ProcellaRX
PRESCRIPTION FOR SOFTWARE QUALITY

Adopting OpenText Aviator in Life Science Organizations

WHITE PAPER | JUNE 2025

The life science industry faces stringent regulatory requirements and demands meticulous testing processes. OpenText Aviator, with its AI-powered testing capabilities, offers a compelling solution to enhance testing efficiency, accuracy, and compliance. This white paper provides strategic and technical recommendations for life science organizations looking to adopt OpenText Aviator within their testing workflows.

Strategic and technical recommendations by:

Dori Gonzalez-Acevedo
CEO ProcellaRX

DISCLOSURE STATEMENT

This document outlines ProcellaRX's perspective on the responsible and strategic adoption of Artificial Intelligence (AI) within the life sciences industry. The insights and recommendations presented herein reflect our views on industry-wide practices and governance models; they are not indicative of current internal implementations or product deployments by ProcellaRX.

While informed by regulatory frameworks, industry standards, and published research, this white paper is not intended as regulatory guidance or a substitute for independent compliance advice. References to frameworks such as GAMP, ISO, or FDA/EMA initiatives are used to illustrate best-practice alignment, not to imply endorsement or conformance.

ProcellaRX offers this white paper as part of our broader commitment to advancing ethical, transparent, and effective innovation across healthcare and life sciences.

Table of Contents

About ProcellaRX

At ProcellaRX, LLC, we provide strategic product advisement and digital validation consulting services, helping pharmaceutical, biotechnology, and medical device companies, product vendors, and managed service providers drive innovation while ensuring regulatory compliance. Specializing in Computerized System Validation (CSV), Computer Software Assurance (CSA), and quality management solutions, we help organizations optimize compliance, efficiency, and technology adoption.

Our expert team brings a forward-thinking, risk-based approach to regulatory compliance, seamlessly aligning business objectives with industry standards. Through strategic consulting, automation enablement, and digital validation solutions, we accelerate digital transformation initiatives, enhance software quality testing, and implement scalable, future-ready validation frameworks.

By leveraging deep industry expertise and innovative methodologies, we enable our clients to streamline compliance processes, maximize operational efficiency, and stay ahead of regulatory expectations. Whether integrating advanced testing automation, enhancing AI-driven validation strategies, or implementing next-generation quality management systems, ProcellaRX empowers organizations to transform today's applications into future-ready solutions that drive industry progress.

LEADERSHIP TEAM

Dori Gonzalez-Acevedo
CEO

Staci Spencer
VP of Quality

Grace Shaw
VP of Global Compliance Strategy and Solutions

Mission and Vision

At ProcellaRX, our Mission and Vision drive our dedication to reshaping compliance and advancing industry standards. We combine deep expertise with forward-thinking solutions to simplify regulatory complexities, enhance operational efficiency, and equip organizations to lead with confidence in a rapidly evolving landscape.

MISSION

At ProcellaRX, we go beyond compliance—transforming today's applications into future-ready solutions that align with the industry's evolving needs. Through strategic innovation and expert guidance, we enable seamless regulatory integration while driving efficiency, quality, and sustainable progress.

VISION

We see a future where compliance is a catalyst for innovation, not a barrier. By bridging regulatory expertise with transformative solutions, we empower organizations to adapt with agility, embrace industry advancements, and set new standards for growth, efficiency, and leadership.

ProcellaRX
PRESCRIPTION FOR SOFTWARE QUALITY

Executive Summary

Adopting OpenText Aviator offers life science organizations a powerful means to transform their testing processes. By following these strategic and technical recommendations, organizations can leverage Aviator's AI capabilities to enhance testing efficiency, accuracy, and compliance, ultimately accelerating the delivery of high-quality, safe, and effective products.

