




Blockchain in DevOps

Implementing transparent
continuous delivery



To stay competitive, companies have to keep up with the pace of business automation and technology delivery transformations in the industry, while continuing to meet control requirements throughout the development and production lifecycle.

Introduction

Manage SoD compliance using blockchain

To increase business agility, large companies are adopting Bimodal DevOps processes and, almost uniformly, all of them are finding their Segregation of Duties (SoD) policies are one of the key obstacles to faster transformation. Due to blockchains' intrinsic traits, Bi-modal DevOps processes implemented on a blockchain will inherently enable a transparent SoD compliance, increasing an organization's delivery efficiency and agility.

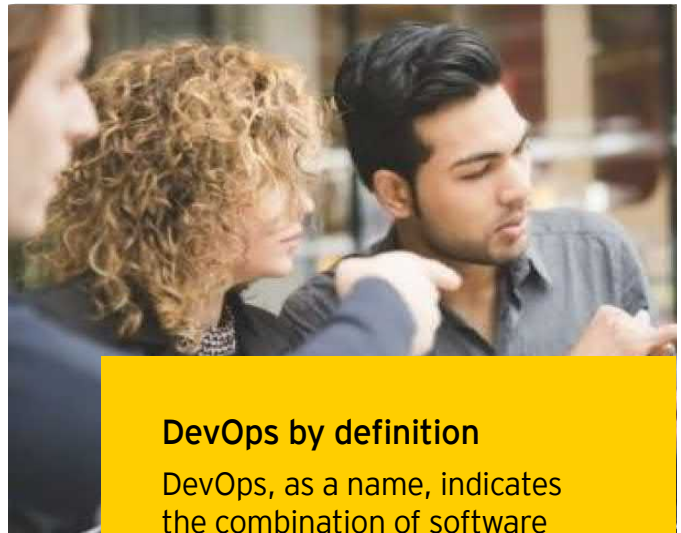
Companies implemented their SoD policies more than a decade ago in response to control-driven regulations, including Sarbanes-Oxley (SOX) in the US, the European Union's 8th Directive, viewed by some as Europe's SOX equivalent, J-SOX (the Japanese SOX) and Payment Card Industry Data Security Standard (PCI DSS). SoD in IT delivery means implementing roles, permissions and responsibilities so that one person alone cannot introduce a technology change in a production system without auditable control. However, SoD policies were defined for the traditional "waterfall" model of software development, whereby separate environments and teams are established at every stage of the delivery process, each separated from the other by physical and logical controls.

To stay competitive and respond quickly to perpetual industry disruptions, companies have to keep pace with business automation and technology delivery transformations in industry, while continuing to meet the control requirements throughout the development and production lifecycle. This renders the traditional waterfall approach unsustainable.

Adoption of DevOps software development model is driven by the need to increase organization's agility and accelerate business transformation to quickly respond to ever more frequent disruptions in their industries. Organizations are adopting DevOps to accelerate overall software development lifecycles (SDLC), integrate agile development, continuous change management and rapid deployment. These delivery cycles need to be executed over a shorter period of time – sometimes as quickly as a few hours.

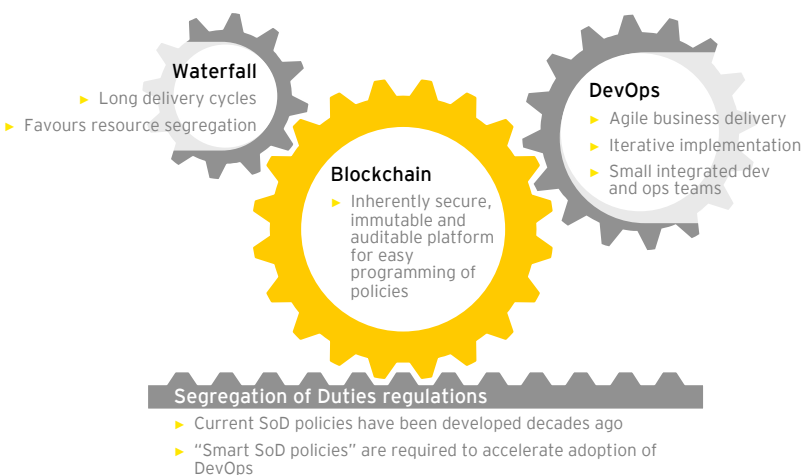
In DevOps environment, traditional SoD controls become "walls" physically separating teams, which ultimately increases in the number of development, testing, deployment and maintenance resources, slowing down delivery of the final product. To accelerate implementation of DevOps across an enterprise, we need an innovative solution that will efficiently and seamlessly implement SoD controls in an agile environment.

