pandemic catalyzed an overall surge in digital transactions and interactions. But are IT systems and applications keeping up with the increased load? A 2022 report from CISQ pointed out that costs linked to poor quality software had risen to at least $2.41 trillion in the United States, accompanied by a growth in accumulated software Technical Debt (TD) to ~$1.52 trillion, which would indicate otherwise[2]. The report also highlighted that software supply chain problems with underlying third-party components (especially Open-Source Software, aka OSS) had risen significantly. At the same time, software vulnerabilities have led to greater losses from cybercrime.

Given this, it is not surprising that the IT teams are in constant maintenance mode. A global survey in 2023 by Dynatrace of 1,300 CIOs and senior DevOps managers in large organizations reported that continuous software release cycles made it more and more difficult to ensure code security and dependability– the study found that "78% of organizations deploy updates every 12 hours or less", while "54% deploy updates every two hours or less"[3]. The price of coding at speed is being paid by quality. DevOps teams spend "nearly a third (31%) of their time on manual tasks involving detecting code quality issues and vulnerabilities, reducing the time spent on innovation", the Dynatrace report notes. But digital transformation pressures mean that 55% of the organizations surveyed were "forced to make tradeoffs among quality, security, and user experience to meet the need for rapid transformation". Furthermore, unrealistic deadlines and the pressure to deploy releases can result in poor version control, further accumulation of technical debt, production failure and, eventually, risky software releases, which further feed the vicious development cycle.

How can IT developers address this? The answer from the C-level is deploying DevSecOps for sound development and delivery processes. In the Dynatrace survey, 94% of CIOs said that expanding DevSecOps wider within their organizations was key to driving faster, more secure software releases and a stable path to digital transformation.

[1] https://www.worldbank.org/en/news/press-release/2022/06/29/covid-19-drives-global-surge-in-use-of-digital-payments

[2] Cost of Poor Software Quality in the U.S.: A 2022 Report - CISQ (it-cisq.org)

[3] Global CIO Report Reveals Growing Urgency for Observability and Security to Converge | Business Wire

Why DevSecOps

DevSecOps is a framework that mitigates quality risks by bringing together development, security, and operations from start to finish of the software development lifecycle. Integrating Security enables rapid release dev cycles to stay within organizational IT safeguards. In addition, the framework allows security concerns to be addressed as they occur within Continuous Integration (CI) and Continuous Delivery (CD) workflows. This not only forestalls the ballooning of security concerns into enterprise-wide showstoppers, but also makes fixing security issues much cheaper. Key benefits include:

- **Early detection of vulnerabilities**
- **Increased team velocity as bugs are identified and fixed**
- **Greater efficiency without the overhead of technical debt accumulation**
- **Shorter delivery cycles as software can securely move to production without last minute rework and iteration**
- **Automation of repetitive tasks**

The Evolution of the idea

Back in 2007-08, a (then) revolutionary idea was floated to bring those who wrote code together with those who deployed it. This removed a major disconnect in the SDLC process that was directly responsible for undeployable code and unhappy customers. Instead of code being developed and then tested/deployed on production platforms linearly, the DevOps movement created a continuous integration (CI) and delivery (CD) cycle made possible by teams working together instep.

A few years later, a third component was added – security – in view of increasing cybersecurity threats. Earlier Security used to make a late entry to the application cycle, post-development, during deployment and testing. But security checks were unable to keep up with the pace of releases possible through agile and DevOps practices, and delays resulted due to the late identification of security vulnerabilities.

The new three-way framework addresses security issues as they arise from the CI-CD pipelines. Shift Left is the practice of slotting quality and performance activities early in the development process and right through application development. It helps teams anticipate issues that may arise with respect to performance/quality right through the development process as well as avoid costly re-development late in the cycle.

Therefore, essentially DevSecOps is an orientation to a certain culture where team roles are aligned along the lines of cooperation. The emphasis is on responsiveness to business needs, with security as the shared responsibility. To actualize the development process, CI/CD provides the framework, including automation and tools, for continuously identifying requirements, feeding into the development, integration and testing cycle and then onboarding/monitoring deployed systems.

How can a DevSecOps strategy succeed?

Need for collaboration:

DevSecOps requires combining multi-functional teams and removing team silos.

Automation:

Needs to be built early in the process, part of the Shift Left paradigm – for instance, replacing manual security configurations by automation. Processes like access management and vulnerability checks must be automated right through the development cycle.

Security first mindset:

Security cannot be an afterthought but inbuilt into the development process from the get-go. Developers need to understand security requirements and design the software with that in mind from the beginning.

Key implementation challenges

Complicated branching/pipeline setup:
A practical complexity encountered in CI implementation, this may lead to lack of adoption or be error prone.

Security vulnerabilities in CI/CD pipeline:
Can cause security issues if not implemented correctly, and additional controls, such as for example processing to manage sensitive data, need to be put in place.

Inadequate training:
Not all developers have the formal security skills to implement DevSecOps practices. Formal training may be needed to fill this gap.