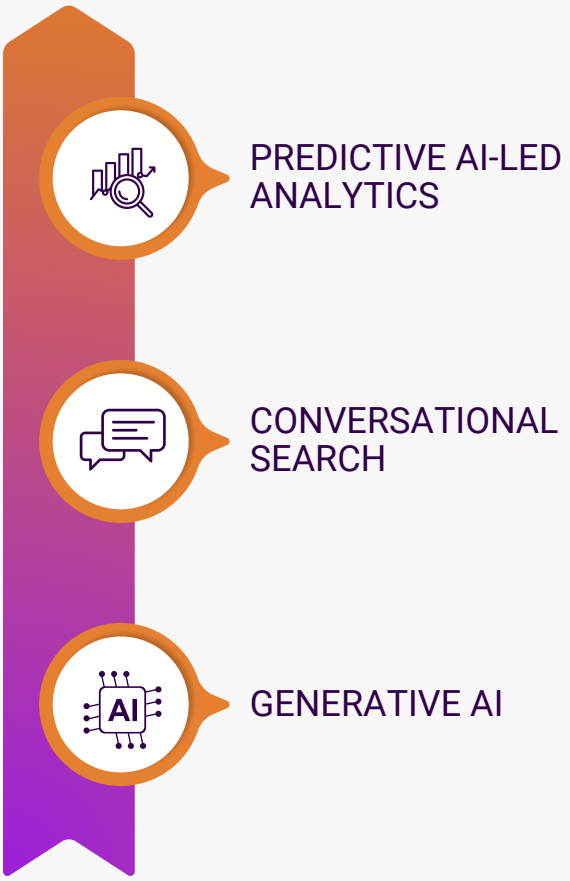
- Efficiency
- Accuracy
- Compliance

Introduction

Life Science Organizations are preparing to integrate OpenText Aviator – a suite of AI and generative AI capabilities – into its cloud-based OpenText ecosystem (ALM, UFT, LoadRunner) when it becomes available in 2026. This presents an opportunity to enhance software testing and QA workflows with AI-driven automation, predictive analytics, and intelligent assistance (OpenText offers broad AI capabilities for the enterprise with Aviator). OpenText Aviator brings predictive AI-led analytics, conversational search, and generative AI into daily workflows (AI for Business & Enterprise AI Platform | OpenText), and OpenText's roadmap shows plans to embed these Aviator features across DevOps and testing tools (OpenText DevOps Cloud Innovations | OpenText). The following recommendations outline how Life Science Organizations can strategically leverage Aviator for testing, integrate it into the SDLC, and ensure compliance and readiness in a GxP-regulated environment.

OpenText Aviator presents an opportunity to enhance software testing and QA workflows with AI-driven automation, predictive analytics, and intelligent assistance.

OpenText Aviator brings



Strategic Recommendations

Define Clear Objectives and Scope

Define the scope of the initial Aviator implementation. Start with a pilot project focusing on a specific application or process to demonstrate value and gather learnings. Avoid trying to implement Aviator across the entire organization at once.

Before implementing OpenText Aviator, clearly define the objectives you aim to achieve. Consider the following:

- **Improved test coverage:** Identify specific areas where Aviator can enhance coverage, such as regression testing or API testing.
- **Reduced testing time:** Set realistic targets for time savings through automation and AI-driven analysis.
- **Enhanced compliance:** Determine how Aviator can support compliance with regulations like FDA 21 CFR Part 11.

Establish a Center of Excellence (CoE)

Create a CoE to champion the adoption of Aviator and drive best practices. The CoE should include representatives from testing, development, regulatory affairs, and IT.
Key responsibilities include:

- **Developing Aviator usage guidelines:** Define standards for test automation, data management, and reporting.
- **Providing training and support:** Equip teams with the skills and knowledge needed to effectively use Aviator.
- **Monitoring and measuring results:** Track key metrics to assess the impact of Aviator on testing performance.
- **Promoting knowledge sharing:** Facilitate collaboration and knowledge transfer across teams.

Foster a Culture of Automation

Successful Aviator adoption requires a shift towards a culture of automation. Encourage teams to:

- **Embrace test automation:** Prioritize automation for repetitive and time-consuming tasks.
- **Develop reusable test assets:** Create modular test components that can be easily reused across multiple projects.
- **Collaborate effectively:** Foster collaboration between testers, developers, and business analysts to ensure clear requirements and testable code.
- **Continuously improve:** Regularly review and optimize testing processes to maximize the benefits of Aviator.

Technical Recommendations

Integrate Aviator into Existing Toolchains

OpenText Aviator integrates with various testing tools and development environments. Ensure seamless integration with your existing toolchain, including:

- **Test management systems:** Integrate Aviator with your test management system (e.g., Micro Focus ALM, Jira) to streamline test planning, execution, and reporting.
- **CI/CD pipelines:** Incorporate Aviator into your CI/CD pipelines to automate testing as part of the software development lifecycle.
- **Source control systems:** Integrate Aviator with source control systems (e.g., Git) to manage test scripts and data.
- **Defect tracking systems:** Connect Aviator with your defect tracking system to automatically report and track defects.

Leverage Aviator's AI Capabilities

Aviator's AI capabilities can significantly enhance testing effectiveness. Key areas to leverage include:

- **AI-powered test generation:** Use Aviator to automatically generate test cases based on requirements, code, and historical data.
- **Smart test execution:** Utilize Aviator's AI to dynamically adjust test execution based on risk and impact.
- **Intelligent defect analysis:** Leverage Aviator's AI to identify root causes of defects and prioritize remediation efforts.
- **Self-healing tests:** Enable Aviator's self-healing capabilities to automatically update tests when UI changes occur.