Blockchain natively brings features that can be easily extended to implement existing and new controls compliant to SoD regulatory requirements. In this document, we outline a practical, risk-based approach to managing controls in DevOps environments based on blockchain technology.



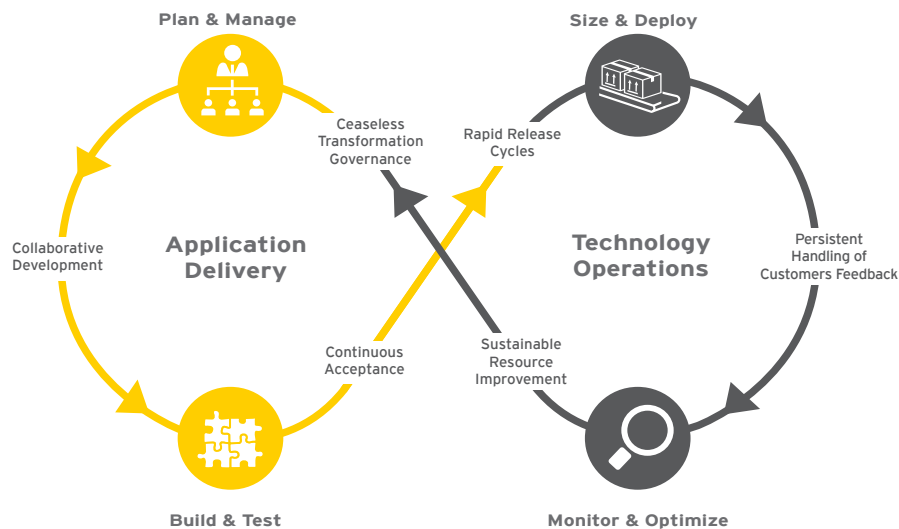
DevOps by definition

DevOps, as a name, indicates the combination of software development (dev) and operations (ops), with a focus on cross-departmental integration and automation. The goal is to create a culture and environment that will increase delivery speed through process automation and reduce costs while continuing to meet control requirements.



DevOps

Accelerating software delivery and IT support processes



DevOps indicates the combination of software development (dev) and operations (ops), with a focus on cross-departmental integration and automation. The goal is to create a culture and environment that will increase delivery speed through automation and reduce costs. The DevOps approach to SDLC requires trust and collaboration among multiple parties, while ensuring compliance with control requirements.

Agile development models, characterized by cyclic and incremental implementation of a solution, is an integral part of DevOps delivery. Projects are divided into cycles, or sprints, each of which involves similar activities but incorporates additional features. The objective at the end of each sprint is to have a minimal viable product or MVP that can be transitioned into production.

While DevOps has been in practice for the last few decades, it has gained greater prominence in recent years as many of the leading tech giants have adopted it.

In short, DevOps model includes the following key characteristics:

- ▶ Agile development processes
- ▶ Continuous collaboration between business, development and IT operations personnel, including the creation of a blended DevOps role (e.g. DevOps artisan)
- ▶ Rapid and frequent production releases, sometimes even a few times throughout the day
- ▶ Apply modern development tooling combined with automation of testing and production deployments
- ▶ Increased heavy use of cloud and virtualized environments

- ▶ Access to production systems to troubleshoot and remediate issues

In compliance with industry control standards, delivery teams implement processes with the following key objectives:

Fraud and error prevention

- ▶ Ensure roles, permissions, responsibilities and processes are set up so that one person cannot alone introduce a change without oversight.

Process change control

- ▶ Ensure the enforcement of controls through an automated process.

