



# ***The Role of DevOps in the Transformation of Service Delivery***

---

Kristof Kloeckner, Giovanni Lanfranchi,  
Gopal Pingali

---

# 01 DevOps and Design Thinking as an Accelerated Feedback Loop

---

## Business depends on IT performance.

DevOps extends lean and agile practices by focusing on the end-to-end software and service delivery process, from client requirements to development and operations, and by establishing a continuous delivery feedback loop.

DevOps was first practiced in “born-on-the-cloud” organizations, leading to the integration and even merging of development and operations teams. In recent years, it has been adopted successfully in traditional enterprises and hybrid cloud environments as well.

In 2014, IBM defined DevOps as an “enterprise capability for continuous software delivery that enables organizations to seize market opportunities and reduce time-to-customer feedback.”

DevOps practices are designed to dramatically improve both efficiency and effectiveness of the delivery process to increase speed and lower cost and risk. Companies such as Nationwide have seen an 80% reduction of critical defects, 90% on-time delivery of projects, 70% increase of system availability, and productivity improvements that put 82% of their agile teams in the top two quartiles of the industry.

For efficient delivery, DevOps draws heavily on the lean concept of a continuous delivery pipeline. This requires removal of waste and bottlenecks at every step of the process to achieve faster throughput. Process improvement methods such as Value Stream Mapping are an integral part of the DevOps toolkit, as are elimination of hand-offs, automation of tasks, and reduction of work in progress.

For effective steering of the delivery process, DevOps draws on the design thinking and agile concept of continuous feedback loops, with honest measurements and a focus on quality and operational stability. The agile principle of “working software” at every iteration is extended to “working software in a production (or production-like) environment.”

Moving test, and especially integration test, to the front of the development process helps reduce risks and leads to the concept of continuous integration and continuous testing. A high degree of test automation is essential for successful DevOps implementations, in particular for maintaining quality with significant reduction of delivery cycle times.

Measurements are key to DevOps for ensuring continuous improvement. Applying analytics and cognitive technologies allows fine-tuning through better insights into success patterns. We expect *IBM Services Platform with Watson* to have a significant impact on our DevOps execution, especially combined with extensive use of Application Performance Monitoring (APM) and sophisticated event management and monitoring.

DevOps has implications on organization, culture, process, architecture, and tools. In 2014, an IBM-sponsored study by Forrester identified the most important factors that drive rapid delivery cycles satisfying business stakeholders, based on a survey of 600 IT professionals.