Implement Robust Data Management Practices

Effective data management is crucial for successful Aviator implementation. Consider the following:

- **Test data generation:** Create realistic and representative test data sets using Aviator's data generation capabilities.
- **Data masking:** Mask sensitive data to protect patient privacy and comply with data protection regulations.
- **Data versioning:** Implement data versioning to ensure that test data is consistent and reproducible.
- **Data governance:** Establish clear data governance policies to ensure data quality and integrity.

Use Cases for AI in Testing

OpenText Aviator can transform testing by generating test assets, accelerating automation, and providing intelligent insights. Key use cases include:

Automated Test Case Generation

AI analyzes requirements to generate comprehensive and diverse test cases, ensuring broader coverage and identifying gaps human testers might miss.

Test Automation & Script Generation

Automatically converts manual test cases, natural language, or user observations into executable, codeless test scripts, significantly accelerating automation development.

Documentation Summarization & Analysis

AI summarizes and analyzes extensive testing documentation, reports, and project artifacts, offering concise insights and improvement suggestions.

Test Optimization & Defect Detection

Utilizes AI to analyze test executions and defect patterns, optimizing test suites, prioritizing high-risk areas, and detecting anomalies for early defect identification.

Predictive Performance Analysis

Applies AI to performance data for predicting issues, forecasting system behavior under load, and aiding root cause analysis to ensure strong application performance and scalability.

USE CASES FOR AI IN TESTING

Automated Test Case Generation

Aviator's generative AI can analyze requirements or user stories and produce diverse test cases; filling coverage gaps that human testers might overlook. By leveraging its deep understanding of language and code, the AI can quickly generate high-coverage test scenarios based on functional descriptions (10 Ways Generative AI Will Revolutionize DevOps | OpenText Community). This saves time and ensures more comprehensive testing coverage, reducing the risk of defect escape (10 Ways Generative AI Will Revolutionize DevOps | OpenText Community).

TAKEAWAYS

- **Automates test case generation directly from requirements.**
- **Enhances test coverage by identifying gaps in manual testing.**
- **Achieves significant time and resource efficiencies.**
- **Mitigates the risk of defect escape.**

Test Automation & Script Generation

The platform will enable automatic generation of test scripts and automation from existing assets. For example, OpenText's ValueEdge (ALM) has introduced an Aviator feature that converts manual test cases into automated, codeless tests using generative AI (Discover the Future of Functional Testing with ValueEdge 24.3 | OpenText Community). This means Life Science Organizations can take a manually written test procedure and have Aviator instantly produce an automated UFT script or model-based test, significantly accelerating automation development.

In the future, Aviator may even generate test scripts from natural language descriptions or from observing user flows (e.g. converting videos of manual steps into test scripts (OpenText DevOps Cloud Innovations | OpenText)). These capabilities will boost testing efficiency and reliability by reducing human effort in writing automation and by continually learning to maintain scripts as the application changes.

TAKEAWAYS

- **Converts existing manual test assets into automated scripts.**
- **Significantly accelerates automation development timelines.**
- **Reduces the manual effort required for script authoring.**
- **Enables future script generation from natural language or user flows.**

USE CASES FOR AI IN TESTING

Documentation Summarization & Analysis

Aviator's AI-powered assistants can summarize and analyze testing documentation and results. OpenText's "Smart Assistant" is a generative AI virtual assistant that can analyze project artifacts (like requirements, defects, test results) and suggest improvements (Get started). Life Science Organizations can use such capabilities to auto-summarize test execution reports, release notes, or requirements into concise insights for decision-makers. For example, the AI could generate a summary of a large validation protocol or aggregate key points from multiple test incident reports.

Additionally, OpenText's Content Aviator already offers chat-based search and content summarization across enterprise documents (OpenText™ Content Aviator) – a similar feature could be applied to testing knowledge (e.g. quickly querying past test evidence or summarizing a lengthy validation document). This saves time for teams and leadership in understanding large bodies of QA documentation.

TAKEAWAYS

- **Automates the analysis and summarization of test artifacts.**
- **Distills lengthy reports into concise insights for decision-makers.**
- **Enables rapid, conversational search across all testing documentation.**
- **Accelerates comprehension of complex QA and validation data.**

USE CASES FOR AI IN TESTING

Test Optimization & Defect Detection

Generative AI and machine learning can help optimize the test suite and detect problem areas faster. Aviator will be able to analyze past test executions, code changes, and defect patterns to identify high-risk areas of the application and recommend where to focus testing. For instance, AI pattern recognition can spot anomalies or patterns in test results that humans might miss (10 Ways Generative AI Will Revolutionize DevOps | OpenText Community). It could automatically detect areas with frequent failures or flaky tests and suggest additional tests or code inspections.