Auditability

- ▶ Provide evidence of how the process was enacted over the life of a project or system.

Auditable Bimodal DevOps delivery

Transition from “waterfall” to “agile” delivery

Traditionally, IT delivery processes, were designed with “waterfall” project delivery in mind and as such these processes enforce physical segregation between developers, testers, and IT operations.

However, control requirements simply mandate that the software delivery process must safe guard against unauthorized changes to production systems and provide auditability. The control requirements do not mandate the number of staff working on a project to deliver the solution.

In large modern enterprises, the heterogeneous technology environment necessitates Bi-modal DevOps. This is a two-speed solution delivery model, where “agile” sprints are throttled and synchronized with “waterfall projects” to implement changes in systems.

In Bimodal DevOps programs, deployments in production are gated by “go/no-go” decisions across all projects. This allows for common governance, delivery management, planning, sizing and business readiness. To adhere to rapid dynamics of agile delivery, developers might need role-based access to production environments, while maintaining an indisputable trace of their activities.

Bimodal DevOps implementation that is compliant with control requirements should also increase communication and collaboration and establish mature change management processes. This creates an environment whereby an unscheduled or unauthorized change will be identified and corrective action will be taken.

The following are some examples of delivery controls that can be audited in a DevOps model:

- ▶ Management of delivery artefacts, e.g. requirement, usecase, design, development, test and release documents
- ▶ Traceability of code changes, code branches and release tags
- ▶ Authorizations for builds and promotion of code into delivery environments, especially production
- ▶ Access to production systems through role-based access controls
- ▶ Logging with the ability to review delivery requests, approvals and activities

To further increase delivery efficiency and transparency organizations should consider automating their Bimodal DevOps processes.



What is blockchain?



Blockchain was initially designed as a technology to enable cryptocurrencies, and is built on the principles drawn from cryptography, game theory and peer-to-peer networking. It is a networking technology, similar to world-wide-web (www), that enables a decentralize exchange of data. In a wider sense, blockchain is a distributed database (ledger), which maintains a continuously growing list of timestamped and encrypted transaction records organized in blocks, with each block being linked to a previous block, forming a chain.



A blockchain natively has the following characteristics:

Decentralized

- ▶ It is a distributed network of databases with potentially no central authority.

Synchronized

- ▶ All activity on the blockchain is automatically reflected across the network in near real-time.

Traceable

- ▶ The transaction timestamp is recorded in a block, i.e. all data exchanges can be verified to have taken place at a point in time.

Immutable

- ▶ Validated data is irreversible and cannot be tampered with preventing fraudulent or accidental overwriting. Its built-in hashing, linking and consensus algorithm prevents change of historical records.

Secure

- ▶ All recorded transactions are individually encrypted, but the transactions can be traced back through their cryptographic identities.

Scalable

- ▶ A new participant can be rapidly added to the network and a full copy of the ledger will be replicated in their node.

Collaborative

- ▶ All network participants agree to the validity of each of the records. A new block is only adopted by the network once a majority of its participants agree that it is valid.

Auditable

- ▶ Provides visibility into transactions through digital signatures, which binds each party to the data exchange in real-time. The digital signatures can be tied back to real-life identities depending upon the implementation.

Programmable

- ▶ A concept known as “smart contracts” allows for instructions to be embedded within nodes, allowing automated rules to be executed for each transaction when predefined conditions are met.

Smart contracts are becoming a central capability of blockchain platforms, where rules can be embedded into the blockchain through code. A smart contract is a computer program code and network binding document that is capable of facilitating, executing and enforcing the negotiation or performance of an agreement.



With the establishment of Bitcoin as the first implementation of blockchain technology, followed by the introduction of other cryptocurrencies, most people have some idea of how blockchain technology is applied in the context of a public network infrastructure, also called “unpermissioned blockchains.”