A problem of plenty:
There is a multiplicity of tools to choose from in the DevSecOps environment, which gets complicated when there are differences in toolsets between security and other teams. Development environments spanning several public clouds, each with its set of automation for security, can also spawn alert fatigue among developers. Mastering multiple tool sets requires investments in time and resources as learning curves can become steep, resulting in higher TCO.

Choosing the right tools

As enterprises embrace platforms such as Salesforce.com, which comes inbuilt with Low Code No Code (LCNC) capabilities resulting in very nimble solution delivery, automation of the software engineering processes is essential. LCNC is an enabler of quick customization, but this adds more complexity to using the platform, as thousands of various types of metadata files and dependencies must be managed. Adopting DevSecOps helps manage this complexity so that enterprises can fully harness Salesforce functionality. DevSecOps for a Salesforce platform automates the process of testing and validating changes, further enforcing stricter solution delivery norms as prescribed for a multi-tenant platform, while enabling demand-led customization as well as the deployment of changes to multiple development environments and production systems seamlessly.

For long, Salesforce has relied on its partner ecosystem to provide solutions for CI/CD automation - evident in the rich cast of tools available - that cover the gamut of services, including, but not limited to source control, profile management, sandbox management, pipeline orchestration, configuration management, application lifecycle management, data backup and recovery and cyber security.

Commercial toolkits available include copa.do, AutoRABIT, Gearset, Gitlab, Bamboo, etc; enterprises can also look to open-source tools overlaid on Jenkins, an open-source automation server.

With such a partner ecosystem available for dominant platforms like Salesforce, the path to smoother SDLCs via CI/CD must be paved with the right tools. The key is to start with a sound environment/sandbox strategy with a reporting overlay, such as a dashboard that can report on development status, error percentages, version control and vulnerabilities. Over this layer, one needs more advanced analytics that can report on progress to plan.

Implementing it right

To ensure that their DevSecOps strategy succeeds, organizations need to address implementation challenges and keep key processes in mind from planning through to execution and tracking.

Planning stage

Clear establishment buy-in:

Reflected by definition of goals relating to productivity, business KPIs, timelines, security requirements and end-user key impact areas

Secure by design:

Data privacy and application security concepts to be embedded at design phase itself with all stakeholders fully aligned

Environment provision:

Need to provision tools and resources to support a CI/CD strategy

Pipeline Branching, merging standards and conflict resolution:

Need to adopt simplification in branching as well as processes and tools to manage auto merging and code conflict

Execution stage

Checklists:
Process adhesion to be enforced through checklists for key deployment and migration procedures

Performance impact check:
Vulnerability assessments to be placed front and centre, reflecting the security first mindset

Key metric measurements:
Analysis and tracking of KPIs is critical

Control and Feedback stage

Leveraging analytics:

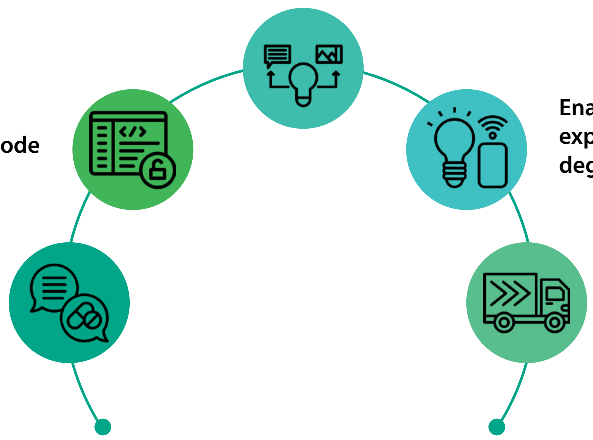
Knowledge captured from the new environment to be applied for deep analysis of process efficiencies

Course correction & refinement:

Analytics results to be deployed into feedback loops to meet goals and business KPIs

Where is DevSecOps headed?

As is the case for multiple areas in the software development business, the future of DevSecOps is entwined with the embrace of AI. This could play out as more intuitive automation, self-healing through advanced analytics, and turning data into answers with observability. These capabilities could manifest themselves in functionality such as:



