PLATFORMATION

Whitepaper

How DevOps and GRC go hand in hand

The ultimate balance between agility, compliance and risk awareness

Contents

1. Introduction

There is no doubt that there are challenges to overcome in creating a business that is both agile and yet compliant and risk aware at the same time. In this white paper, we will explain how to tackle these challenges using the power of DevOps, the Enterprise Service Management (ESM) Now Platform and the Integrated Risk Management approach and solution.

Situation

In the current agile-driven business world, (IT) companies are more and more shifting towards self-steering teams and applying concepts such as DevOps and CI/CD to deliver value faster. In doing so, time-consuming process steps like (CAB) approvals and manual unit tests are being eliminated. This means organizations rely more on the skills and expertise of its employees and the automated change controls that they implement and maintain.

Complication

This paradigm shift, where ownership over the change management is more decentralized and resides 'within the business', seems to create potential conflicts for (internal) Governance, Risk and Compliance practitioners. It's their job to implement compliance frameworks and to manage and address corporate risks. This decentralization leads to decreased oversight over change management processes for independent organizational units like CAB and/or Risk & Compliance Depts. This raises valid concerns, because the decreased visibility and transparency–as a result of increased autonomy in Agile teams–might just increase the risk of non-compliance and/or risk exposure due to the implementation of unwanted or unauthorized changes.

Key question

Agile DevOps practices, however, can directly contribute to a more efficient, reliable and effective overall GRC program. Common tools used in DevOps and in CI/CD configurations, such as a deployment pipeline, automated test frameworks, and source control, produce and hold large amounts of (potentially useful) data. This data is often scattered across a wide landscape of different tools though. The key question here is how this data can be consolidated and leveraged by GRC practitioners to (automatically) validate adequacy, appropriateness and applicability of the configured (automated) change controls. In the end, you want to determine whether the risk of unwanted changes is being addressed by these controls set up in the DevOps configuration(s).

2. Our people, processes & technology approach

Harmonizing the worlds of GRC and DevOps requires a 3-way approach, taking the people aspect into account along with changes in processes and technology.

The human factor

It is not uncommon that DevOps team members and GRC practitioners view each other as obstacles, as their short-term objectives are often conflicting. Where the DevOps teams focus on faster delivery of functionality, GRC practitioners need to make sure that this way of working does not harm the (functional) integrity of the systems.

DevOps teams strive for automation of change management processes, as this contributes to their key value drivers–faster delivery time and increased agility. Hence, they follow the Agile Scrum principles, which means more trust is put in the self-steering teams, leading to reduced independent oversight. As a result, the core task of the Compliance Officers and other GRC practitioners will shift from a strictly controlling to a more active role, in which they provide guidance and advise on compliancy and risk during the software development Sprint cycles. This will aid the Agile teams with timely and accurate Sprint assessments and effort estimates.

The GRC practitioner will also actively assist in designing logical and automated functional change tests and/or determine whether other automated test criteria and automated controls are logical and adequate.

GRC practitioners will be actively involved in Sprints

Slighty different processes

With respect to the Change Approval process, a similar decentralization is taking place in DevOps/Agile environments. Traditionally, a Change Approval Board (CAB) is involved in almost all non-standard changes and needs to review and approve them. In Agile teams, however, peer reviews of the code and builds are being performed before anything is committed to the pipeline. Depending on the organizational context, releases to a production environment can only be deployed after approval is given using e.g. an (integrated) workflow-based mechanism or system.

