Managing stakeholder trust through DevOps

The KPMG Controls Observability Solution

KPMG Technology Risk Insights

What you can't see can hurt you

The complexity of modern IT environments often creates new ways for technology to fail. In the Software Development process, the introduction of DevOps changed the way business and IT develop and deliver value. Delivering value to the customer quickly is prioritized and achieved through the use of automation ruthlessly throughout the process.

However, this automation does not come without its own set of risks. As organizations look for ways to automate their processes, no "one size fits all" approach emerges. Organizations are required to identify business cases, select tools and technology, and implement them, all before integrating the data into their existing routines. A recent survey shows that large modern organizations are relying on 25 or more tools in their Software Development lifecycle. These tools all produce separate metadata that is required to be analyzed and monitored by the teams using these tools – often, in disjointed and repetitive methods.

The proliferation of these tools also increases the risk that these tools, often crucial in the automation and deployment of value to the customer, are configured incorrectly, either intentionally or by accident. The risk that a user turns off a configuration to save time in development may seem innocent enough – but when the process relies on automation, even one misconfiguration could result in an outage. This requires organizations to attest to not only the functionality and correct configurations of systems, but also ensuring that access is locked down appropriately across the entire toolchain. In a modern organization where roles are fluid, this sort of rigidity can slow down deployment of value greatly, reducing the value of these tools and leaving teams frustrated.

What if there was a way to mitigate these risks and overhead burden on teams across the organizations' three lines of defense? If there was a way to utilize the full potential of these tools and the benefits of an automated environment, while still providing a framework that allows full visibility into your development processes at a granular, tactical level? This is where the true value of the controls observability framework shines. The Controls Observability framework solution allows organizations to define control requirements and systematically assess compliance in an automated, real-time manner as changes move throughout the development lifecycle from planning to deployment. While successful organizations are prepared to both prevent and acknowledge failure, they all can safeguard against damage with effective controls and continuous monitoring.

Previously, we've introduced the Controls Observability Framework, an 8-step approach to designing, implementing, and improving monitoring capabilities. In this post, we'll expand upon this and discuss how organizations are operationalizing and implementing these solutions, following the below 8 steps for designing, developing, and deploying an observability solution with visibility across the Software Development Lifecycle:

Controls Observability Framework Operationalization

1 Define

By leveraging Policies, Standards, and control frameworks, organizations can create a robust control inventory by understanding processes and the detailed data generated by these processes, and what requirements are needed for control compliance in collaboration across the lines of defense.

2 Ingest

By leveraging existing technology such as automated jobs, scheduled tasks, real-time API calls, and webhooks, organizations can retrieve relevant metadata on an event-by-event basis for analysis.

3 Standardize

Once the data has been collected, a process should be undertaken to standardize the language and presentation of the metadata for easy processing. The raw results should be stored in a safe and standard location for analysis as needed.

4 Validate

Once results have been cleaned, by using a systematic policy engine configured in collaboration with all stakeholders, organizations can automate the "testing" of policy compliance for each control defined. This engine should be standardized so that users have the ability to modify, update, classify, and stratify for control compliance on an "as needed' basis, while still enforcing compliance with a minimum set of standards.

5 Report

The "Pass/Fail" result for each policy checked should be reported along with the rationale and systematic tie-back to the source data. This result should be stored and represented in an easy-to-read dashboard for analysis and reporting on compliance with defined controls.

6 Notify

By developing a notification schema based on the control failure, control risk, involved parties, and other requirements, notifications of non-compliance can be sent in a real-time manner, such as email or messaging application, as issues are identified.

7 Resolve

Once stakeholders have been notified, root cause analyses can be conducted to identify why the non-compliance has happened. This non-compliance is then tracked to resolution, and any impacts are resolved. This analysis can be automatically generated leveraging existing service resolution tools and workflows.

8 Prevent & Enhance

By using the policy non-compliance results, coupled with learning models, organizations can identify "problem areas" to enhance controls and enforcement of processes, both systematic and procedural, to reduce non-compliance areas identified.

The KPMG Controls Observability Solution can systematically show how this platform, which combines the 8 steps above, can identify instances of non-compliance with policies (such as peer review requirements), and how organizations are made aware of these instances in real-time to identify, resolve, and prevent non-compliance. This results in a robust observability program enabling organizations to deliver value to customers that is quicker, safer, and more cost-effective.

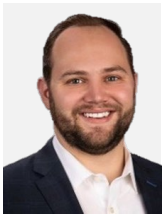
KPMG Controls Observability Framework

Discover more insights on how organizations can leverage the Control Observability framework to provide observability throughout their control processes.

Contact us

Lavin Chainani
Managing Director,
Technology Risk
KPMG LLP
T: +1 410-949-8834
E: lchainani@KPMG.com

Chip Deskins
Manager,
Digital Lighthouse
KPMG LLP
T: +1 919 664 7318
E: jdeskins@KPMG.com



Andrew Hall
Manager,
Technology Risk
KPMG LLP
T: +1 480 459 3480
E: ajhall@KPMG.com

Some or all of the services described herein may not be permissible for KPMG audit clients and their affiliates or related entities.

kpmg.com/socialmedia