**Introduction and Objective**

**Overview of Central Automation Reporting**

Central automation reporting is a strategic initiative currently under development, aimed at enhancing visibility and control over the testing processes within our bank. This system is designed to centralize the tracking of automation test cases, facilitating the transition from manual to automated testing across various sprints. Once fully implemented, it will enable the monitoring of the progression and elaboration of test automation, ensuring a systematic and efficient testing framework.

**Importance in Our Organization**

The forthcoming implementation of the central automation reporting system, led by our executive leadership team, is pivotal for enhancing precision and efficiency in our testing operations. As we expand our automation efforts, the centralized supervision and analysis of test executions become crucial. This leadership-driven initiative is essential for enabling strategic decision-making within the organization. Upon its completion, the system will provide a comprehensive view of the automation landscape across different subdomains, facilitating better decisions and optimal resource allocation.

**Objectives of the System**

* **Tracking Automation Progress:** It will maintain a detailed record of the number and progress of automation test cases, helping us to understand the pace and scale of our transition from manual to automated testing.
* **Assessing Asset Health:** By evaluating the test execution status, the system will assess the health of each asset, offering insights into potential risks and areas for improvement.
* **Mandatory Integration:** Integration into this system is mandatory for all teams, as it will serve as the single source of truth to validate the automation scripting and execution efforts, thereby standardizing how data is reported across the organization.
* **Supporting Major Releases:** The system will offer crucial support for major enterprise releases by providing a clear picture of the cumulative health of all applications within a subdomain.
* **Guiding Decision-Making:** Leadership will be able to utilize the detailed analytics provided to make informed decisions about test coverage, identifying how much is currently available and determining the necessary levels for optimal performance and risk management.

**FAQ: Integration with the Central Automation Reporting System**

**Q: Why integrate with the central framework when we already generate standard test reports using TestNG, Allure, or other frameworks?**

**A:** While the reports generated by frameworks such as TestNG or Allure are beneficial for test case execution analysis and defect management within individual teams, their visibility and format are not consistent across the wider organization. These reports provide valuable insights at the team level but lack the scope needed for broader strategic decisions. Integration with the central automation reporting system is crucial as it consolidates and standardizes data across all teams, enabling leadership to drive strategic decisions around releases, automation coverage, and overall asset health at a higher level.

**Scope and Stakeholders**

**Scope of the Reporting System**

The central automation reporting system is a comprehensive initiative designed to encompass all applications and assets within the bank that undergo a testing lifecycle. This includes not only those assets currently equipped with automated test cases, whether created by testing or development teams, but also extends to those teams and assets that do not currently implement automation but are planning to transition towards automated testing practices. Initially, the focus will be on assets actively undergoing testing and which have integrated automated testing procedures.

**Key Stakeholders**

* **Automation Governance Authority (AGA) Team:** This team drives the initiative, overseeing the development, deployment, and evolution of the central automation reporting system. They are responsible for setting standards, ensuring compliance with the reporting requirements, and facilitating the integration of all QE teams into the system.
* **Quality Engineering (QE) Teams:** All QE teams across the bank are crucial stakeholders as they are both contributors to and primary users of the system. They are responsible for providing accurate and timely data regarding the automation test cases and outcomes for their respective assets. Additionally, QE teams are essential in providing feedback for improving and refining the reporting system.
* **Development Teams (as applicable):** For assets where development teams are involved in writing automated tests, these teams also become important stakeholders. Their role involves aligning their testing frameworks with the central reporting requirements to ensure consistency in data reporting across the bank.
* **IT and Technical Support Teams:** These teams play a supportive role by ensuring that the necessary technological infrastructure is in place and maintained. They assist in resolving any technical issues that arise during the integration and operation of the reporting system.

**Expansion Plans**

As the system matures, plans include scaling the reporting to include all teams within the bank that may transition to automation. This phased approach ensures that the system evolves in alignment with the bank’s overall automation strategy and capability developmentTop of Form

**Overview of Automation Framework Architecture**

The automation framework is structured into distinct categories to enhance the management and execution of automated tests across the organization. It is designed to cater to specific asset needs and leverage centralized capabilities provided by the EQE team.

**Components and Workflow**

**1. Categories of Automation Framework**

* **Asset-Specific Frameworks**: These are tailored to the particular requirements of individual assets, utilizing custom or in-house developed tools and techniques for automation.
* **autox**: This framework harnesses central framework capabilities developed and provided by the EQE team, allowing for standardized and efficient test management across multiple assets.

