Deep Science Publishing
https://doi.org/10.70593/978-93-49910-74-4_4

DeepScience
Open Access Books

Chapter 4: DevOps integration in modern insurance platforms

4.1. Introduction to DevOps in Insurance

Traditionally, the practice of insurance has leveraged changes in technology to better serve its customers, increase efficiency and controls, and recognize the value of technology-led change as a value driver. Overall, these factors have significantly improved the service offering and company profitability with insurance reducing the loss ratios over the past decades. In fact, as a result of these investments, and given the new technologies available insurers and reinsurers operate in a far more competitive environment. That said, the driver of value is changing, as reflected by the heightened interest in the reinsurance and insurance tech sector (Fitó et al., 2010; Bass et al., 2015; Erich et al., 2017).

Over the past couple of decades, the rate of technological change has accelerated dramatically with cloud, machine learning, natural language processing, use of APIs, and software development tools allowing companies to adapt even faster than previously. With this has come a much greater power that is democratized more broadly with new tools allowing small startups with limited capital but massive capabilities to challenge the incumbents. This pace of change is set to increase further with the advent of AI-driven products. Given its potential impact on multiple dimensions of our society, regulatory scrutiny will surely increase increasing costs and complexity. To combat the potential loss of market share or even the risk of disruption, insurers have shown interest in digital transformation, to not only safeguard their position but increase gains, efficiencies, and value creation. Insurers have looked to invest in in-house talent and work with insurtechs to experiment with the new technology to identify new opportunities but also deputizing tasks to digitized processes and technology stacks.

This churn in the economics of these industries is driven by several factors including the importance of data and technology in the identification, and support of risk as well as

the use of new technology in pricing as well as the support of risk portfolios (Gruhn & Schäfer, 2015; Wiedemann et al., 2020).

4.1.1. The Role of DevOps in Transforming Insurance Practices

In most industries like Insurance which are not considered tech-driven, there are still a lot of delays, bloating resources, spending, and time developing new features or capabilities implementation for the organizations due to the involvement of age-old and outdated systems and procedures. DevOps principles can help such organizations to streamline and improve their processes by focusing on developing new features and reducing the period for launching a new functionality for the customers. With businesses required to continuously adapt and respond to the environment and customer needs, we will highlight the role of DevOps in transforming these insurance practices. Business, and in our case the insurance sector's business does not function in isolation. With the changing environment, customers and their needs, and preferences may change. This might result in a risk and control challenge for businesses.

Fig 4 . 1: DevOps in Modern Software Development

Hence, the insurance sector must be able to respond to these changing customer demands and offer new capabilities and solutions within a reasonable time frame. The use of new

technology provides opportunities for insurers to develop innovative products and services that allow them to quickly pivot their offerings as well as support operational efficiencies in back-office systems. This, however, is challenged by insurance organizations still running on legacy systems and processes. With the tech space and especially Financial Service companies venturing towards DevOps processes, value chains from tech delivery to business goals are closely linked which will help not only in implementing a new functionality but also reducing the implementation time. In today's market, customers seek innovative, personalized, simple, and intuitive products that offer protection as well as solutions.

4.2. The Evolution of Insurance Platforms

The traditions and conventions that have governed the insurance business over centuries cannot readily be set aside. Consumer demands, the emergence of digital-native competitors, and the legislative and regulatory landscape are transforming insurance. Customers are using their ever-expanding range of digital devices to buy insurance – most often through a mobile app – then manage the policies and make the claims. Local competitors who are tattooed into the local culture, norms, and preferences for insurance are no longer monopolies. These local companies face huge competition from large, sophisticated companies with national and global scale who can use econometric models and the vast supply of algorithms, data technology, and even AI, to solve a particular insurance need at the best price. The local companies are disrupted by the new entrants who leverage their tech advantage to sell underpriced products. Legislation and regulation, which govern the insurance industry and put limits on practice, are, in many cases, outdated and in need of review and reform.