Aviator's capabilities include enhanced risk prioritization – identifying potential risks and prioritizing tests based on severity and impact (10 Ways Generative AI Will Revolutionize DevOps | OpenText Community). This means Life Science Organizations' teams can utilize AI to perform risk-based test optimization, ensuring critical paths are thoroughly tested while redundant tests are minimized. Early AI-driven anomaly detection in test results will help catch defects sooner (10 Ways Generative AI Will Revolutionize DevOps | OpenText Community), improving product quality.

TAKEAWAYS

- **Analyzes historical data to identify and target high-risk areas.**
- **Prioritizes test execution based on calculated risk and business impact.**
- **Optimizes test suites by focusing on critical paths and reducing redundancy.**
- **Enables early defect detection through advanced AI anomaly analysis.**

USE CASES FOR AI IN TESTING

Predictive Performance Analysis

OpenText's roadmap indicates Aviator will support performance engineering analysis and predictive insights (OpenText DevOps Cloud Innovations | OpenText). In practice, this means AI can be applied to performance testing (LoadRunner results and APM data) to predict performance issues and capacity needs. For example, by learning from historical performance test metrics, Aviator could forecast how the system will behave under higher loads or identify trends (memory usage patterns, response time degradation) that predict a future bottleneck. These AI-driven insights would allow Life Science Organizations to proactively tune systems before performance issues occur in production.

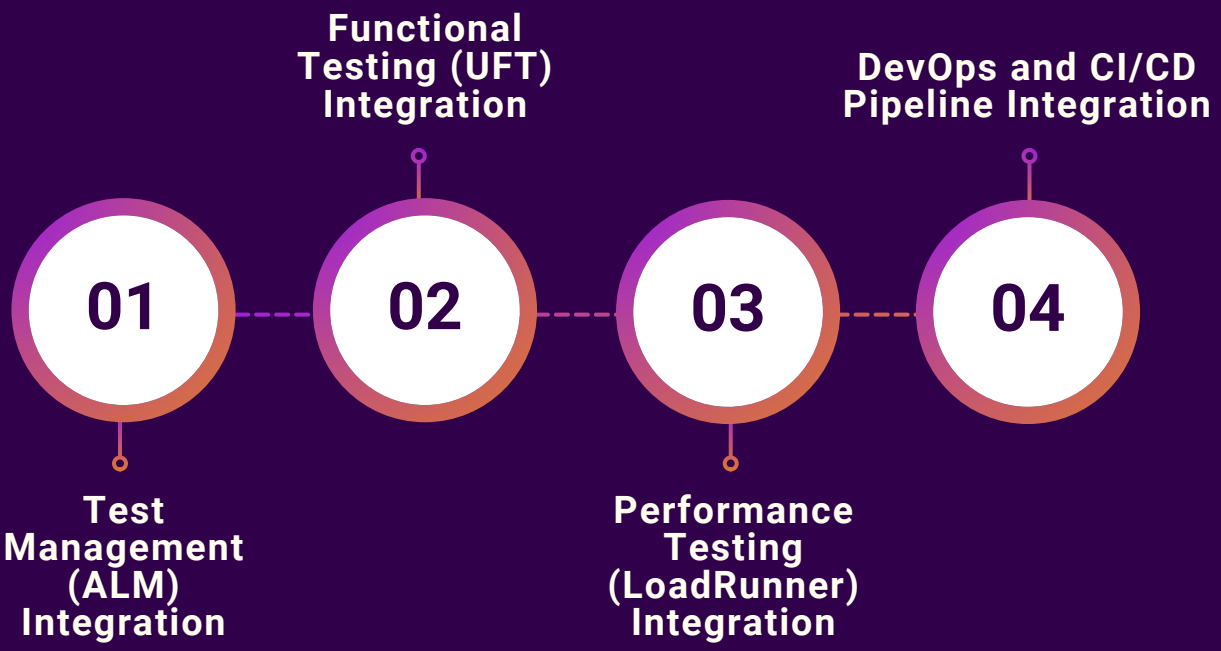
Additionally, AI can assist in root cause analysis of performance problems, correlating metrics and logs faster than a human analyst. Predictive analytics in Aviator (already used to predict project timelines (Get started)) could similarly be applied to predict whether an application will meet its performance SLAs based on current test data. This use case will help Life Science Organizations ensure robust performance and scalability of its applications, with AI providing an early warning system for potential performance constraints.

