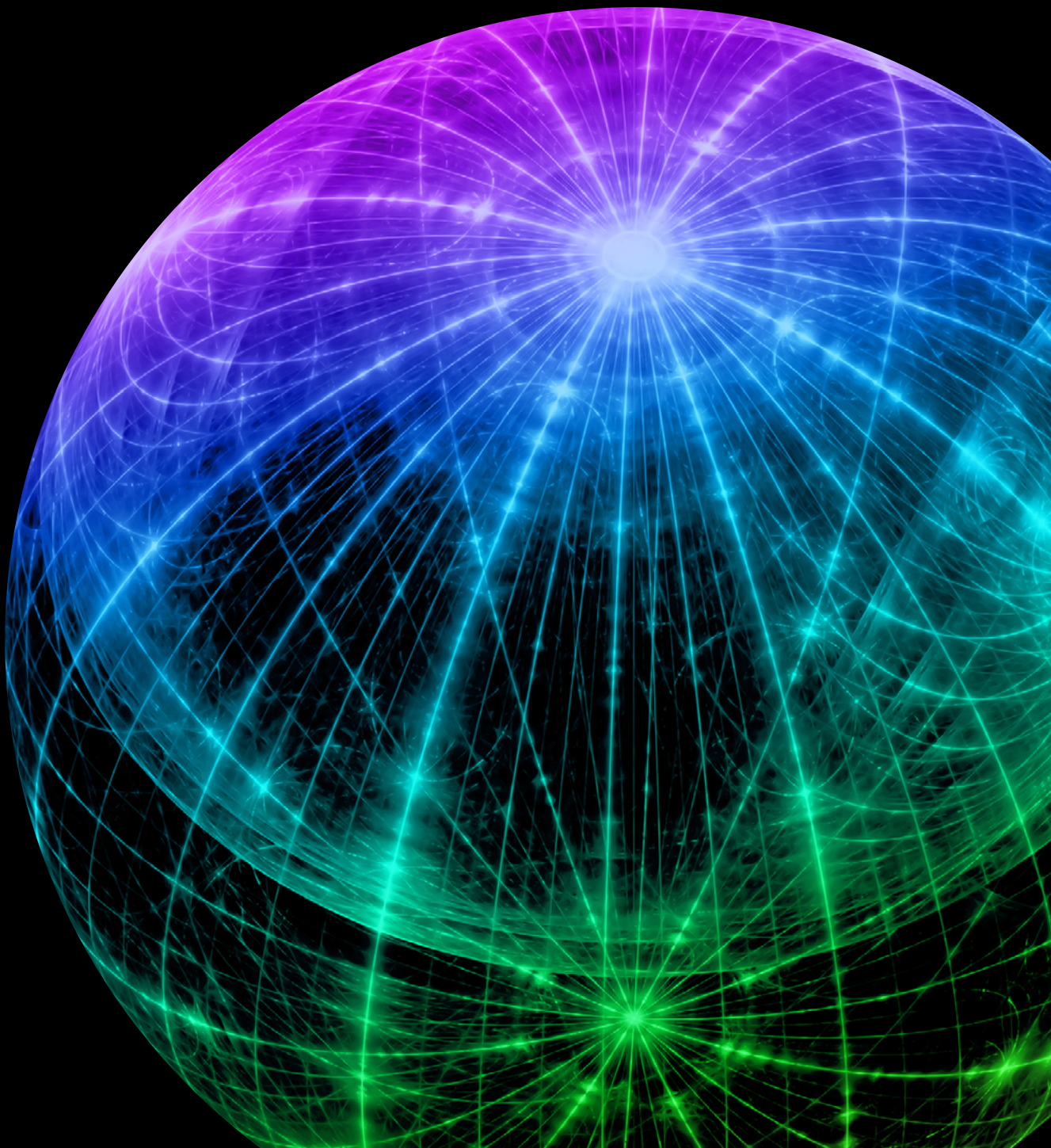
Deloitte.

Internal Audit risk considerations for platform companies



The digital age has enabled the meteoric rise of platform-based companies, many of which have become an integral part of our lives. As the industry defines it, the term “platform company” describes businesses that facilitate exchanges between interdependent groups, typically consumers and producers. These are the social media, content, payment, and other exchange platforms of the world. These companies commonly build their custom platforms to run their business, and no one platform is built the same way.

Platform companies frequently leverage one or a combination of modern software development methodologies (Agile, DevOps, Continuous Integration/Continuous Delivery, etc.) to rapidly meet and scale their business vision. Due to the variety of development methodologies available to the developers of in-house built platforms, Internal Audit (IA) departments will need to rethink how to approach risk considerations surrounding their in-scope processes, specifically around identification of proper general technology controls to mitigate technology-related risks.

Why may this be the case? As technology continues to advance and more open-source tooling options are made available, engineers frequently use a myriad of customizable tools to develop their in-house built platforms. This inherently creates dispersed and complex processes in a control environment and introduces new risks not previously considered. Internal auditors will likely then need to ask smarter questions and think differently to be able to identify specific risk areas in the organization's engineering processes before identifying the proper control points for risk mitigation.