---

**One team, one project at a time (minimize context switches)**

---

**Small batches flowing through the software delivery lifecycle (an important lean principle)**

---

**An application architecture based on loose coupling and APIs**

## In GTS, we are addressing all these elements in our transformation initiative.



### Team organization

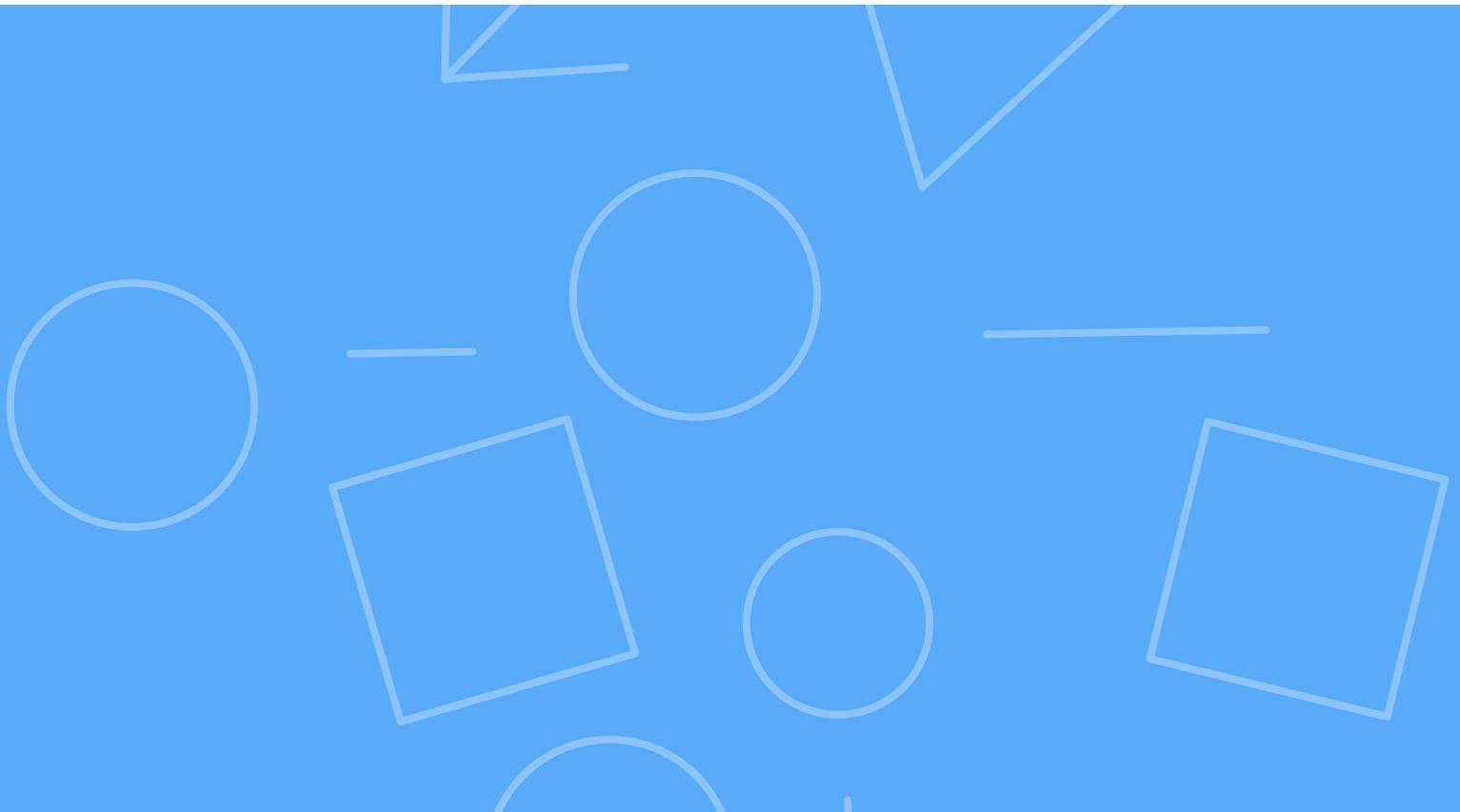
On the side of team organization, elimination of many routine tasks through process streamlining and automation allows practitioners to focus on higher-value tasks such as resolving complex incidents, requests, or exceptions. For these tasks, teams need to be structured differently to collaborate quickly and comprehensively. We are adapting the agile squads/tribes/guilds model to delivery and transforming our Client Innovation Centers along these lines. We are also applying the principles of small batches and reduction of “work in progress,” which are essential to optimizing throughput in a lean approach.



### Architecture

On the side of architecture, DevOps is clearly a complement to and supported by IT as a Service, which allows rapid deployment and flexible consumption. Only an “as a service” model, with APIs as the defining interface of a service, allows frequent updates that become instantly visible to consumers without any disruption. This is especially important for services provided by partners within a services supply chain. As the portfolio of standard offerings and solution building blocks becomes richer, composition and orchestration of existing services will become the norm, as opposed to building services from scratch.

We combine DevOps with Design Thinking to ensure thorough, deep, and systematic client engagements (such as the Cognitive Insights Garage) that we not only deliver fast, but that also provide the right solutions for a given set of client requirements and constraints. Drawing on curated knowledge and composable building blocks enables us to tailor differentiated solutions with a common architecture from standard components, with faster time to value and more intensive co-creation of unique elements.



---

# 02

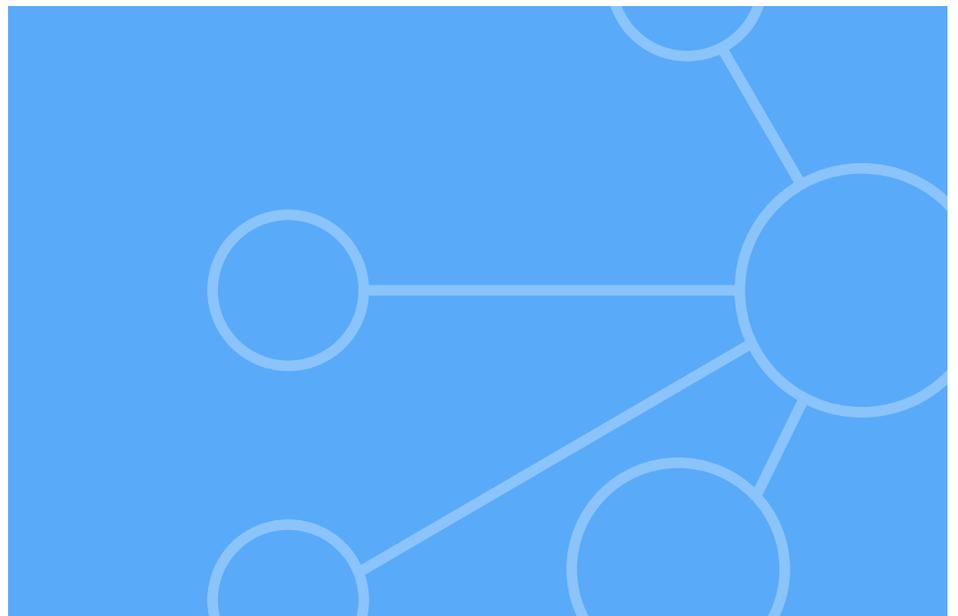
## Our Experience with DevOps in Service Delivery

---

Over the past year, we have introduced DevOps into the GTS organization, in particular into the GTS Labs, and begun to introduce DevOps into our Client Innovation Centers. We have also partnered with clients to help them transform their IT following DevOps principles and methodologies.