Overall Strategy

Teams employing either the autox framework or other in-house automation solutions, and managing tests through a Jenkins pipeline, can currently route their test execution reports to the AGA pipeline. This pipeline, integrated with a Report Parser, processes these reports and funnels the execution logs to Kibana for visualization. This system, effective in shakeout and SIT environments, supports various tools like Allure, TestNG, Mocha, and Serenity. The current setup, which sends limited execution data to Elasticsearch, is scheduled for a significant upgrade. A new plugin, featuring an ELK wrapper, is in development to facilitate direct log transmission to ELK and seamless dashboard integration. Upon release, this plugin will become integral to the autox framework, enhancing data handling and visualization capabilities. Teams not yet on autox are encouraged to start using the existing ELK wrapper to begin transitioning their data to Elasticsearch, preparing for a future shift to this more centralized and robust framework.

#### **Phased Integration and Upgrade Process**

* **Phase 1: Initial Setup and Integration**
  + **Integration via Jenkins Pipeline**: Teams currently using any form of automation framework utilize their Jenkins pipelines to send test execution reports to the AGA pipeline.
  + **Use of Report Parser**: These reports are processed by the Report Parser, which then transmits execution logs to Kibana for visualization. This method is operational and supports various reporting tools like Allure, TestNG, Mocha, and Serenity.
* **Phase 2: Feedback and Iterative Improvement**
  + **Operational Feedback**: As teams integrate and utilize the current system, they provide feedback which is used to refine the processes and prepare for the next integration phase.
* **Phase 3: Introduction of ELK Wrapper**
  + **Development and Testing of ELK Wrapper**: The new ELK wrapper, currently under development, is designed to facilitate direct data transfer to the ELK stack and integrate execution reports directly into dashboards.
  + **Pilot and Evaluation**: Initially, this will be made available to teams using the autox framework for testing and refinement based on real-world use.
* **Phase 4: Transition to ELK Wrapper**
  + **Phasing Out Report Parser**: As the ELK wrapper becomes fully operational and proves its efficacy, the initial solution utilizing the Report Parser will be phased out.
  + **Onboarding to ELK Wrapper**: All teams will gradually transition to using the ELK wrapper. This move will streamline data handling and visualization, offering more robust and integrated reporting capabilities.

#### **Supporting Components**

* **QTest Wrapper**: This component continues to monitor manual versus automated test case counts, providing visibility into automation coverage which is crucial for strategic planning.
* **Integration with Rally**: Planned integration with Rally in QTest to ensure requirement coverage is tracked consistently throughout development cycles.
* **APM Tool**: Utilized to delineate asset-level hierarchies and assist in test cataloging, the APM tool bridges gaps between development, business, and QA teams, fostering a unified approach to asset management.

### Conclusion

This structured, phased approach not only allows for seamless initial integration but also supports a strategic transition to more advanced, centralized reporting mechanisms. By outlining clear phases for adoption and upgrade, the framework ensures that all teams can progressively adapt to the enhanced system, ultimately leading to a cohesive and efficient automation environment across the organization.

### Engagement Model for Onboarding Teams to the AGA Pipeline

* Any team employing either the autox framework or another in-house automation framework, and managing tests through a Jenkins pipeline, has the capacity to funnel their test execution reports into the AGA pipeline. This pipeline integrates with a Report Parser that processes these reports and forwards the execution logs to Kibana for visualization. This setup is currently operational in the shakeout and SIT environments, capable of parsing reports from tools like Allure, TestNG, Mocha, and Serenity.
* The existing solution, which sends limited execution data to Elasticsearch, is slated for an upgrade. A new plugin incorporating an ELK wrapper is under development. This plugin will enable direct transmission of execution logs to ELK and integrate these reports into the dashboard. Once launched, this feature will become a core part of the autox framework capabilities. Teams not using autox will gradually migrate to this centralized framework as part of an overarching consolidation strategy. Meanwhile, teams can use the existing ELK wrapper to begin transferring their data to Elasticsearch.

**4. QTest Wrapper**

* The QTest wrapper is instrumental in capturing both the manual and the expected test cases eligible for automation. This data is then displayed on a dashboard to evaluate the coverage, helping quantify the manual versus automated testing within an asset. Furthermore, it is planned to integrate and link Rally requirements in QTest to standardize the process and enable the retrieval of requirement coverage throughout the development cycles.