TAKEAWAYS

- **Analyzes historical data to identify and target high-risk areas.**
- **Prioritizes test execution based on calculated risk and business impact.**
- **Optimizes test suites by focusing on critical paths and reducing redundancy.**
- **Enables early defect detection through advanced AI anomaly analysis.**

Integration Strategies for Aviator in Life Science Organizations' SDLC

To realize these benefits, Life Science Organizations should embed OpenText Aviator into its software development lifecycle (SDLC) and toolchains. Integration points and strategies include:



INTEGRATION STRATEGIES

Test Management (ALM) Integration

Integrate Aviator directly into OpenText ALM (Application Lifecycle Management) for seamless test planning and design. OpenText plans to “extend OpenText Aviator to ALM for test generation” (OpenText DevOps Cloud Innovations | OpenText), which suggests that in 2026 the SaaS ALM interface will allow users to invoke AI to generate test cases from requirements or user stories. Life Science Organizations should take advantage of this by enabling the Aviator chatbot or assistant within ALM that can interact with testers. For example, a tester could select a requirement and ask the Aviator assistant to “Generate test cases for this requirement,” after which the suggested tests can be reviewed and added to ALM. Aviator could also help elaborate backlog items or user stories (as indicated by “backlog items elaboration” in the roadmap) (OpenText DevOps Cloud Innovations | OpenText) – meaning it can flesh out acceptance criteria or clarify requirements, leading to better-defined test conditions.

Embedding the AI in ALM ensures it fits naturally into the test design workflow, and testers/business analysts can use it during requirements review and test case creation. Life Science Organizations’ ALM administrators should plan to enable this integration and possibly configure it to enforce that any AI-generated test is flagged for review before approval (as part of compliance – discussed later).

TAKEAWAYS

- **Generates test cases directly from requirements within the ALM interface.**
- **Elaborates user stories to refine acceptance criteria and test conditions.**
- **Embeds AI assistance seamlessly into the native test design workflow.**
- **Enables workflows requiring human review of all AI-generated assets.**

INTEGRATION STRATEGIES

Functional Testing (UFT) Integration

Leverage Aviator within UFT and the functional test automation pipeline. OpenText's DevOps Aviator for functional testing will allow features like one-click conversion of manual tests to automated tests (Discover the Future of Functional Testing with ValueEdge 24.3 | OpenText Community) and even generation of test scripts from natural language. Life Science Organizations should integrate these capabilities by updating their UFT One / UFT Developer tools to the versions supporting Aviator (likely OpenText ValueEdge Functional Test modules). Test automation engineers can then use the AI to "generate automated codeless tests" from a manual test specification (Get started). In practice, a manual test case stored in ALM could be fed to Aviator, which produces a draft automated script (for example, a UFT Developer script or a model for UFT One's AI-based testing). This script would use AI-based object recognition (capabilities already present in UFT) to interact with the application (AI Powered Functional Testing and Test Automation Software).

Life Science Organizations should embed this into the workflow by having automation engineers review and refine AI-generated scripts rather than coding from scratch. Additionally, Aviator's AI object detection and smart verification point features will help with test maintenance – it can intelligently identify UI elements and adapt to changes, reducing false failures when apps change (OpenText DevOps Cloud Innovations | OpenText). Integration into UFT means testers will have AI assistance directly in their IDE or testing tool, akin to a "Copilot" for test automation.

TAKEAWAYS

- **Generates automated test scripts directly from manual cases or natural language.**
- **Simplifies test maintenance by intelligently adapting to application changes.**
- **Shifts the automation engineer's role from initial authoring to AI-assisted review.**
- **Provides integrated, "Copilot-style" assistance directly within the UFT environment.**

ProcellaRX
PRESCRIPTION FOR SOFTWARE QUALITY

INTEGRATION STRATEGIES

Performance Testing (LoadRunner) Integration

Incorporate Aviator into performance testing and engineering activities. While specifics are still emerging, OpenText's roadmap lists "Performance Engineering Analysis" as a key AI-driven insight area (OpenText DevOps Cloud Innovations | OpenText). Life Science Organizations should plan for Aviator to be connected with LoadRunner (or LoadRunner Cloud) results and APM metrics. This could take the form of an AI analysis assistant within LoadRunner Analysis that automatically interprets test results: for example, after a performance test run, Aviator could produce a summary highlighting potential bottlenecks (CPU spikes, slow queries) and even suggest likely root causes (e.g. "possible database indexing issue causing slow transaction X").