**The following sections describe our experiences and initial results.**

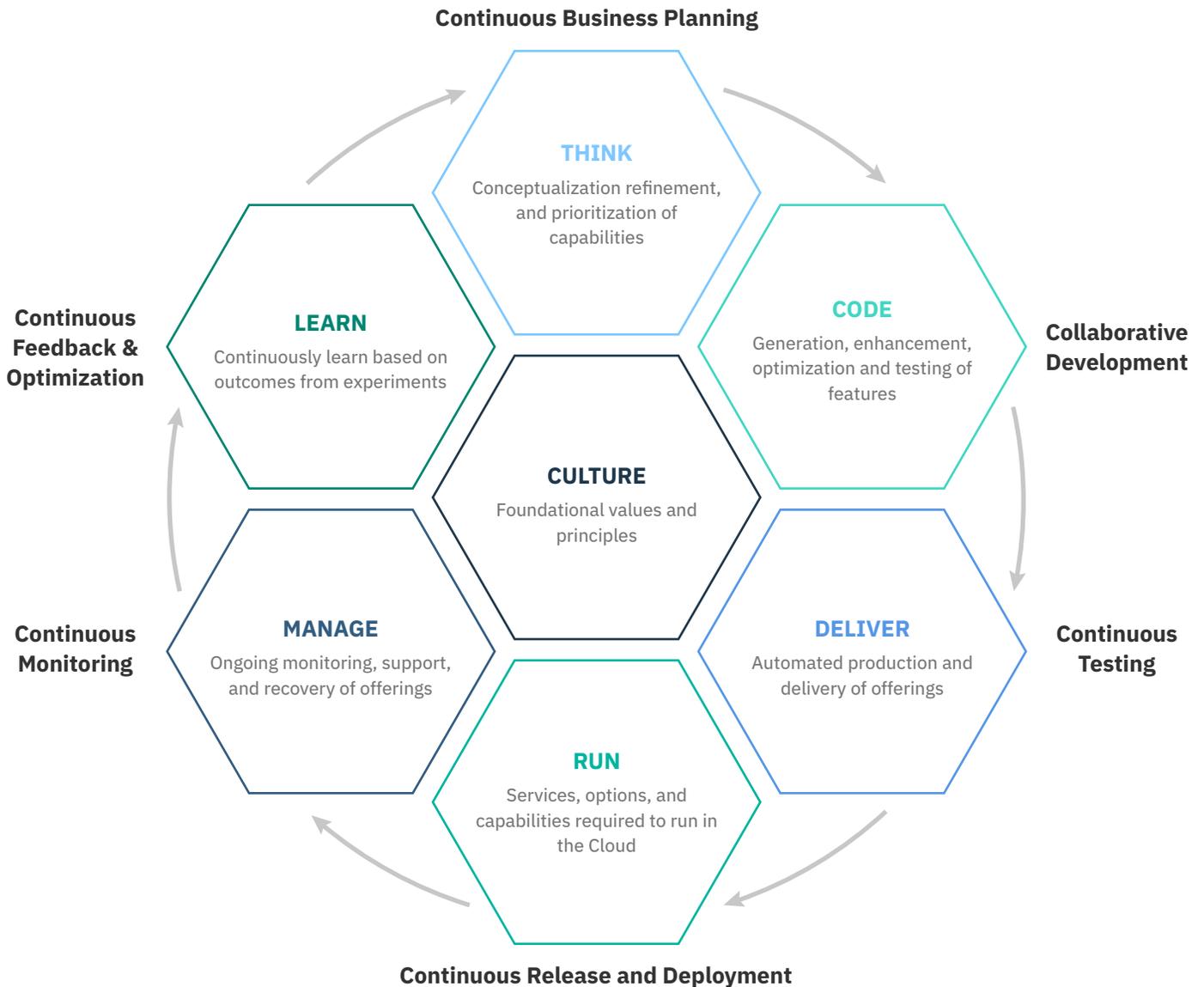
- DevOps in the GTS Labs
- Standardized Toolset as a Catalyst
- Beyond Tools to Practices and Culture
- DevOps in Client Transformation: Key Adoption Patterns and Related Outcomes



# DevOps in the GTS Labs

The GTS Labs are a worldwide team of approximately 950 members organized as agile squads practicing DevOps. The Labs consist of largely co-located teams with a center of gravity in India where we have 600 members. The mission of the GTS Labs is to provide technology leadership for the IBM GTS business with industry leading development, architecture, and agile implementation skills that realize Cognitive Hybrid Assured IT for our clients.