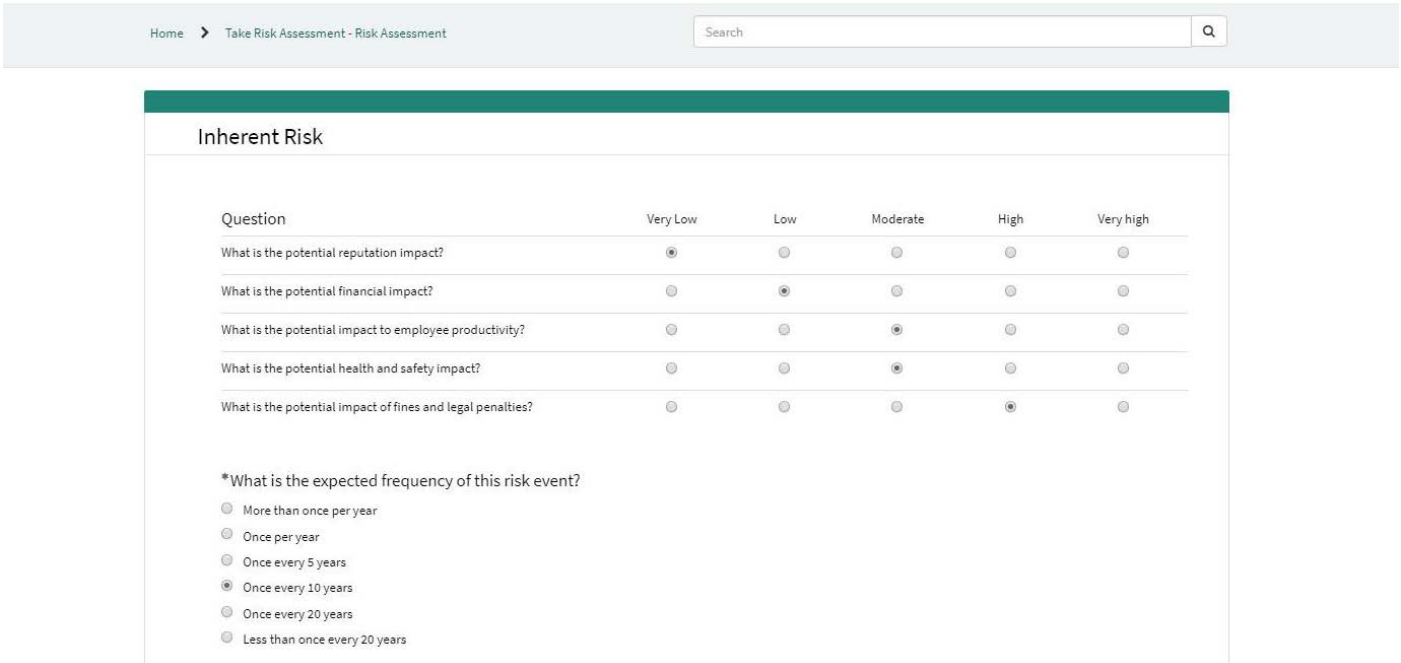

Figure 1. Let end-users perform their GRC activities in the Service Portal.

The lack of an independent Change Approval process may raise concerns with GRC practitioners because of its unreliability.

In order to create a reliable and yet Agile process, the software development Sprint lifecycle will be redesigned, in which standard tasks are being executed by the GRC practitioner to ensure that the operational risks are continuously being monitored and addressed by appropriate change controls. The GRC practitioner will participate in the Sprint cycles during which they advise on activities, to determine whether no unacceptable risks are being taken, such as possible security impacts, confidential data impacted, business features, availability issues, etc. As a result, the observations and recommendations of the GRC practitioners will shift more towards other aspects, such as the test criteria and coverage, and applied automated test framework set-ups, to safeguard the reliability of automated unit and end-to-end regression tests.

Another powerful and effective control mechanism that can be applied is to relate every line of code to a development Story. Subsequently, each Story is related to a business driver and a change. This results in total transparency within the change management and deployment process. You don't have to rely on submitted change ticket content. The GRC practitioner can fulfil an active QA role here during the change process instead of afterwards during (internal) audits. Finally, to mitigate any residual risk, the completeness and accuracy of functional automated unit tests can be periodically validated by e.g. matching development stories to automated test scripts, to further ensure functional integrity of the systems.

Integrating all of these activities in the DevOps way of working will form a highly structured, complete and

Efficiently address all IT change management risks

accurate audit trail for retrospective analysis of the overall change management process in external audits. Above all, it adequately addresses the risk of unwanted changes.

Technological challenges

The challenges for the supporting technology lie in automatic triggering and follow-up of deviations, issues or exceptions in the change management process. The data used to run the triggers is usually already being produced by DevOps tools, such as CI/CD pipelines. However, logical trigger points need to be identified to prevent large numbers of e.g. false positives in the automated key risk indicator sets. You also need to ensure that the right tools are used to manage such data, like automatically triggered follow-up tasks and deviations that enable dynamic reporting on data and workflows.