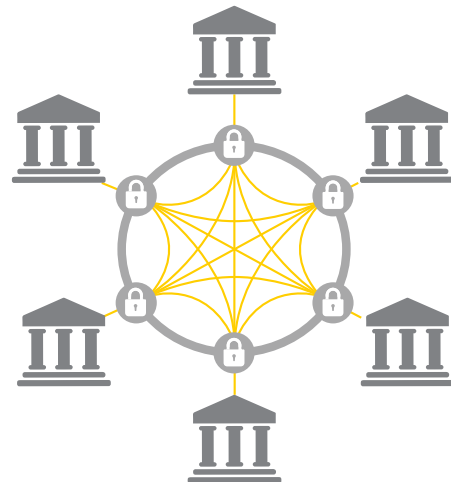
Over the last few years, industry focus has shifted to private and consortium distributed ledgers, commonly referred to as “permissioned blockchains”. The “permissioned blockchains” are usually faster, due to simplified consensus protocols relying on a smaller number of trusted nodes maintaining and validating ledgers’ integrity.

Unpermissioned blockchain



- ▶ All parties can read the transaction data.
- ▶ Anyone can validate transactions or add blocks.
- ▶ No need for a previous relationship with the ledger.
- ▶ No gating or authorizing process to enroll into the transactions scheme.
- ▶ Ledgers replicate the high degree of trust.
- ▶ Theoretically, they are public ledgers.

Permissioned blockchain



- ▶ Formed by a set of known transacting parties.
- ▶ Validation is controlled by a selected set of nodes.
- ▶ Specialized verifiers can be added with the agreement of the current members.
- ▶ Intended to be purpose built, and thus created to maintain compatibility with existing applications.
- ▶ Ledgers replicate the high degree of transparency and accountability.

"Waterfall" delivery

Traditional waterfall model of software delivery

The traditional waterfall delivery process, with SoD controls typically have the following steps:

- 1 A business analyst identifies a problem.
- 2 Developers write the application and enter code into the repository.
- 3 Testers take the application code and test it in a Quality and Assurance (QA) environment.
- 4 Change management requests production deployment.
- 5 The change approval board (CAB) reviews and approves implementation into production.
- 6 The IT operations team takes the tested and certified code packages from the repository and builds release, which is then deployed into production.

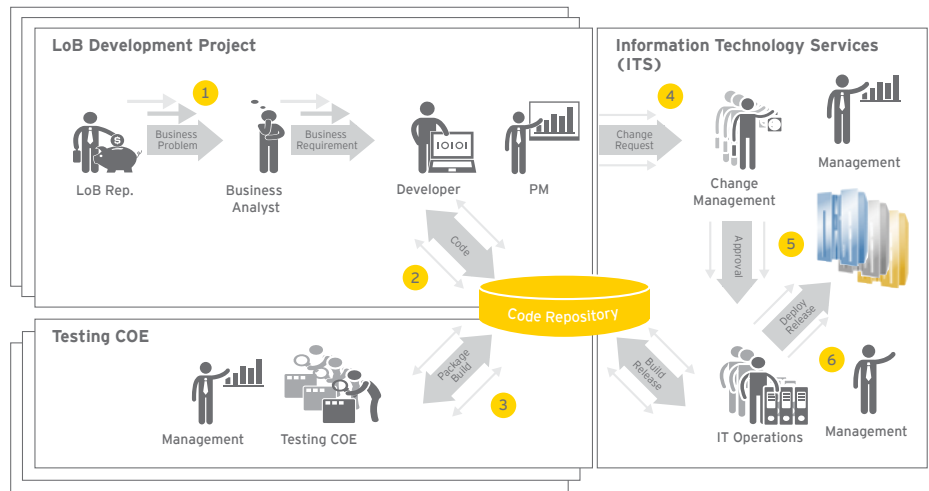


Figure 1. Traditional waterfall service delivery process compelled by organizational SoD confinements

The reality of traditional SDLC is usually more complex, especially in enterprises with multiple line of businesses (LoB), where each LoB has its own growth strategy, history and legacy technologies. The reality looks more like the example in figure 1 with the following characteristics:

- ▶ LoB development teams use different delivery frameworks, code repositories and development tools.
- ▶ Multiple developers are involved in activities with potentially multiple code repositories both within and across LoBs.
- ▶ Organizations that have a testing centre of excellence (TCOE) may have unique challenges integrating in a multiple LoB environment.
- ▶ Development environments may be heterogeneous that would further erode efficiencies.
- ▶ There are non-standardized development environments across various projects, due to the various LoB delivery strategies and development platform specifics. These usually further hinder the efficiencies of generic testers, which are sourced from the shared pool (e.g., the COE).