**5. APM Tool**

* APM (Application Performance Management) is employed to delineate the asset-level hierarchy, crucial for test cataloging. This tool aids in establishing a common framework among development, business, and QA teams to track and manage test executions for specific assets.

**Conclusion**

This architecture not only facilitates efficient test management and execution but also enhances visibility and strategic decision-making across testing processes. By integrating advanced tools and processes, the framework supports a robust and scalable approach to automation, ensuring comprehensive coverage and alignment with organizational goals.

### Prerequisites for Onboarding to the Reporting Pipeline

Before beginning the integration process, teams must meet the following prerequisites to ensure a smooth transition to the reporting pipeline:

* **Automation Pack**: Teams must have an automation pack that is ready to be executed via a Jenkins pipeline. This pack should include all necessary scripts and configurations for automated testing.
* **Version Control System**: It is crucial that all automation code and related scripts are maintained in a version control system, such as GitHub, to facilitate collaboration and version tracking.
* **Basic Automation Framework**: Teams should have a basic automation framework in place. This framework does not necessarily need to have integrated CI/CD processes but should be structured to support eventual CI/CD implementation.

### FAQ for Teams Integrating with the Reporting Pipeline

#### Q: What if our team is only running tests locally and doesn’t use GitHub?

**A:** Teams running tests locally are advised to start integrating their code into a version control system like GitHub as part of the onboarding process. The reporting pipeline team can provide guidance and support to facilitate this transition.

#### Q: Our team has an automation framework but no CI/CD pipeline. Can we still integrate?

**A:** Yes, you can still begin the integration process. The reporting pipeline team will assist in developing and setting up a CI/CD pipeline tailored to your existing framework, ensuring that your automation practices are enhanced and aligned with continuous integration and deployment strategies.

#### Q: What support does the reporting pipeline team provide if we encounter issues during integration?

**A:** The reporting pipeline team offers comprehensive support throughout the integration process, including troubleshooting, technical advice, and regular check-ins to address any challenges that arise. Teams are encouraged to reach out with any specific issues or needs for additional guidance.

#### Q: How long does the integration process typically take?

**A:** The duration of the integration process can vary depending on the current state of your automation practices and the complexity of your systems. A customized onboarding plan will be provided, which will include estimated timelines and major milestones.

#### Q: Is it mandatory to integrate with the reporting pipeline?

**A:** Integration with the reporting pipeline is highly recommended as it enhances the overall effectiveness of testing processes and aligns with strategic goals for quality and efficiency. However, specific mandates may vary based on organizational directives and the criticality of the applications being tested.

Incorporating the metadata requirements for the regression suite into your documentation ensures that the data is organized and accessible within the organizational hierarchy. This will allow anyone to easily locate and interpret test results. Here's a structured approach to document these requirements:

### Regression Suite Metadata Requirements for Reporting Pipeline Integration

To effectively catalog and access application tests under the organizational hierarchy, specific metadata fields are required. These fields facilitate the classification and retrieval of test execution data across the reporting dashboard:

#### Essential Metadata Fields from APM

The following fields are critical as they establish the organizational context for the test assets and are derived from the Application Performance Management (APM) system. For detailed referencing, please see the documentation [here](https://chatgpt.com/c/b07737b7-ff11-41e7-af20-ca0f301ff0d6).

* **Business Unit**: Identifies the business segment within the organization to which the application belongs.
* **Domain**: Specifies the broader category or functional area that the application serves within the business unit.
* **Subdomain**: Further refines the classification within a domain, detailing a more specific area of functionality or service.
* **AppID**: A unique identifier for the application, crucial for tracking and referencing specific tests.

#### Additional Descriptive Fields

To further enhance the categorization and management of the regression pipelines, additional fields are necessary. These fields help in tagging and contextualizing the test executions:

* **Suite Name**: The name of the test suite, which provides a quick reference to the type of tests included.
* **Suite Path**: The directory path or location of the test suite within the repository or test management tool, aiding in navigation and organization.
* **QTest Project Name**: The name of the project within QTest where the test results are stored and managed, essential for aligning test executions with project management activities.

#### Collaboration and Data Integrity

* **Documentation and Discussion**: All metadata fields need to be thoroughly documented and discussed with the reporting pipeline team (formerly AGA team) to ensure clarity and mutual understanding of the data requirements.
* **Data Integrity**: Maintaining the accuracy and consistency of this metadata is vital. Regular audits and validations should be conducted in collaboration with the reporting pipeline team to ensure that the metadata correctly reflects the test executions and asset categorization.