All these forces are causing a seismic and transformational change in the insurance industry. Furthermore, this transformational change is occurring at breakneck speed, and companies that do not radically change fast enough will eventually become dinosaurs. However, such timeshares may also become apocalyptic; such digital transformation will enable companies to surge ahead, virtuous monopolists in a new economy, slaying their competitors and edging them out of the industry. Key to the digital transformation and the shift away from traditional insurance practices is revamping the platform, the integrated model that provides functionality for every aspect of the insurance business. Such platforms remove the challenges of duplicated efforts and conflicting interventions from progressive changes to systems that have become more heterogeneous through decentralized technology choices and partnerships.

4.2.1. Transformative Impact of DevOps on Insurance Systems

With many assets in common, insurers are recognized historical users and developers of data technology platforms. Whether for risk description, assessment of the realized risk, or the management of the contract followed and operated over time. Using technology to evolve internal operational systems, and market-facing customer service systems, has been underway for decades. More recently, these systems have been moving towards a cloud-oriented architecture, and their institution development has migrated to a service owner model whereby stakeholders and users participate in the design and operation of the systems they live with. Part or all of the operations now sit outside of central information technology. This shift has already presented challenges in managing risk ensuring internal systems and practices have adequate governance and protection from potentially damaging events and hazards, and exposing themselves to operational risk through systems evaluation missteps. Insurance industry management is recognized for its attentiveness to the area of operational and systems risk from these new approaches. Insurance provides a tool to fund the sharing of these risks among the community. The DevOps approach has included a tightly coupled, highly integrated build, testing, and release approach for all changes into the shared version. Expanding the management of systems development beyond the central services creates challenges for the organization and enforceable control, but also shares the benefits of dedicated implementation resources with a deep affinity to the business process and local system.

4.3. Key Principles of DevOps

Digital transformation requires modern businesses not only to understand the implications of new technologies but also to direct and build them into new competitive business models. As technology convergence occurs, customers increasingly expect insurance carriers, agents and brokers, and their partners and suppliers to deliver new services as an integrated experience — enabling them to easily navigate complex evolutions, such as a global pandemic, a human loss, distress, any kind of disaster, or become business partners who can be trusted. In response, insurers are applying new digital technologies, including big data analytics, DevOps, cloud, artificial intelligence, insurtech services, and IoT.

For insurers to instill this digital mindset, they must center their technology strategies on innovation speed. An insurance platform presents UI/UX, analytics and quoting, processing controls, business; and a product factory that enables continuous value delivery. There are several key principles directly fostering innovation speed three key initiatives: Modernization, the Virtuous race for Cyber-resiliency, and Synchronizing Development and Operations. Building what the business needs while adopting and continually adapting the best operational model that generates the greatest competitive

business advantage is how innovation speed is accelerated. Do it because you're getting real tangible value. Do it because it's accelerating your business.

Companies that have implemented a DevOps strategy have reported that projects can be delivered faster and more securely, allowing for higher customer satisfaction. These results stem from the principles of fostering collaboration between operations and development, automating certain processes, and testing early and continuously. DevOps automates the manual phases of the software development pipeline, optimizes the remaining steps, and acts as an integrative unit that aligns the priorities and quality guarantees of all those involved. Among companies with a high degree of DevOps adoption, a significant percentage found that their products are governed by efficient and effective compliance, while a majority said that they could deliver secure software and fast.

4.3.1. Core Concepts of DevOps in Insurance

Insurance platforms are typically data-intensive. They usually include multiple partner and product connection points. Core services include: data management, business process orchestration, address and identity geo-verification services, security, operational monitoring, notifications, audit trails, and fraud detection. Development, enhancement, quality checking, and maintenance of modern insurance platforms cannot be done in isolation anymore from business needs and business activities. Platforms form the backbone whereby business activities are performed. Furthermore, many insurance activities that were once done by humans are now being transitioned to services. For these reasons, DevOps must focus on insurance business processes alongside technology artifacts.

The DevOps mode imposes a risk that management reviews such as architecture, security, performance, resilience, and compliance are neglected. In the early days of DevOps, this challenge was referred to as "pushing faster." Practices such as architecture refactoring and additional compliance checks were designed to mitigate the risk. However, metrics that compare dev and work across releases between dev and ops are more generally recommended. Better together. DevOps teams can effectuate insurance transformational initiatives quicker because of close collaboration with lines of business. Insurers cannot neglect change management techniques. Several core concepts of DevOps in the insurance context are as follows.