- ▶ Change management often is a stand-alone function, not integrated with SDLC. This causes change management to be viewed as an obstacle that slows delivery.
- ▶ The change requests usually are not standardized, making it difficult to fully trace their origin, participants or impacts, even when there is a single enterprise change management platform in place.

Organizations facing these types of situations should consider adopting DevOps as a means to increase the pace of software delivery.

Enterprise DevOps Blockchain

Bimodal DevOps implementation using blockchain

As organizations aggressively adopt DevOps, the question is: can blockchain enable and expedite DevOps implementation in large enterprises?

A blockchain based code repository would offer a decentralized solution with auditability and immutability. The new intermediary would sequentialize delivery and manage process latencies.

A conceptual solution for implementation of the DevOps auditable change control process using a permissioned blockchain is shown in figure 2.

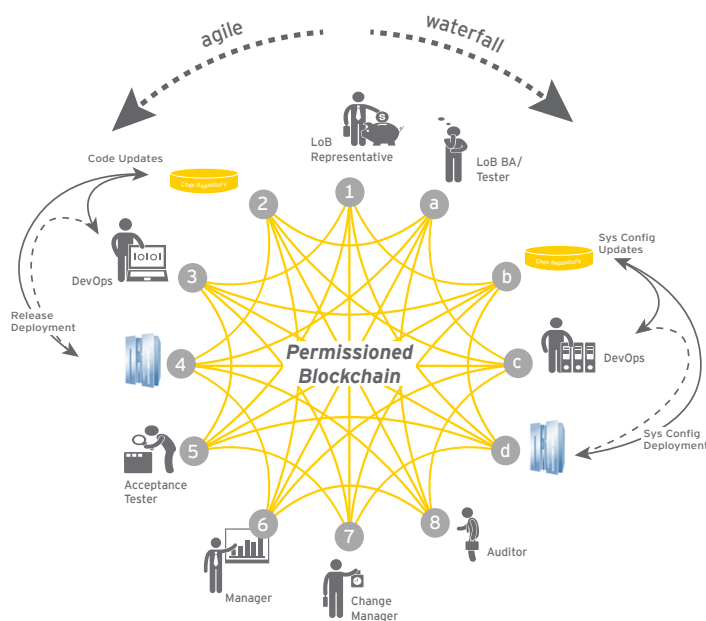


Figure 2. Bi-modal DevOps implementation using blockchain

Key components and characteristics of the proposed Enterprise DevOps Blockchain (EDOB) solution are as follows:

- ▶ All activities are automatically recorded as transactions on EDOB.
- ▶ All network participants run nodes in EDOB, which is implemented as a permissioned blockchain, and smart contracts are executed for each transaction (i.e., DevOps requests, approvals or activities).

- ▶ The definition of a transaction is extended to include a proof of ownership of modules, module change activity, authorization and validity.
- ▶ Transactions can be independently validated and processed.

EDOB can be implemented using single-chain or multi-chain blockchains as an underlying infrastructure. The choice of underlying blockchain infrastructure is independent of the approach, implemented processes or on user experience.

The EDOB solution follows **agile DevOps activities** represented in figure 2 in **counter-clockwise direction**.

1. The problem is captured as a root-transaction into EDOB, thus initiating a blockchain trace, where any related delivery activity is recorded as a subsequent transaction.
2. Code repository are integrated with EDOB and activities within the repository are treated as a transaction in the network.
3. Examples of activities that would be transactions in the EDOB:
 - ▶ Develop new code, update existing code or finalize code for release
 - ▶ Test and regression-test features delivered during a sprint
 - ▶ Investigate issues and promoting code in the production environment
4. In an ideal scenario, deployment of the code into production is fully automated and also recorded as an EDOB transaction:
 - ▶ Furthermore, any access to and any activity in production is recorded in EDOB; this includes manual deployment activities, if the deployment process is not fully automated
 - ▶ Automated vulnerability scanning at release build time can be triggered and its outcome will be treated as a separate transaction recorded in EDOB