Delivery of contextual solutions

Security as code

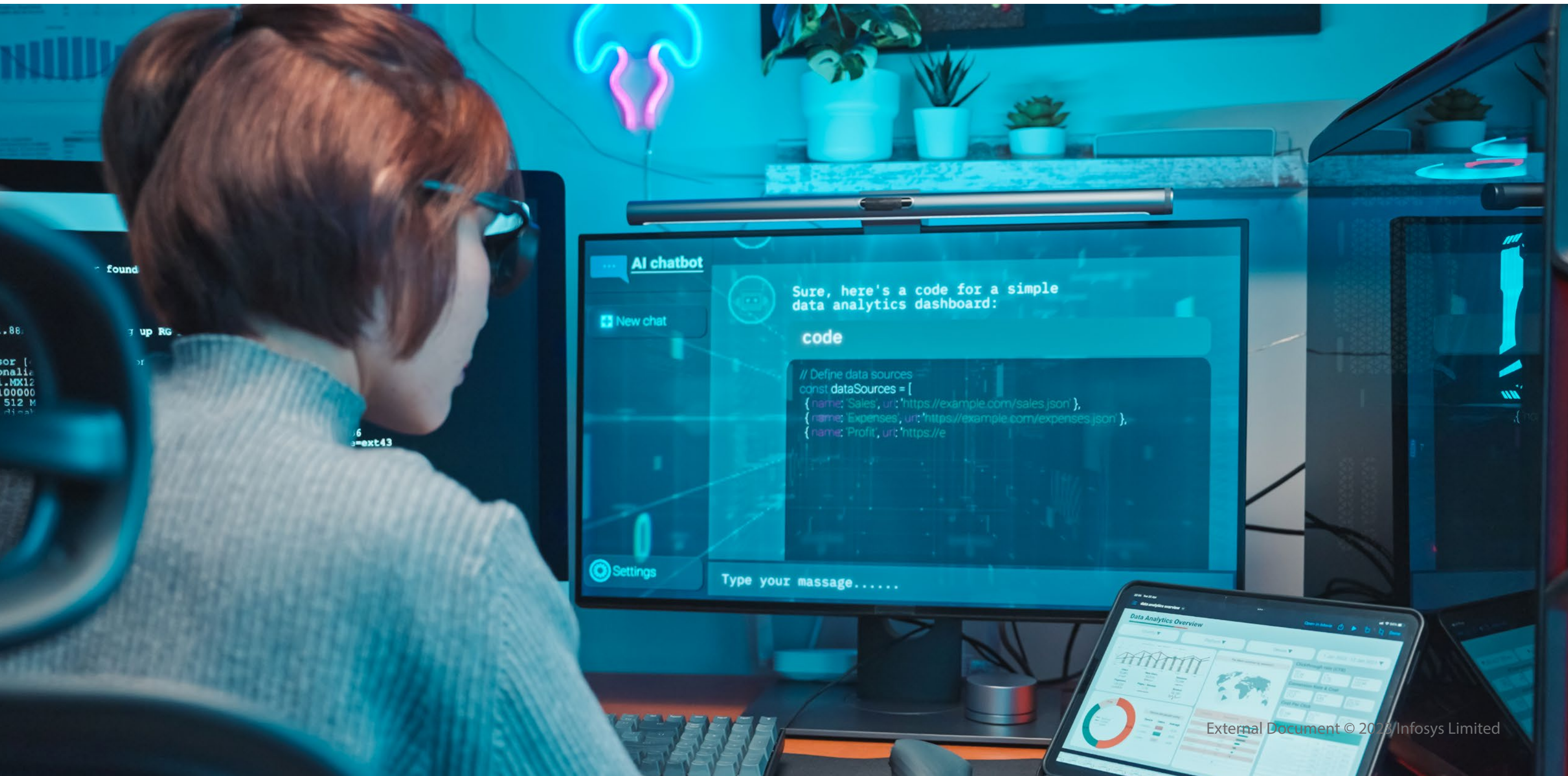
Enablement of integrated human expertise across teams for a higher degree of efficiency

Auto-remediation of security issues

Greater delivery predictability

A sound DevSecOps framework will also be required as more enterprises move towards LCNC applications. LCNC democratizes software development and brings the business side into play within development teams, but it could also create the perfect storm for the SDLC process as organizational IT frameworks and safeguards are diluted by new developers. Intuitive DevSecOps toolkits can spot and course-correct before costly errors are made.

In essence, DevSecOps balances agility and safety within the SDLC. It is the right insurance against costly software errors and late deliveries. But it is much more than a bunch of automation tools – embracing a DevSecOps strategy requires a cultural shift that aligns cross-functional enterprise teams on a singular, goal-oriented path to successful software delivery.



Sure, here's a code for a simple
data analytics dashboard:

code

```
// Define data sources
const dataSources = [
 { name: 'Sales', url: 'https://example.com/sales.json' },
 { name: 'Expenses', url: 'https://example.com/expenses.json' },
 { name: 'Profit', url: 'https://e
```

Type your massage......

About the Author

Kannan Narayanan is a Salesforce MVP and a seasoned Salesforce Practitioner with 50+ Salesforce credentials and 10+ years of association with the Salesforce ecosystem. He focuses on providing advisory architectural services, de-risking solution delivery by bringing in design and architecture best practices along with focusing on ensuring maximum efficiency in delivery automation via tools and accelerators. Kannan has also delivered 5 AppExchange listings, right from ideation to solution development. Kannan is also a member for Salesforce Partner Advisory Boards (Platform & AI+Data+CRM).

At Infosys, Kannan heads the CoE - Architecture Practice and Technology Consulting Group. Kannan is a mentor volunteer at the NASSCOM Industry Mentoring Program **(https://nasscom.in/)** in cloud computing and Trailblazer Community Leader, for Chennai Architect Group.

LinkedIn-> https://www.linkedin.com/in/kannan-narayanan-architect/

For more information, contact askus@infosys.com

Infosys®
Navigate your next

Infosys.com | NYSE: INFY

Stay Connected