4.4. Benefits of DevOps in Insurance

1. Faster Time to Market

The need for DevOps in insurance arises due to structural and operational limitations preventing organizations from moving at a higher velocity. To overcome these challenges, DevOps does not simply add a set of tools and resources. It influences the firm through process, technology, and culture changes that create the right environment for rapid innovation while ensuring the operation remains resilient to software outages. Today, core policy and claims management systems in insurance often run on decades-old code structures that must be continuously re-engineered. Due to long refresh cycles, firms struggle to address increasing demands for new functions. The inability to rapidly create new sub-systems and workflows around existing functions closes off opportunities to better align and personalize service to meet customer expectations. The high cost of operational maintenance does not allow insurers to commit sufficient resources towards building new application and solutions capabilities and underestimates the upside benefits from expansion in use cases and functions.

The need for a speedier time to market is especially critical today at a time when insurance firms are facing intense pressure to not only grow but also diversify – both in terms of the segments they serve and the opportunities they provide. This means creating new products in areas like telematics or embedded products or expanding variants of existing solutions like parametric coverages for travel or re-insurance. A characteristic of the soft market currently affecting both the property-casualty and life sectors is that many firms are close to their combined ratio limits. In difficult circumstances like these where increasing premium prices does not guarantee sustainable profitability, firm-centered technology solutions with a low cost of maintenance and a high ROI on functionality refresh are essential.

2. Improved Collaboration

With DevOps, development, and operations work in step and collaborate closely on desired functions and processes and any required modifications for tuning and performance assessment based on observed data. A virtuous circle of continuous design, code, deploy, run and feedback emerges that role backlogs out of both departments. Internal and external customer experience is optimized as flows and even minute tweaks of functionality and sequence are tuned for performance. The long refresh cycles of the past – of weeks and months during which released functions stayed idle – are virtually eliminated. There are effective build codes and staging environments available for multiple code versions. Communication on what is working well and not so well and the relative priority of releasing each part for production is open and transparent. Not surprisingly, new tools are emerging to fast-track DevOps enablement in other industries. The global cloud hosting majors are re-packaging many infrastructure

components as pre-configured microservices to facilitate the ease and speed of development, testing, and deployment. Many of these services, especially around integration, are being actively customized for insurance.

4.4.1. Faster Time to Market

Insurance companies operate in high-stakes, highly competitive, sharply evolving environments where smart technological solutions are central to devising business models that make sense. From new insurance products to new distribution channels, insurance initiatives have multiplied in the interest of improving the business's bottom line. Traditionally, insurance has been about high-complexity, long-cycle, risk-averse business. However, rapid technological advancement, evolving consumer preferences, receding product life cycles, an ever-changing regulatory environment, and pressures on cost and performance competency drive the need for insurance companies to act differently and to think differently—engaging in rapid transformation, innovation, and rejuvenation. Insurers need to speed up their journey toward disruptive innovation by reconfiguring their creativity and research functions, optimizing their DevOps, and adopting modern approaches to technology implementation that leapfrog conventional deployment practices. In addition, the associated shifts in customer and distribution behaviors, the entrance of new players, and the potential effect of globalization also require continual investments in advancing and upgrading the operating model underlying the core insurance processes—actuarial, underwriting, claim settlement, premium collection, product and service development, marketing and sales—along with the technology driving them.

Insurers need to shorten the time to market for their transformational initiatives in a DevOps-as-a-service model. Insurers also seek to leverage new technologies in optimizing the product and technology-development cycles to achieve better results in shorter time frames—realistically compressing the development cycle to as little as 30 days, with shipping out of beta models to production within 60–90 days of product identification. During the product development process—and its service lifecycle thereafter—the utilization of social media and mobile apps is becoming essential in communicating, capturing, and analyzing data related to customer experiences, usage issues, and corrective action during the product development process.