Internal Audit risk considerations for platform companies

Platform risks: Change management

Segregation of duties (SOD) in a DevOps environment

SOD in a DevOps environment differs from SOD in a traditional monolithic application architecture environment. It is typical for software engineers in a DevOps environment to follow an agile process to implement the changes they develop after receiving a peer review. Therefore, achieving SOD requires implementation of preventive or detective measures throughout the DevOps process to prevent an individual software engineer from unilaterally developing and implementing a production code change. Typically, although not necessarily, this is achieved by requiring peer review on code changes combined with restrictions on build and release tooling to prevent unapproved code changes from being deployed to production.

Source code management (SCM)
Changes to source code can be made via the user interface (UI) of the SCM tool, via a command line interface (CLI), or in a local editor such as Visual Studio. Without proper configurations within the SCM tool, changes could be made to the source code without review or approval. If additional controls are not in place in downstream systems, this creates a potential segregation of duties risk in which a single engineer could make a code change and deploy it to production without review or approval.

Testing
Multiple methodologies can be used to test a code change. Failure to conduct unit testing could lead to a failure to identify errors in the code introduced by the change. Failure to conduct integration testing could cause a disconnect between the microservice that it is being made to and the upstream and downstream microservices it both relies on and is relied upon for. Failure to conduct end-to-end testing could result in failure to identify an unanticipated impact that the change to the microservice had on the broader process it supports. In a mature DevOps environment, financially relevant testing may also be conducted to reduce the risk that a change will likely have an intentional impact on downstream financial reporting.

Build
Build tooling is used to prepare new or modified source code into a format readable by the production hosts the code will be deployed to. This tooling can be used to point to code sourced from an unapproved location (i.e., private builds, external docker images, etc.). Such activity will likely not be easily identifiable via controls placed on downstream systems because it may appear to have been created appropriately from within the build tool. This creates the risk that upstream change management controls over SCM and testing could be circumvented without detection.

Artifact storage
Artifact storage locations are used to hold code prepared by the build tooling prior to being deployed to the production environment. Changes to artifacts stored within the artifact storage tool creates the risk that upstream change management controls over SCM, test, and build tooling could be circumvented without detection.

Deployment
Deployment tools take the code prepared by the build tooling and upload it to production. Configuration changes within the deployment tooling could allow engineers to deploy software sourced from outside company-accepted tooling (i.e., private builds, external images, etc.), creating the risk that upstream change management controls over SCM, test, build, and artifact storage tooling could be circumvented. Without proper monitoring in place over deployments put into production, these activities could go undetected.

Platform risks: Access security

Direct and indirect access to production

Production access can be granted to both hosted and on-premise production hosts via UI, CLI, and/or other protocols such as secure shell (SSH), remote desktop protocol (RDP), and AWS Systems Manager (SSM). Users with direct access to production can modify the code running on the server and/or the data stored on the server. Additionally, configuration changes could be made directly to the production hosts themselves, allowing the host to receive potentially unapproved software.

Further, in a DevOps environment, multiple tools are typically used to deploy code to production. Users with privileged access to these tools can modify them to circumvent change management controls within the deployment process, leading to the risk of unintentional or inappropriate changes being made to the production environment. Additionally, some DevOps environments are set up such that the tooling is granted privileged access to modify the production environment. In these environments, failure to properly restrict access (often at the network level) to the tool creates an attack vector for external threats attempting to access and/or modify the production environment.

Platform risks: IT operations

Integrations and interfaces

In platform architectures, communication between services is vital. It is important for IA to understand the types of integrations and how to address risks related to the completeness and accuracy of data.

Data aggregation and transformation

Very often we find that data flowing through platforms is aggregated, transformed, and cleansed. Tooling and/or in-house written queries are used to assist with such cumbersome tasks, especially given the data-centric world we live in today. With that said, IA needs to consider the risks associated with inappropriate transformation logic, as it can lead to inaccuracies in the data that can be difficult to reconcile or revert downstream.

Conclusion

With the widespread adoption of modern-day software development methodologies and companies building their own different platforms to operate their businesses, IA departments will need to equip themselves with new skills to properly uncover the core risks that require mitigation. Without a strong understanding of these rapidly changing technologies, internal auditors run the risk of not being broad enough in their inquiries to uncover many avenues of potential risks in the established processes.

Authors

Jimmy Yu
Partner, Risk & Financial Advisory
Deloitte & Touche LLP
jamesyu@deloitte.com
+1 714 436 7309

Jason Ho
Senior Manager, Risk & Financial Advisory
Deloitte & Touche LLP
jasonnho@deloitte.com
+1 213 688 6974

John Apel
Consultant, Risk & Financial Advisory
Deloitte & Touche LLP
japel@deloitte.com
+1 303 542 4013

Deloitte.