An integrated solution like the Now platform is needed to easily consume and manage data from different DevOps environments. Then, you can set up dynamic workflows, triggers, deviation criteria and notifications tailored to the particular DevOps pipeline configuration. Some examples of automated change controls in a DevOps environment of which the output can be easily consumed and consolidated are:

- Static code analysis
- Source control output
- Compliance rules for change management
- Automated approval and follow-up event creation for outages

Let's look at each of these in more detail.

Figure 2. Easy to configure dashboards to give an instant overview of all integrated DevOps and GRC data.

Static code analysis

Used to validate the code in the CI/CD pipeline based on various criteria like:

License Scan

It is common to reuse existing code from the internet in the form of libraries, but you will need to check if all of these libraries allow code reuse.

Code Style

Make sure each team member uses the same code style, ensuring a maintainable code base.

OWASP Scans

Perform security scans to filter out the most common faults such as SQL injection.

Vulnerability Scan

Any project may contain as many as 500 dependencies, so this scan helps to identify any known vulnerabilities.

Source control output

Luckily, all code resides in the source control. A developer builds a production version of an app by committing the code to the source control, activating the CI/CD pipeline. This action can be used as a doorstop to control output. While scans ensure code quality, a rule can be put in place here to ensure all code lines are related to a Story number, creating a highly granular and complete audit trail of all change releases.

Compliance rules for change management automated approval

You can build several rules to enforce compliance rules:

- Upon completion of functional tests, they are marked as Approved–either automated or manual but approval is obligatory
- All stories related to changes are marked as Approved by the Product Owner
- Every Story has a business or technical requirement that acts as a basis for the change

Follow-up event creation for outages

In case of an outage, an event or follow-up task is automatically generated. A newly created automated test case will detect and prevent the same outage from recurring.

The integrated technology solution described above stimulates collaboration between GRC practitioners and DevOps teams, with little to no need for retrospective interference (as part of traditional IT audit engagements) by GRC practitioners in key change management processes. Relevant change management data is automatically captured and presented in an understandable and logical manner in ServiceNow, where the integrated risk management modules are owned and managed by the GRC department. Standardized reporting capabilities in the integrated Risk Management solutions in ServiceNow can be easily configured, which can then be used and interpreted by the GRC practitioners.

3. Key take-aways

Harmonizing the worlds of GRC and DevOps may seem contradictory and difficult at first, but the first key to success is becoming aware of the paradox. The contradictions and conflicts that arise from trying to fuse the GRC and DevOps concepts just appear to be there. In fact, with the right guidance and capabilities, it will soon become evident that the two worlds can perfectly go hand in hand, strengthening and complementing one another.

Using the power of the integrated Enterprise Service Management Now platform, we can guide and support you in reforming GRC functions with respect to change management by adequately leveraging and integrating all DevOps activities. This will deliver great value with substantial efficiency gains, decreased risk of non-compliance and risk exposure–without impacting organizational agility–resulting in greater competitive advantages for your (global) enterprise.

About Plat4mation

Based in The Netherlands, Belgium, Germany, USA and India, we are an Elite ServiceNow Partner dedicated to delivering world-class products and services for the ServiceNow platform. We are driven to realize maximum value in the IT, employee, and customer workflow experiences for each one of our customers. We do this by providing a flawless customer experience utilizing our extensive expertise.

Since our inception in 2013, we have grown to more than 200 employees globally, and we are still growing strong! With a team of specialized consultants, we aim for the highest possible results while creating jaw-dropping experiences for our customers.

Authors

Carel Jansen - GRC Competence Lead
carel.jansen@plat4mation.com

Guus Hutschemaekers - Enterprise DevOps Lead
guus.hutschemaekers@plat4mation.com

Contact

Plat4mation BV
Arthur van Schendelstraat 650
3511 MJ Utrecht, The Netherlands

+31 30 760 26 70
info@plat4mation.com
www.plat4mation.com

ServiceNow Elite Partner