4.4.2. Improved Collaboration

While enterprise silos were initially used to enhance security, the digital transformation of the market has made such components difficult to defend against cyber criminals. New threats require insurance companies to collaborate more efficiently and facilitate

new technologies. DevOps reduces the friction between teams, motivating insurance professionals to share knowledge with development-focused employees. When all sides have access to innovative, new technologies, employees become excited about the insurance industry's potential, rather than focusing solely on compliance. For the enterprise, collaboration creates a unified, singular vision. When market strategy, design thinking, and analytics development teams share a common goal, they become efficient at pursuing that goal. DevOps integrates rules and algorithms into insurance systems by enabling faster feedback from stakeholders. All parties share the focus of identifying warning signs and areas of opportunity by developing a set of dashboards focused on core metrics. Digital products include scheme design to implementation release, undergoing four development steps with technical and domain experts included from the onset. Internal components of the underwriting system and workflow are also being moved out of individual teams and published in a collaborative content store. Becoming a collaborative company does not come without its challenges. Employees expect these products to be innovative, yet simple. Developers want to ensure the product remains of high quality, reducing the chances of market failure. However, insurance companies recognize that portfolio diversity becomes increasingly difficult to manage through enterprise-level controls. Additionally, the difference in deadlines between an innovative digital product and its insurance peers creates challenges with performance evaluations. However, if you enhance expectations with career growth, employees will collaborate and work toward a common goal.

Fig 4 . 2 : DevOps Market Share

4.4.3. Enhanced Quality Assurance

In insurance, software quality is paramount. The release of defective software may compromise the ability of insurers to process new business and service existing policies, with repercussions to business continuity, loss of revenue, and most importantly, regulatory compliance. As such, traditional manual QA and testing is an incredibly laborious, time-consuming, and error-prone endeavor, with compromises made in terms of timing or coverage. Insurers cannot introduce a delay in processing an insurance application that is ready for underwriting, but at the same time, they cannot afford to let an application flow through a broken platform where core functions are inadvertently broken. There is also the challenge of continual investments in legacy systems where refactoring and improving coverage through unit testing may not be the ultimate goal, but are necessary to usher in new capabilities that allow for the application of state-of-the-art artificial intelligence systems for risk classification. To address these challenges, insurers look to QA automation.

Insurers are using automation tools and companies are adopting DevOps to spend less time on error-prone routine tests of software that has previously been tested extensively and gain greater available time to assure higher-value quality initiatives and initiatives with a greater impact on insurance policy processing. For example, it makes sense to use automated testing tools for the simplest and most routine accounting checks, while maintaining a manual testing procedure on more complex business logic in the underwriting and claim processing areas that require nuanced management judgment expertise along with an understanding of the company’s risk appetite. In summary, to ensure high-quality insurance systems, insurers are moving away from monolithic manual testing, and increasingly using QA automation tools to streamline routine testing of areas of policy processing, freeing up in-house and outsourced testing resources to focus on more meaningful testing efforts.

4.5. DevOps Tools and Technologies

Modern DevOps practices in insurance platforms increasingly leverage several tools to enable Continuous Integration and Deployment automation stages incorporated in the Agile Software Development Life Cycle. The evolution of Continuous Integration tools began with the earliest releases of Unix-made utilities and grew with more integrated frameworks. These tools enable automating the build and acceptance testing process usually hosted on cloud-based infrastructures and triggered by repository webhooks.

Containerization technologies, such as Docker and environments orchestration solutions allow insurance providers' IT teams to deliver microservices at the desired release frequency while guaranteeing no unexpected side effects will happen to production

services' availability and performance. When running a production cluster, the tool provides DevOps Engineers access to observability, monitoring, and logging using a toolset at the cluster, node, and workload levels. By using this toolset for collecting high-definition observability metrics, it allows root cause analysis for production incidents and service-level framework reports and enables the creation of service-level indicators and objectives for monitoring work services' operational performance. Therefore, the adoption and avail of the DevOps toolchain are especially important when implementing a Monitoring as Code framework. In this framework, all monitoring setup, configuration, and code integration are managed as code artifacts. Having them versioned using a strategy enhances the reliability and low-friction upgrade and maintenance as part of the cross-platform microservice-to-cloud hybrid integration DevOps integration strategy.