If Life Science Organizations use monitoring tools (like AppDynamics, Dynatrace, etc.), integrating those data streams with Aviator's analysis could provide a holistic view. Additionally, predictive models could be integrated into performance test pipelines – e.g., an Aviator service that, once a test is done, predicts how many users the system can handle before response time violates thresholds. To embed this in the SDLC, Life Science Organizations' performance engineers can include an "AI analysis" step in the test workflow (possibly via an API if provided by OpenText). Over time, as Aviator learns from multiple test runs, it will enhance the performance optimization feedback loop. Ensuring this integration might involve working with OpenText services to connect LoadRunner results repository to the Aviator analytics engine.

TAKEAWAYS

- **Automates performance analysis to identify bottlenecks and suggest root causes.**
- **Predicts system capacity and potential violations of performance thresholds**
- **Integrates LoadRunner and APM data to create holistic performance insights.**
- **Continuously enhances analysis by learning from successive test executions.**

ProcellaRX
PRESCRIPTION FOR SOFTWARE QUALITY

INTEGRATION STRATEGIES

DevOps and CI/CD Pipeline Integration

Plan for Aviator to be available in Life Science Organizations' CI/CD and DevOps toolchain. OpenText Cloud Editions are delivering new AI features every quarter, and they emphasize out-of-the-box integration with DevOps tools (e.g. Jenkins, GitHub Actions, GitLab integrations are on the roadmap) (OpenText DevOps Cloud Innovations | OpenText). Life Science Organizations should integrate Aviator triggers in the CI/CD pipeline – for example, after a build is deployed to a test environment, have Aviator automatically generate a set of smoke tests or regression tests to execute. If using Jenkins, a plugin or API call could invoke Aviator's "suggest tests" feature on each new user story completion. Similarly, Aviator's Smart Assistant might integrate with requirements management to ensure traceability (e.g., via webhooks when a new requirement is added, generate tests and update ALM).

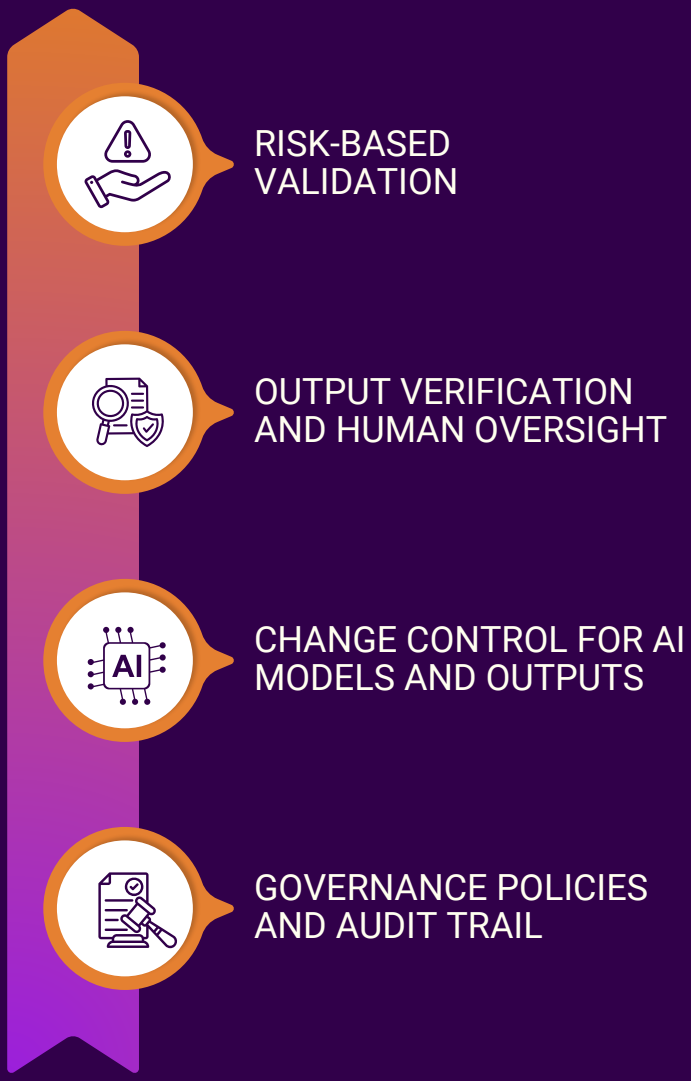
Integration with DevOps means treating Aviator as an "AI service" in the toolchain: pipeline steps for AI-driven test generation, results analysis, and even code reviews. OpenText's DevOps Aviator also suggests an interactive chat with any entity (potentially a chatOps integration) (OpenText DevOps Cloud Innovations | OpenText) – Life Science Organizations could enable a chat interface (for example, in Microsoft Teams or Slack) where engineers query the AI about the status of testing or ask it to run certain analyses. Embracing these integration points will embed AI deeply into the SDLC workflow, making it a natural part of development and testing rather than a separate siloed tool.