The GTS Labs were set up in mid-2016 with DevOps built into their very fabric. At the GTS Labs, we embrace the IBM definition of “DevOps as enterprise capability for continuous software delivery and management that enables organizations to innovate rapidly to capitalize on new market opportunities and reduce the cycle time to collect and react to customer feedback.” This includes an end-to-end cycle of practices in continuous business planning, collaborative development, continuous testing, continuous release and deployment, continuous monitoring, and continuous customer feedback and optimization as shown in the figure below.



## Standardized Toolset as a Catalyst

Best-of-breed tools across the lifecycle of software development and delivery can be a catalyst to ignite and influence the adoption of new practices. The tools we rely on to design, build, and deliver great outcomes shape our daily interactions with our team and our ability to solve problems efficiently. The DevOps toolset in the GTS Labs is based on the IBM Whitewater initiative, whose mission is to use technology to drive the adoption and mastery of modern practices at scale. IBM Whitewater provides secure and hosted enterprise versions of cutting-edge tools along with early access to emerging ones. The current set of tools in Whitewater includes Slack Enterprise Grid, Mural, InVision Enterprise, Trello Enterprise, and ZenHub Enterprise for Collaboration; GitHub Enterprise and Travis CI for Development; PagerDuty for Operational Support; Safari for Learning; and Contrast Security and Signal Sciences for Security.

GTS Labs has leveraged the IBM Whitewater hosted tools and complemented these with others such as Jenkins and Urban Code Deploy to realize integrated tool chains for continuous development, integration, release, and deployment. An example tool chain uses ZenHub for requirements management; Eclipse to create new code for a new feature; code review through a GitHub pull request; code commits in GitHub; continuous integration, automated build, and static code analysis through Jenkins; automated deployment of successful builds into dev or integration environments with Urban Code Deploy; artifacts storage with Docker Registry; UI testing of the application with Selenium orchestrated by Jenkins; and collaboration with automated notifications across dev, test, build, and deploy phases through Slack. Such an integrated DevOps tool chain has been incorporated into *IBM Services Platform with Watson* to enable the development of new services on the platform.



---

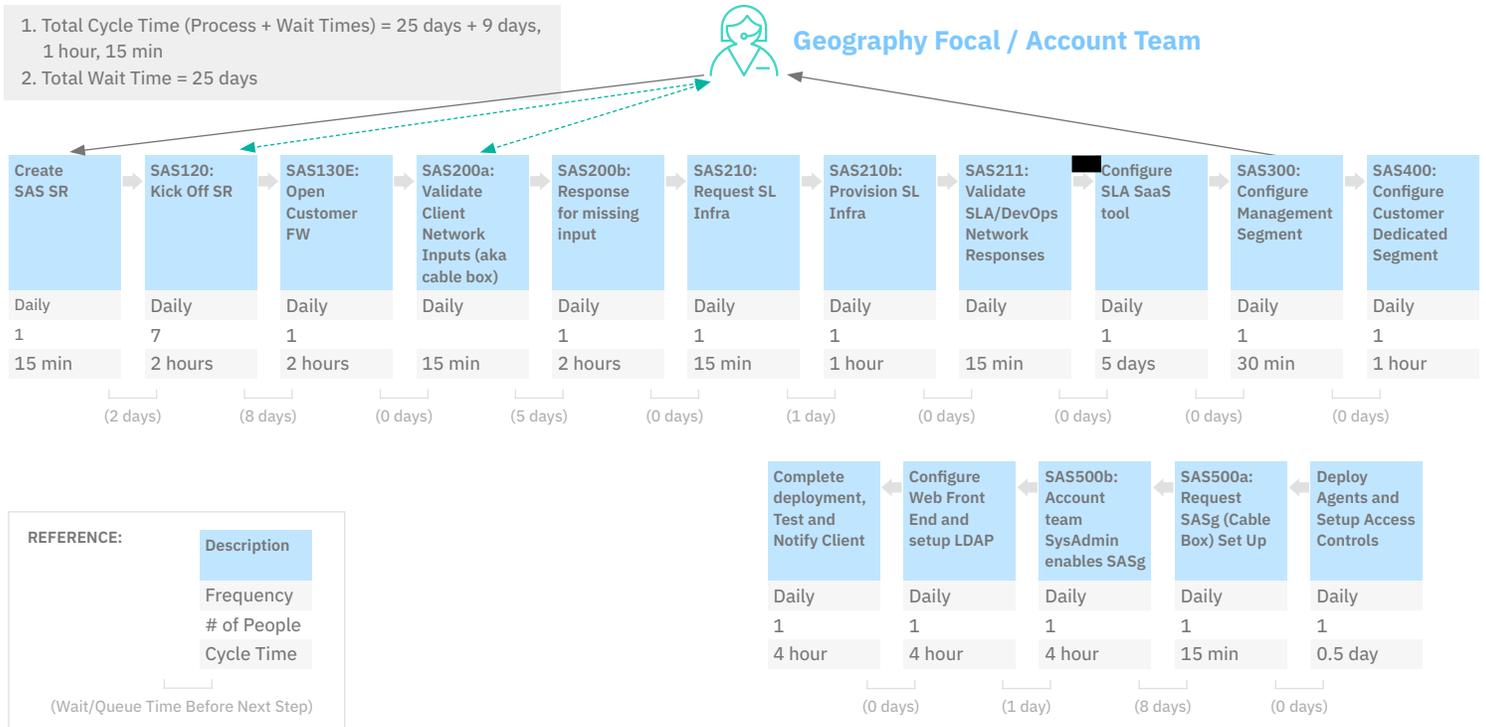
***“The tools we rely on to design, build, and deliver great outcomes shape our daily interactions with our team and our ability to solve problems efficiently.”***