4.5.1. Continuous Integration Tools

Modern CI/CD tools that support the DevOps pipeline include configuration management that automates the deployment of servers or applications by integrating different scripting languages and version control systems with application packaging or orchestration. Such tools help application developers provision the server and deploy the code on different environments using a wide range of plugins. Plugins are created by organizational policies for applying security scanning tools in the entire CI/CD pipeline. They also implement different scanning tools for code quality and for enforcing coding standards for different programming languages before deploying the code to various test environments. These CI/CD tools trigger different quality checks from the last commit till deploying the code to production during the deployment process to enforce DevOps principles and streamline the development cycle path to minimize the feedback loop using automated feedback processes. Different branches for development, quality assurance, and production are created and automatically tested for code quality and integration by CI/CD tools, which, if successful, allow the code to be merged with production. Such tools help application developers provision the server and deploy the code on different environments using a wide range of plugins. Plugins are developed by organizational policies for applying various static security scanning tools at different stages in the entire CI/CD pipeline. They also implement different scanning tools for code quality and for enforcing coding standards for various programming languages before deploying the code to different test environments. There are different types of CI/CD tools available based on the licensing model. Examples of these CI/CD tools include open-source tools that use an agent architecture and various plugins for building the CI/CD pipeline process, and enterprise tools that do not require any plugin for the CI/CD pipeline process. Popular cloud tools require the application developers to define the CI/CD pipeline as code in the respective SCM for updating it, and any change related to the CI/CD process by the user requires a new branch check-in for validation.

4.5.2. Containerization Technologies

Though adopting a microservice architecture increases the complexity of applications, it also offers great opportunities for modernization. Modern applications provide digital-first insurance services, fast time to market, and endless integration possibilities using APIs. Microservice architecture is commonly supported by containerization technologies. Microservices are deployed in modular containers that can run multiple instances based on infrastructure demands, can be hosted on-premise or in the cloud, and are started and stopped by automated orchestration tools. Lifetime management and discovery of the deployed services can be automated using orchestration tools.

Since 2008, when the open source project entered into the Linux kernel, the container life cycle on Linux has been supported in the kernel. The initial release enabled a limited version of containers, with native file systems, networking, and control groups. Not too much longer, in March 2012, innovated the initial release of the containers project, a lightweight command-line utility for spawning and running containers. This project was later integrated into the runtime of the open-source project, which democratized container technology. Incorporated novelties like image control via Dockerfile, private repository service, a composition engine that could start and manage multi-container applications, etc. With that, opened a world for developers allowing them to treat applications as a set of self-sustained modular services instead of platforms dependent monolithic blocks.

Offered a largely innovative development model, but it also introduced a new tension between developers and IT. Developers were incentivized to use the latest technology to speed up development and prove growth, while IT was worried about the implications for security and reliability, needing to manage architecture dependencies without control. With the developers enabled by container technology, the mindsets were changed, and through faster project lifecycles, the DevOps culture and tools matured and evolved with the need for support from the container ecosystem.

4.5.3. Monitoring and Logging Solutions

All the elements of a DevOps process must be monitored to assess how it is behaving when it is working properly but also when it has faults, and when it is failing. To take the necessary actions, for instance, to correct the system behavior or to inform the stakeholders, it is necessary to collect data, store it, and then to be able to verify it through a dashboard. The general categories of monitoring and verification tools are: (1) solution monitoring, (2) solution usage statistics, and (3) solutions audit trails.

Solutions Monitoring

Monitoring tools check the behavior of a solution, such as its health and performance. The monitored items can be: system resources consumed, external system connections, application performance and behavior, and the external services driven or consumed by the application.

Some of the popular tools to monitor cloud services in a general way are: various monitoring tools. These general monitoring tools are complemented by other specific tools dedicated to specific programming language-based applications, services, and components.