TAKEAWAYS

- **Integrates into the CI/CD pipeline as a callable "AI service."**
- **Automates test generation and execution based on pipeline triggers.**
- **Enables interactive control and querying of the test process via ChatOps.**
- **Maintains continuous traceability between requirements and test assets.**

Validation and Governance in a GxP-Regulated Environment

Adopting generative AI in a GxP context (where software quality can impact product safety and efficacy) requires careful validation, change control, and governance. Life Science Organizations must treat OpenText Aviator as a GxP-impacting system component and establish controls to ensure its outputs are reliable and compliant:



VALIDATION AND GOVERNANCE

Risk-Based Validation

Even though Aviator involves non-deterministic AI, it must be validated for its intended use in the testing process. The FDA's 21 CFR 820 (quality system regulation) requires that software used in production or quality systems is validated according to an established protocol (Decoding The FDAs Draft Guidance On Computer Software Assurance For Medical Devices BioPharma). Life Science Organizations should apply a Computer System Validation (CSV) or Computer Software Assurance (CSA) approach focusing on critical quality risks. Under the newer CSA guidance, FDA encourages leveraging vendor testing and documentation, and then performing targeted scripted and unscripted testing to cover high-risk functions (Decoding The FDAs Draft Guidance On Computer Software Assurance For Medical Devices BioPharma).

For Aviator, this means Life Science Organizations can rely on OpenText's internal validation for standard functionality but must validate critical use cases where AI decisions could impact GxP outcomes. For example, if Aviator generates test cases that will be used as evidence for product release, Life Science Organizations should verify (perhaps through sample comparisons or parallel manual case design) that the AI-generated tests meet requirements and do not omit critical scenarios. This validation should be documented (test plans, test results) to demonstrate that Aviator performs as intended in Life Science Organizations' environment.

TAKEAWAYS

- **Requires formal validation of the AI tool for its specific intended use.**
- **Advocates a risk-based validation approach, such as Computer Software Assurance (CSA).**
- **Leverages vendor testing for standard, low-risk tool functionality.**
- **Focuses internal validation efforts on critical, GxP-impacting AI outputs.**

VALIDATION AND GOVERNANCE

Output Verification and Human Oversight

As a governance rule, human experts should review and approve AI-generated outputs that impact regulated deliverables. In practice, when Aviator generates a test case or summary that will become part of a validation package, a qualified tester or QA person should verify its accuracy and completeness before it is accepted. This “human-in-the-loop” oversight is a critical guardrail in a GxP setting to ensure that any AI errors do not directly propagate into the validated state of the system.

Life Science Organizations can institute a procedure where AI-produced test scripts or reports are clearly flagged and cannot be marked as “approved” in ALM until a human reviewer signs off. By doing so, Aviator becomes a powerful assistant rather than an autonomous source of GxP records. This approach aligns with industry best practices that stress AI augmentation with human judgment to maintain quality and compliance (Artificial Intelligence Governance in GxP Environments | Pharmaceutical Engineering). It will also help during audits: Life Science Organizations can demonstrate that while AI suggested a result, a human accountable person approved it, thus meeting regulatory expectations for decision-making.

TAKEAWAYS

- **Mandates human expert review and approval for all regulated AI outputs.**
- **Implements "human-in-the-loop" oversight as a critical quality guardrail.**
- **Establishes the AI as a powerful assistant, not an autonomous decision-maker.**
- **Ensures clear human accountability for all final GxP records during audits.**

VALIDATION AND GOVERNANCE

Change Control for AI Models and Outputs

OpenText Aviator's AI models will likely evolve (the vendor may update LLMs or algorithms over time). In a regulated environment, these changes must be assessed under Life Science Organizations' change control system. Life Science Organizations should require OpenText to notify companies of significant updates to the Aviator service (new model version, new major feature) and treat those like system changes. Under change control, perform an impact assessment: e.g., if a new version of the generative model is introduced in 2027, does Life Science Organizations need to re-test some functionality to ensure outputs are consistent with the previous version? Using CSA principles, critical thinking should be applied – focus re-validation on areas of higher risk.