---

# Beyond Tools to Practices and Culture

While DevOps involves the use of standardized tools across the lifecycle as shown above, it goes beyond just tools to the set of practices and capabilities that define a culture of constantly looking for opportunities to accelerate time to value, improved customer experience, and increased capacity to innovate. All projects or squads in the Labs are periodically evaluated for the level of maturity in each of the practices resulting in prioritized actions for improvement. Such analysis has resulted in significant improvements in development. For example, dev and test sprints were reduced from four weeks to two weeks, and time for code to production was reduced from 10 weeks to three weeks.

While these are significant improvements in speed of development of production level code, end-to-end value stream mappings reveal much greater opportunities for acceleration in time to value through a focus on deployment, onboarding, and actual use of newly released capabilities in delivery operations. The figure below shows a sample value stream mapping of the deployment of a new capability. As indicated in the figure below, there is significant waiting time going into days between several stages of operation, while the processing time is only in minutes. Insights into these bottlenecks lead to key actions for eliminating wait time such as streamlining and automation of approval processes, real-time monitoring and alerts, automated pre-validation checks, and automated assessments of the readiness of an environment.



DevOps at GTS Labs is therefore a truly end-to-end view of development and operations that takes a data-driven, automated, and user-centric approach to accelerate time to value. This approach is applied to both the development of new offerings that are offered in the market and the deployment of new capabilities to improve the efficiency of day-to-day management of IT for enterprises. For example, there were 15 releases of server lifecycle automation rolled out into delivery operations in six months, while the velocity of development was improved by 32%.

**This whole approach transforms Enterprise IT Operations to a new cognitive delivery model in which:**

- New and updated offerings are continuously delivered, much like apps on a smart phone
- IT is consumed as a service from a federated catalog
- An underlying cognitive platform collects and analyzes data to realize a delivery environment that is continuously learning and improving

# DevOps in Client Transformation: Key Adoption Patterns and Related Outcomes

Our experience in DevOps is not just limited to internal usage; it is a critical element in the way we work with our clients in their transformation journeys.

Client motivation and major entry points into DevOps are different depending on factors such as client maturity, the nature of the workload, and type of industry. However, over the past few years, we have observed some clear cluster delineation in terms of use cases and ultimate goals. Matching this, rather than adopting a “one size fits all” approach, we have progressively refined a set of well-defined “DevOps Adoption patterns.” We have already successfully leveraged this approach with many important clients and have achieved proven business value.

The first category of customers has embarked on a hybrid cloud transition, moving workloads from traditional datacenters to a combination of clouds, including leveraging Software-Defined Everything. They want to take advantage of all the important

cloud characteristics such as flexibility, cost, and speed, and they realize that this requires profound changes in their e2e development and operations processes.

The second category of clients wants to deeply leverage agile and lean methodologies (along the lines of “The Lean Startup,” by Eric Ries) in order to streamline their business processes and eliminate any waste to get to a higher level of efficiency.

Lastly, we have a third category of clients that want to strongly embrace the paradigm of “disruptive innovation” as a way to increase their footprint (especially in highly competitive markets), gain market share, and dramatically increase their understanding and intimacy with their client base. Rapid innovation, business composability, and hyper-fast feedback loops are important elements for this set of clients.

To best address these three sets of needs and motivations, GTS has developed the following “DevOps Adoption patterns”:

## DevOps Workload Transformation

In this pattern, we help the client not only to move their workload to the cloud but to (re-)define their architecture and development processes to fully take advantage of cloud in terms of speed, cost, and quality of service. Containerization of applications is an important approach in this context.