Solution Usage Statistics

Usage statistics tools capture and expose aggregated information regarding how the application is used. They provide metrics such as: counts of unique and total daily users, events triggering or not various features, session duration, geographic origin of users, etc. Some of the popular usage statistics tools are: various usage statistics tools.

Solution Audit Trails

Audit trails tools gather information about what have they done, typically in log files, and are used as record keeping solutions where corrective actions can be taken. Based on these events, audits are carried out to make sure that users are complying with the expected operational procedures.

4.6. Challenges in Implementing DevOps

Although the benefits of DevOps have been widely documented, integrating this paradigm into traditional organizational structures is still not straightforward. Several obstacles often delay the adoption of DevOps in organizations. The following three obstacles are the most frequently highlighted: cultural resistance and inertia toward change, integration of legacy systems with modern procedures, and skill gaps in teams. A thorough exploration of these obstacles will allow insurance players to design suitable mechanisms to mitigate their potential impact.

Cultural Resistance

Cultural change and adaptation a well-known barriers to innovative technology adoption. This is no different when it comes to adopting a DevOps methodology. Traditionally, operations and development teams have had opposing goals. Development teams are rewarded for rapid service development and deployment, whereas operations teams face consequences for errors and downtime. This leads to friction between

development and operations, which may not collaborate and communicate cohesively. How such teams are traditionally funded leads to a major problem. Companies hire large software development teams and smaller operations teams, leading to operational teams lacking the engagement needed to properly maintain services. The disintegration of the previously existing demarcation lines between development and services ownership, where teams are rewarded and funded exclusively on their ability to deliver code in a rapid-fire environment, leads to a two-faced company culture. On one side, there is a team that does not see the human cost associated with mischaracterizing development as an isolated phase. On the other side, some teams are charged by the company for maintaining the current flimsy state of affairs of design and deployment and may be blamed when update cycles reach deeply destructive lengths.

4.6.1. Cultural Resistance

Cultural resistance arises from a mismatch in organizational structure and the streamlined integrated element. Pace of Change, rules and the size of the organization helps reduce conflict. To facilitate implementation, organizational elements and processes need to be aligned.

The adoption of DevOps practices requires changes in the principles, practices, and tools used for service development and support. It also changes the interaction between Dev and Ops teams, the way they collaborate, and the continuous feedback from production to development to validate the original assumptions and decisions and put customers and business interests first. The cultural changes that accompany DevOps implementation reduce silos for teams, improve communication across all teams involved, and heighten the emphasis on quality and performance. It ensures that security is embedded into the processes and that the service is resilient as it is being designed and implemented. These required changes are not just differences in the way of working. It does affect the performance review and recognition systems. The silos between teams are also being broken. This is not a simple deviation or glitch; this is leading to horizontal lines of reporting and accountability and a supportive shared services mindset. Some areas that require active change intervention are business ownership and focus on the solution over just delivering the project. Performance metrics also need to change from being only financially focused to having a customer and process focus. Beyond the changes in the intersection of the teams, the concept of self-service platforms and support is core to making the DevOps journey possible. There also need to be rules about providing these centralized horizontal capability functions to enable the product teams to focus on differentiation and business impact.

4.6.2. Legacy Systems Integration

Despite the many benefits of DevOps implementation, there are still challenges that any enterprise must face, like the following:

DevOps requires intense collaboration and sharing of tasks and resources. If development, QA, and operations teams have traditional silo structures, cultural resistance will impede automation. Traditional structures focus on individuals, while DevOps Policies focus on the team. Employees should understand that their rewards and recognition are based on the overall success of the team and the business, not on the number of lines of code that were tested, nor on the number of incident tickets they resolved.

Legacy systems are components of long-lived software systems. These components are often viewed in the context of their later years when their functionality is less frequently exercised, they are costly to maintain and replace, and their technology is obsolete compared with current offerings. This may be based on decisions made in earlier years of the software system's life cycle. At this point, re-engineering components become the best possible solution, decreasing costs, enabling easy change, improve performance, among others.