5. The finalized code for release can trigger notification through smart contract to a tester to commence acceptance testing (i.e., execution of end to end functional, security and technical test-cases).
 - Acceptance testing is automated to execute a set of regression tests.
 - Each executed test script is captured as a transaction into EDOB.
 - The tester's acceptance of the release is captured as a release ready transaction in EDOB.
 6. Management and responsible executives have transparency over projects in EDOB.
 - Any management activity request or approval can be recorded into EDOB.
 7. The release ready transaction allows management to trigger a notification to change management for approval.
 - As EDOB keeps a full and immutable record of the history of preceding activities, the CAB or other approving body will have all the required information to quickly make a decision to approve or reject the request.
 - Upon review and approval, a new transaction is captured in EDOB, and it can be used as a trigger to automatically execute promotion of the approved release into production
 8. At any point, internal or external auditors can review the process and access previously completed, in-delivery or planned projects, overseeing the organization's compliance with control requirements.
- b) A code repository, is integrated into EDOB, recording all the activities as transactions in the network.
 - c) Any change in production will be automatically recorded into EDOB as a new transaction.
 - d) Code changes can trigger testing activities, which will be recorded into EDOB.
 - e) All requests and approvals are captured by the project manager in EDOB as transactions.
 - f) Similar to step seven in agile delivery, a change request from an IT project manager recorded on EDOB can trigger a change manager review.
 - g) Internal or external auditors will have full transparency regardless of the project, team or methodology used during the delivery.

In Bimodal DevOps, teams can apply different delivery models. For example, teams may use the agile approach to deliver digital capabilities, while other teams, working with a legacy system will use the waterfall approach.

The flexibility of the EDOB solution to support different delivery models allows organizations to transition from traditional to DevOps delivery, while maintaining auditable and transparent records.

Not only does EDOB enable the implementation of control compliance of agile continuous delivery, it also transparently **supports the traditional waterfall delivery** approach. This is represented in the **clockwise scenario in figure 2**.

- a) A business analyst can submit a new business requirement document (BRD) that is recorded as a new transaction in EDOB.
 - Smart contract rules impose mandatory approval requirements, and the outcome will be recorded as a new transaction.
 - The automatic notification can trigger execution of delivery activities once the BRD is approved.

Benefits of EDOB

Bimodal DevOps Blockchain

Bimodal DevOps delivery implemented using an EDOB solution will increase organizational transparency and process auditability, by addressing a number of existing challenges or challenges that may arise during a transition to full adoption of DevOps.

In a highly regulated environment, there are many benefits to the traceability an EDOB solution offers:

- ▶ A trusted data source where transaction records once created and confirmed cannot be altered or removed from the ledger.
- ▶ Meet control requirements using veracity and authenticity of immutable facts, data, processes and events related to any transformational change.
- ▶ Smart-contracts built-in mechanism allow for dynamic requirements compliance.
- ▶ Smooth transition from traditional delivery to faster and more agile delivery.
- ▶ Improved security and increased automation of delivery in simple, auditable and compliant fashion.

Overall, EDOB is a good candidate for early adoption of blockchain technology by an organization due to its narrow focus on increasing efficiency and agility of delivery processes. By implementing EDOB, technology teams will apply blockchain technology, allowing them to gain insights into best practices and improve technical skills, allowing for adoption of the blockchain technology in other areas of the business.

Additionally, an EDOB solution brings a number of technical benefits, which are inherently acquired from blockchain characteristics:

- ▶ The distributed nature of blockchain is designed for fault tolerance and provides high availability and disaster recovery.
- ▶ Theoretically, it is an infinitely scalable network infrastructure, and will be automatically synchronised by replicating the content of the EODB ledger.
- ▶ By relying on peer-to-peer network architecture, the solution runs 24/7, where each node is managed by a respective network participant.
- ▶ Through “smart contracts”, EODB can easily be integrated into an existing delivery infrastructure by relying on API interfaces.



Approach to DevOps adoption ...