## DevOps Agile/Lean Process Transformation

In this pattern, we apply the best combination of lean and agile methodologies to dramatically improve delivery efficiency for the client; the basic principle is to limit “work in process” across the various stages.

## Cognitive Insights Garage

Here we leverage design thinking and DevOps practices to build an immersive co-creation experience with the client that is focused on rapid experimentation and prototyping with data-driven insights.

Before we provide more details for these patterns in terms of key practices and major outcomes achieved, it is important to underline that all these DevOps patterns leverage foundational kernel capabilities offered by *IBM Services Platform with Watson*.

Reusable assets and reusable content (e.g., reference architectures, integrated common stacks, workload patterns, and recipes) that feed all these DevOps patterns and enable the client to put them into action.

Knowledge bases that appropriately inform, assist, and provide insight to the client at every stage in the DevOps journey.

Standardized offerings with Assured Hybrid Implementations (certified and pre-integrated cloud stacks) that allow true composability in the definition of the foundation for each DevOps scenario.

These capabilities provide a framework for rapid development and deployment, as well as reusable assets that clients can use to build differentiated solutions.

---

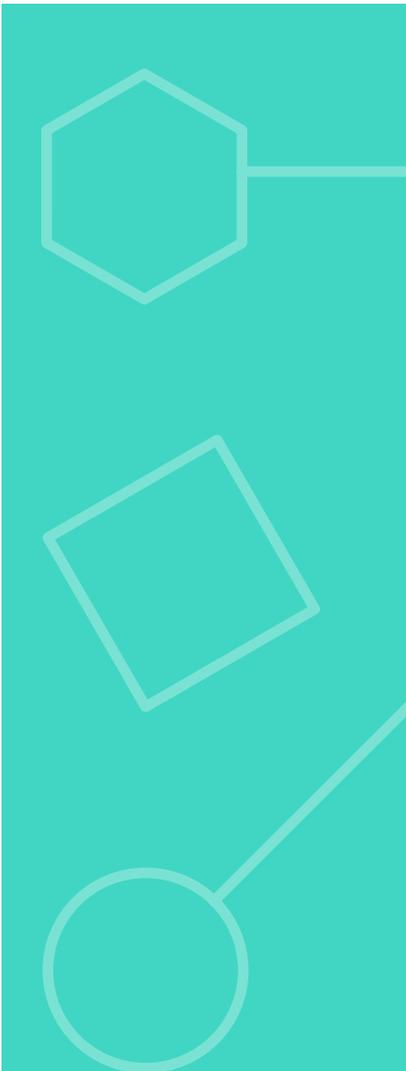
# 03

## Highlights of our DevOps Adoption Patterns

---

In this section, we want to provide the major highlights concerning our three DevOps Adoption Patterns, which have helped GTS to substantially transform the client relationship by infusing key elements in terms of flexibility, speed, continuous feedback loop, and client co-creation.

- DevOps Workload Transformation
- DevOps Agile/Lean process transformation
- Cognitive Insights Garage



# DevOps Workload Transformation

Our involvement is often initiated by the CIO who wants to provide to his lines of business a more flexible and modern environment in support of critical workloads.

As the name of this pattern implies, this DevOps Workload Transformation is mainly workload driven, and we are leveraging it to cover several important use cases:

- Migration and transformation of an existing workload to a cloud environment (aka cloud migration)
- Adaptation of the workload to take advantage of modern technology such as SDE, orchestration, and containerization
- Redefinition of the entire e2e application development chain

It is also important to underline that we are closely working with IBM Global Business Services (GBS) to define this pattern. (In fact, it starts in many cases with an “application down” client value proposition.) The ability to articulate ONE common integrated IBM DevOps approach to the client is often a winning factor and a key differentiator.

In terms of key metrics and outcomes enabled by the DevOps Workload Transformation pattern, we may consider two classes of business benefits:

---

## Greater efficiencies in managing the workload with significant decrease in cost

- Improvement of around 70% in the provisioning of workloads through orchestration technology powered by a DevOps pattern content factory
- Migration cost reduced by a factor of 3
- Dramatic increase of flexibility through SDE adoption with cost reduction of about one order of magnitude with respect to traditional environments

---

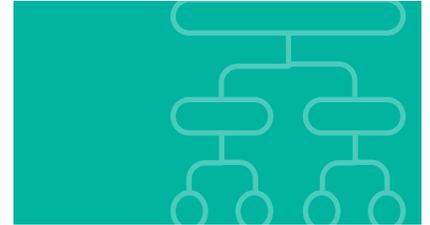
## Velocity improvement in time to market and remarkable increase in quality and SLA characteristics

- Shorten development cycle from months to weeks with continuous feedback loop
- Code quality and overall stability improved by at least a factor of 2



# DevOps Workload Transformation

An interesting example here is the DevOps transformation we have performed, in close synch with IBM Global Business Services (GBS), for one of the largest transportation companies in Europe concerning its major ticketing multichannel application. (This is the largest transportation app in Europe with more than 20 million tickets per month supporting the entire ticketing sales process for high speed trains and all regional and local transport.)