An established insurance company, for example, may have been using the same technology for new development, as well as over layers of legacy integrations, for over 30 years. If their structure and business are bottle-necked by legacy but still integrated, for different reasons, the inclusion of newer supporting technologies, with DevOps suddenly becomes impractical. Newer endpoints developed with DevOps-based engineering must be reviewed, tested, and released in multiple steps until integrated, all while the legacy structure is still being used.

4.6.3. Skill Gaps in Teams

The majority of the challenges surrounding the implementation of DevOps are indisputably related to the cultural impact it has on organizations. The integration of development, operations, and application support within a company can result in the creation of problems that would otherwise not be there if they were managed separately. One of the most common of these problems is the reduced responsibility that developers have once the application is implemented, as the operations and production support deal with problems that might arise afterward. It is crucial to make sure that developers understand that they are equally responsible for issues that appear at a later stage as operations and support teams. To overcome this cultural gap, one recommended method is to keep the production support engineer next to the development team as long as

possible. This gap can also be reduced by rotating engineers across teams in each of the DevOps process phases.

When a development team is transitioning from traditional practices to DevOps practices, engineers will have no one to rely on for learning. In the beginning, teams should have somebody who was part of the original DevOps transition to aid and guide teams. Furthermore, to make sure that the adoption of DevOps in the company is successful, engineers should be given the necessary time and guidance to learn. It is also the company's responsibility to provide the infrastructure with the necessary tools that help engineers ease their learning phase. QA teams will have to transition from the manual testing processes to automated test creation and execution. Here too, there will be a new learning phase, as well as possible inexperience with creating automated tests. In companies with established DevOps tools, processes, and phases, teams are more likely to accelerate the learning process faster.

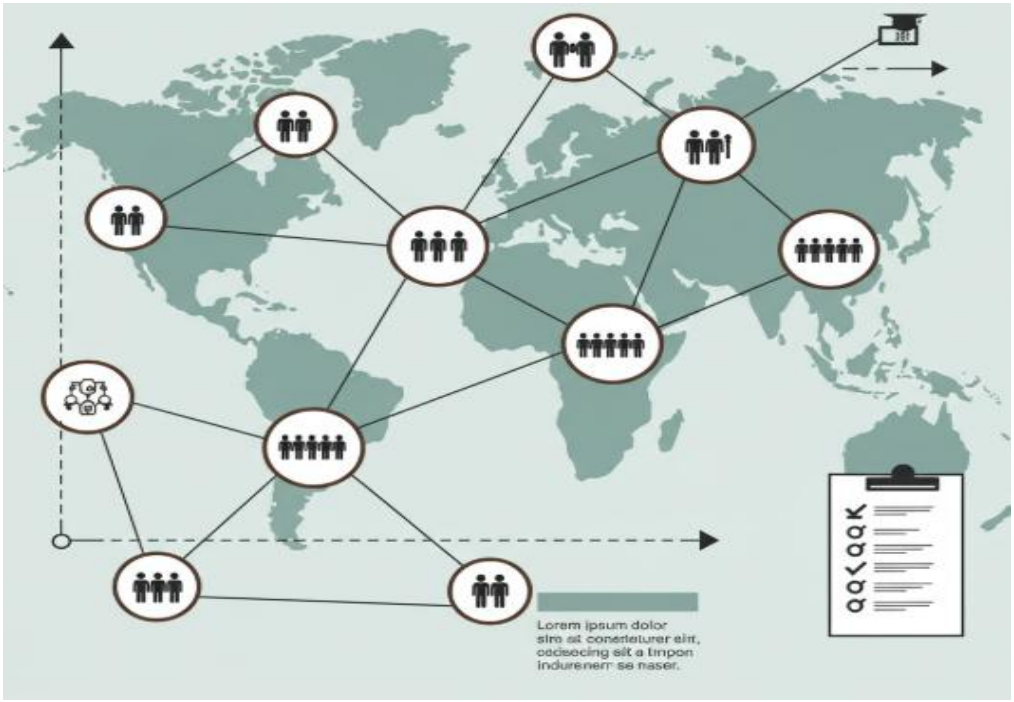

Fig 4 . 3 : Skills Gaps in Your Workforce

4.7. Conclusion

A few segments have embraced significant transformation over the recent years and the insurance area is one of them. This improvement has been set off by Information Technology, as it enhances functional efficiencies, availability, customer services, and