... and EDOB solution implementation

Organizations adopting DevOps and considering implementing an EDOB solution should start with a definition of a trust model for their organization. Transformation of existing capabilities or adoption of new ones, implemented using a blockchain-based solution, requires identification of areas within the organization that are trusted vs. those that are not.

The application of a blockchain-based solution should always start from those that are assessed as less trusted. Those who are trusted can execute controlling functions in permissioned networks. Implemented for technology delivery processes, it means analysis, development, testing and deployment are less-trusted areas, while management, change management and audit functions are oversight and control areas with higher trust.

The process of adopting the DevOps model should be iterative, starting with a pilot, in order to control the scope, impact and risks on the organization. Each iteration should execute the following six steps to accelerate adoption of the model with every iteration and to ensure successful implementation of an EDOB solution:

- ▶ **Scope definition** sets out an understanding of the enterprise delivery roles, respective activities, and defines a sustainable stage of bi-modal DevOps processes once the iteration is complete.
- ▶ **Design** a map for each activity in the delivery process that needs to generate an EDOB transaction and identify its associated roles and access rights.
- ▶ **Implement** by incorporating additional rules into new or existing smart contracts in EDOB and add nodes to include any additional bi-modal DevOps delivery participants.
- ▶ **Testing** draws on data from previous steps to produce an analysis of conflicts between the established roles and activities and existing SoD policies. The results should highlight conflicts with SoD policies by user, by role, by group or by activity. This analysis serves as the compliance testing package disclosed to management, audit parties and regulators.
- ▶ **Mitigation** limits the potential impact of SoD conflict violations. This step can be completed concurrently with remediation or, it can be performed last, when conflicts have been reduced to their minimum.
- ▶ **Remediation**, with the goal of permanent correction of SoD conflicts, includes role redesign, role cleanup, user appropriateness review, or SoD policy updates according to control requirements.



There is no prescribed leading practice or method for remediation of conflicts. Remediation activities generally fall into two categories: tactical cleanup of the user population and strategic role redesign. The tactical component represents the items that can be addressed quickly, while role development typically involves a full complement of organizational changes in people, processes and technology.

Conclusion

SoD policies remain an integral part of an organization's internal controls. These policies have been designed in response to controls and regulatory requirements which have been designed to prevent fraud and material misstatements. These regulations mandate that controls be put in place to ensure that no individual has excessive rights to execute transactions across an entire business process without trusted checks and balances.

Many organizations struggle with their legacy SoD; on one hand there are control requirements that must be adhered to and on the other hand, there is the need to be more agile, responsive and faster to market. In addition, the complexity of delivering a change across numerous enterprise systems leaves many organizations struggling with implementation of basic internal controls.

While the appropriate level of effort and emphasis must be placed on compliance with the organization's current SoD policies, companies must also strive for simplicity and precision in the execution of their controls. Blindly following internal directives defined more than decade ago can be a costly mistake that hinders the benefits of DevOps in large organizations, especially those with multiple LoBs or service lines.

Blockchain technology is well positioned to implement Bimodal DevOps delivery that meets control requirements in business and technology environments of all sizes. A well-understood, role-driven, documented and automated service delivery process is only a requirement, which, when implemented on a blockchain using smart contracts, will provide immutable traceability of approved delivery activities.

In an environment of rapid software development and deployment, a well-designed, risk-based Bimodal DevOps delivery implemented using EDOB will enable compliance, enhance controls, streamline and redesign key delivery processes. This, in turn will increase an organization's efficiency and agility.