According to this pattern, the following key DevOps principles have been implemented:



Entire (re-)definition of the e2e DevOps chain at a major European transportation company; formation of squad/tribes with co-located members from IBM and the client



Agile process to manage requirements and redefinition of the company's e2e application development chain



Continuous build and continuous deployment (leveraging IBM UrbanCode)



Continuous monitoring of system KPIs for proactive problem identification



## This approach has led to significant measurable results:

- Increased speed: 55% decrease in build deployment times
- Improvement in “nonfunctional requirements”: Significant improvement in performance and stress; no production incidents over the past nine months
- Improved quality: Improved defect signature with 80% fewer severity 1 and severity 2 defects; 110% decrease in defect density
- Time to market: Increased velocity (+44%) in fulfilling new (and unexpected) sets of requirements coming from business lines

# DevOps Agile/Lean process transformation

We have also successfully applied DevOps to help our clients dramatically improve their key operational processes, including incident management, compliance management, availability management, and risk management.

In this context, leveraging the Agile@GTS community, our agile coaches, and our more than 800 agile champions, we have developed a set of DevOps methods that have proven to be successful at many accounts we have partnered with in the last 12-18 months. (We have applied this transformation pattern to more than 50 key accounts in 2016.)

In general, we have observed double-digit efficiency improvements (in the range of 50% to 80%) for these processes across all the accounts we have engaged with using our DevOps Agile/Lean Transformation pattern. The aggregated “account improvement agility index,” defined as the average percentage increase in operational lead time improvement, has increased 40% since January 2017 and 254% since January 2016.

In this pattern, we have combined the best of lean and agile methods with their key practices in terms of Scrum and Kanban; more specifically, we have applied the “Keep the work in process under predefined thresholds” principle from the “Theory of Constraints” to identify current bottlenecks and then manage flow for maximum output.

It is important to note that we have applied this DevOps Agile/Lean Process Transformation pattern not only to key accounts, but also to our Delivery/Client Innovation Centers (CICs), which comprise more than 20,000 practitioners worldwide.

We piloted the agile (DevOps) transformation in our Costa Rica CIC, and completed this work at the end of 2016. We are now in the final completion stages for Argentina, Brazil, and the Czech Republic, with a roadmap to complete the remaining CICs in 2017.

In addition to the agile/lean practices already mentioned above, other elements have also made possible the successful massive scale-out of this DevOps pattern:



## Reorganization focus

We have not dictated a “one size fits all” organizational model; instead we have aligned teams to process value streams, and we have improved their autonomy and self-consistency.



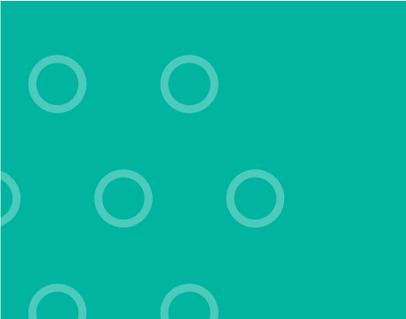
## Education and community focus

We have created a vibrant community of practitioners supporting this CIC DevOps transformation, and we have taken a diligent metrics-driven approach to measuring team progress and maturity to rapidly course-correct when needed.

**The results we are experiencing concerning the CIC transformation are quite impressive. For Costa Rica, we have measured the following improvements:**

- Service Desk: 66% improvement in ticket resolution; 19% improvement in call handle time
- Integrated Services (Distributed): 65% reduction of server build time for Windows Virtual; 57% reduction of patching tasks execution time
- Identity & Access Management: 23% productivity increase as a whole team from elimination of duplicate requests
- MMS – Platform Management: 125% productivity increase; 39% time reduction in execution of tasks

# Cognitive Insights Garage



It is becoming more and more evident that during the discussions related to the definition of a new opportunity or to the expansion of an existing engagement, clients require a new and completely different type of interaction with their service provider. Traditional one-way vendor presentations no longer work. In transformational engagement, these presentations must be augmented or even replaced by a far richer and more dynamic set of interactions focused on hands-on experience, prototyping, and ultimately client solution co-creation.

To address these critical needs, we have developed a Cognitive Insights Garage pattern that we have successfully deployed into our Solution Hubs and our GTS Labs.

## This Cognitive Insights Garage pattern is a synergistic fusion across three elements:



### Design thinking methodology

Here we work with the client to better understand the problem space, the key personas involved, the major scenarios (“hills”), the key use cases the client wants to realize, as well as other critical design thinking artifacts such as persona’s goals, empathy maps, and UX mockups.