satisfaction, all the while cutting operational costs. The traditional technology architecture created for specific implementations is at present being supplanted by the forthcoming Gen Next tech. The progressions in technology architecture build the accessibility of a new model to meet customer demands. The emergence of Engineering, Construction, and Management provides organizations with an opportunity to build distinctive capabilities-based systems in the dispersed technology architecture. In conclusion, this paper makes a call for fresh paradigms for an intelligent Augmented Modern Enterprise, and shares implementations to demonstrate how unique strengths and capability augmented with more seamless and production-oriented operating models, some of which borrow principles from DevOps, are required for the future. The paper also shares some tactical building blocks that Technology Management and delivery groups can take on from a Scope Management, Delivery Execution, Release & Spin Up, and Finance Control perspective, wherein the recommendations to borrow from principals from Fast IT and Development Operations if implemented, can help build the required capabilities on demand, ramp up and ramp down plans in a more orchestrated manner, all the while improving quality and speed by optimizing the "getting there faster" dimensions. The IT Operating landscape of Modern Enterprises is also rapidly evolving, therefore CxO teams must assess, adopt, build, and implement the right models for their transformations around their unique context and set achievement ambitions reflecting their individual and collective Risk appetites and Growth aspirations. In closing, the Digital Transformation Imperative, and the development, operations, and delivery, Engineering, Construction, Delivery, and Management collaboration for IT and Its Systems for Modern enterprises is founded on Trust, the Development Task, the Operational Task, and the Leadership Tasks for CIO/CTO and their CxO teams, and their Organizational Development, Technology Service Management, Development, Operations, and Delivery capabilities and skills.

4.7.1. Future Outlook of DevOps in the Insurance Sector

For a long time, there has been an IT gap between the innovation and agility in the consumer internet sector and the services typically offered by enterprise services such as insurance. Given that other enterprise services have started catching up, as evidenced by the emergence and growth of new companies, the insurance sector will not be an exception in adopting technologies that are influencing an industry's evolution. The question only seems to be when and how quickly these innovations will be inserted into enterprise functions to disrupt the industry. Given the positive outcomes seen in the adoption of modern technology practices in other industries, it does not take a long stretch of imagination to look at adoption in the insurance sector as being a key part of this evolution.

We see that with the digitization of solutions across the financial industry and with the demand for more and more on-demand and personalized user experience, there is a growing need for an increased pace of innovation and work function coordination. Such functions involve Content and Media, Risk Assessment, Actuarial Function, Underwriting, Financial and Performance Management, Claim Processing, Recoveries Management, Asset Management, and Customer Servicing. These coordinated internal functions use multiple perspectives of the same pool of data and compute heavy models that need to be updated, improved, and run through model life cycles to draw insights and offer intuitive and trustworthy personalization. Imagine how highly efficient these functions could be by applying various operational capabilities, with the approach enabling such capabilities to be framed, delivered, and operated in unison. With the demand for ever-increasing personalization, organizations that do not embed such capabilities into their core will seem outdated in no time.

References

Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective. Addison-Wesley Professional. https://doi.org/10.5555/2849240

Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A Qualitative Study of DevOps Usage in Practice. Journal of Software: Evolution and Process, 29(6), e1885. https://doi.org/10.1002/smr.1885

Wiedemann, A., Krcmar, H., & Röglinger, M. (2020). The DevOps Phenomenon. Business & Information Systems Engineering, 62(1), 1–5. https://doi.org/10.1007/s12599-019-00626-3

Fitó, Ò., Guitart, J., & Macías, M. (2010). Business Support for the DevOps for Cloud Services. Future Generation Computer Systems, 26(8), 1174–1181. https://doi.org/10.1016/j.future.2010.02.010

Gruhn, V., & Schäfer, C. (2015). DevOps for Insurance IT: Enabling Agile Transformation. In: 2015 IEEE 8th International Conference on Cloud Computing, 935–940. https://doi.org/10.1109/CLOUD.2015.135