References

- "DevOps and Segregation of Duties, Bob Aiello November 2016," *Agile ALM DevOps Website*, <http://agilealmdevops.com>, accessed January 2017.
- "Auditing DevOps - Developers with Access to Production," Douglas Barbin on December 2012, *Schellman website*, <https://www.schellmanco.com>, accessed January 2017.
- "DevOps Survival in the Highly Regulated Financial Industry," Manuel Pais, July 2016, *InfoQ website*, <https://www.infoq.com>, accessed January 2017.
- "An introduction to immutable infrastructure," Josh Stella, June 2015, *O'Reilly website*, <https://www.oreilly.com>, accessed January 2017.
- "Continuous Delivery and ITIL: Change Management," November 2010, *Continuous Delivery website*, <https://continuousdelivery.com>, accessed January 2017.
- "Avoiding the pointless blockchain project," Gideon Greenspan, November 2015, *MultiChain website*, <http://www.multichain.com>, accessed January 2017.
- "Smart Contracts Explained," *Blockchain Technologies website*, <http://www.blockchaintechnologies.com>, accessed January 2017.
- "Banking on Blockchain: Charting the Progress of Distributed Ledger Technology in Financial Services," John McLean, Finextra Research Ltd, 2015, *Finextra Research Ltd website*, <https://www.finextra.com>, accessed January 2017.
- "Understanding blockchain and the opportunity for financial institutions," 6 October, 2015, *Banking Tech website*, <http://www.bankingtech.com>, accessed January 2017.
- "Blockchain: Powering the Internet of Value," Peter Frøystad, Jarle Holm, 2015, *Evry website*, <https://www.evry.com>, accessed January 2017.
- "Top 10 Mistakes in Enterprise Blockchain," Ray Valdes, David Furlonger, Dale Kutnick Dec 2016, *Gartner website*, <https://www.gartner.com>, accessed January 2017.
- "Cryptotechnologies, a major IT innovation and catalyst for change," May 2015 EBA Working Group on Electronic and Alternative Payments, *Euro Banking Association website*, <https://www.abe-eba.eu>, accessed January 2017.
- Don Tapscott, Alex Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World* (Portfolio 10 May 2016).
- William Mougayar, Vitalik Buterin, *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology* (Wiley 2016).
- A risk-based approach to segregation of duties*, EYGM Limited, 2010.
- Blockchain: the hype, the opportunity and what you should do*, EYGM Limited, 2016.
- The evolution of distributed ledgers and the future of financial services*, EYGM Limited, 2016.
- SecureDevOps - is security the bottleneck?*, EYGM Limited, 2016.

The blockchain labs are the latest in EY's multi-million-dollar investment in EY wavespace™ - a network of global innovation and growth facilities that drives innovation in disruptive technologies and links EY firms, labs and professionals worldwide. To learn more about EY wavespace™ and how we can support innovation for your organization, contact one of our professionals:

Kimberly Connors

+1 416 943 4466

kimberly.connors@ca.ey.com

Petar Nikolic

+1 416 943 3114

petar.nikolic@ca.ey.com

Ron Stokes

+1 416 943 3013

ron.stokes@ca.ey.com

Nebojsa (Voya) Vojinovic

+1 416 943 2126

nebojsa.vojinovic@ca.ey.com

Abhishek Sinha

+1 416 943 4537

abhishek.sinha@ca.ey.com

Ivica Popovic

+1 416 943 4460

ivica.popovic@ca.ey.com

Pramod Gopalakrishna

+1 416 943 4560

pramod.gopalakrishna@ca.ey.com

Thierry Belanger-Roy

+1 514 874 4639

thierry.belangerroy@ca.ey.com

EY | Assurance | Tax | Transactions | Advisory

About EY

EY is a global leader in assurance, tax, transaction and advisory services. The insights and quality services we deliver help build trust and confidence in the capital markets and in economies the world over. We develop outstanding leaders who team to deliver on our promises to all of our stakeholders. In so doing, we play a critical role in building a better working world for our people, for our clients and for our communities.

EY refers to the global organization, and may refer to one or more, of the member firms of Ernst & Young Global Limited, each of which is a separate legal entity. Ernst & Young Global Limited, a UK company limited by guarantee, does not provide services to clients.

For more information about our organization, please visit ey.com/ca.

© 2017 Ernst & Young LLP. All Rights Reserved.

A member firm of Ernst & Young Global Limited.

2269018

ED None

This publication contains information in summary form, current as of the date of publication, and is intended for general guidance only. It should not be regarded as comprehensive or a substitute for professional advice. Before taking any particular course of action, contact EY or another professional advisor to discuss these matters in the context of your particular circumstances. We accept no responsibility for any loss or damage occasioned by your reliance on information contained in this publication.

ey.com/ca