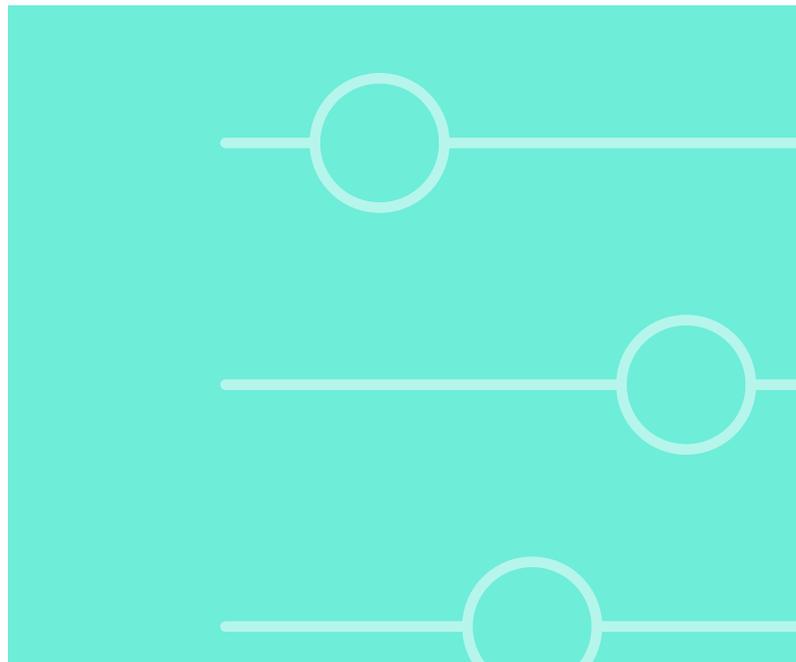
### Agile techniques

In the prototyping phase that is performed in direct touch with clients, we leverage all the digital assets we have available in our *IBM Services Platform with Watson* and apply agile principles such as scrum and rapid iterations to quickly converge toward client goals.



### DevOps methods

The goal is to include the client in our DevOps chain as a key stakeholder in our continuous delivery and continuous feedback loop. Leveraging a DevOps approach, we naturally progress toward the definition of an MDE (Minimal Delightful Experience) that provides to the client tangible evidence of what the solution would look like. This always represents an opportunity for us to better understand the client domain, use cases, and industry specific requirements; for the client, it is a unique opportunity to directly experience how easy it is to tune and compose GTS offerings on top of *IBM Services Platform with Watson* to perfectly fit client use cases.



# Cognitive Insights Garage

This Garage approach is experiencing great traction in the field with more than 20 sessions already performed with important clients across various key Industry segments, including finance, retail, energy, pharmaceutical, and government.



## The outcomes derived by this DevOps pattern can be categorized in two parts:

### Business value provided to clients during Garage sessions

- For example, for a large consumer electronic company, we have developed an MVP for the service request catalog (based on Cloud Broker) that enables digital/interactive submission of a request for an IT offering/service with the right support, compliance, cost, and quote within 30 minutes versus 10 days.
- In the case of a large government institution, we have prototyped a Cognitive Migration Advisor that advises the client on the best options for consolidating workloads from distributed to mainframe.

### Augmented buying propensity and enhanced client partnership

- In the case of a large client in the electronic industry, this has led to a significant expansion of the initial RFP (from around 40 servers to be migrated into hybrid cloud to more than 600).



We are currently working to further expand the client co-creation aspect of our Garage approach by including new assets such as the Cognitive Solutioning Advisor, which applies our cognitive models to client requirements documents and greatly facilitates tradeoff analysis and the exploration of various solution options.



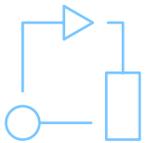
# Moving Forward

In this paper, we have described the central role of DevOps in our service delivery transformation as a set of methodologies and practices that complement IT as a Service and cognitive delivery.

We have shown examples of quality and velocity improvement for service delivery, as well as impact on client transformations. We strongly believe that the increasing adoption of DevOps

within GTS will substantially transform the relationship with our clients. We will extend it to cover the entire lifecycle, all our delivery centers, and key clients, and we expect additional differentiation through increased automation and refined cognitive feedback loops.

## Finally, we want to highlight three key strategic dimensions for our DevOps fabric moving forward:



### Ecosystems and third parties enabled

through published APIs; strong brokerage and orchestration services to enable rapid onboarding of partner services



### Powered by Cognitive

(through our platform) with “cognitive agents” at all levels supporting continuous feedback loops with increasingly autonomic actions; continuous improvement and client co-creation leading to higher efficiency and effectiveness



### Community-based

with GTS, client, and partner communities becoming the stewards of the continual refinement of our DevOps practices and the curators of our accumulated knowledge