For example, if an update claims to improve test generation, Life Science Organizations could sample a few requirements and have the new AI generate tests, comparing them with previous outputs for equivalence or improvements. If differences are found, assess if any could adversely affect quality (perhaps an important test case is no longer generated). Maintain documentation of these assessments and any re-validation actions. Additionally, configuration control should be applied to any prompts or templates Life Science Organizations develops for Aviator. If Life Science Organizations create specific prompt scripts or AI workflows (e.g., a defined query to generate a particular type of report), those should be version-controlled and change-controlled, since they define how the AI is used. Governance should also ensure that any time an AI output is saved as a GxP record, it is locked down and cannot dynamically change later (to preserve data integrity).

TAKEAWAYS

- **Subjects all vendor-supplied AI model updates to formal change control.**
- **Requires a risk-based impact assessment to guide re-validation efforts.**
- **Applies version control to all internally developed prompts and templates.**
- **Ensures data integrity by rendering final GxP records immutable.**

VALIDATION AND GOVERNANCE

Governance Policies and Audit Trail

Implement an AI governance framework that defines acceptable use and monitoring of Aviator. This should include SOPs or working instructions on using generative AI in validation activities – detailing steps users must follow (like the review process above, or restrictions on inputs). The system should also have an audit trail for AI usage: for instance, logging the prompt given to Aviator and the output it provided, timestamped, and linked to the tester who invoked it. This creates traceability, so if an issue arises (e.g., an AI-suggested test missed a defect), Life Science Organizations can analyze the chain of events.

Governance should address data privacy and security as well – ensuring that any data sent to Aviator (even if via a secure cloud) does not include sensitive patient or product data beyond what's necessary. If needed, mask or avoid using production data in prompts (for example, describe scenarios abstractly rather than using real patient records). The governance framework can include AI-specific quality checks – for instance, periodic review of a sample of AI outputs to ensure they remain of high quality. By establishing these guardrails, Life Science Organizations addresses the concerns of bias, transparency, and control that come with AI, as recommended by industry experts (Artificial Intelligence Governance in GxP Environments | Pharmaceutical Engineering). In essence, treat Aviator as a semi-autonomous junior analyst that needs oversight and whose "work" must be documented and audited like any other regulated process.

TAKEAWAYS

- **Establishes a formal governance framework with SOPs for acceptable use.**
- **Mandates a complete, user-attributable audit trail for all AI interactions.**
- **Enforces strict data privacy controls to protect sensitive information.**
- **Institutes periodic quality reviews to monitor AI output and performance.**

Compliance Considerations: FDA CSA and Good Machine Learning Practice (GMLP)

In deploying Aviator, Life Science Organizations should ensure alignment with the FDA's Computer Software Assurance (CSA) guidance and emerging Good Machine Learning Practice (GMLP) principles:

Aligning with CSA (Computer Software Assurance)

Applying Good Machine Learning Practice (GMLP)

COMPLIANCE CONSIDERATIONS

Aligning with CSA (Computer Software Assurance)

The FDA's draft guidance on CSA (for production and quality system software) advocates a risk-based, critical thinking approach to validation, moving away from exhaustive documentation toward assurance focused on high-risk outcomes (Decoding The FDAs Draft Guidance On Computer Software Assurance For Medical Devices BioPharma) (Decoding The FDAs Draft Guidance On Computer Software Assurance For Medical Devices BioPharma). For Aviator, Life Science Organizations should document a risk assessment of its intended uses: e.g., using AI to generate tests has a risk of missing critical tests; using AI to summarize documents has a risk of misinterpretation. Mitigation for these risks (as discussed) is primarily human review and targeted testing of the AI.

Under CSA, Life Science Organizations can leverage OpenText's own testing of Aviator and avoid duplicating effort for low-risk functions (Decoding The FDAs Draft Guidance On Computer Software Assurance For Medical Devices BioPharma). For instance, basic functionality like the UI of the Aviator chat or non-GxP features can rely on vendor qualification.

For Aviator, Life Science Organizations should document a risk assessment of its intended uses.

Life Science Organizations' assurance activities should then concentrate on what's novel: the quality of AI outputs in Life Science Organizations' context. Unscripted testing (exploratory trials of the AI on various scenarios) can be a part of this assurance, aligning with CSA's acceptance of unscripted testing to demonstrate fitness for use (Decoding The FDAs Draft Guidance On Computer Software Assurance For Medical Devices BioPharma). The approach should be documented in a validation plan that references the CSA principles – focusing on critical quality attributes (e.g., test correctness) rather than 100% of requirements. This will satisfy FDA expectations that Life Science Organizations is ensuring the tool is fit for its intended purpose in a quality system without placing an undue validation burden that stifles innovation.

COMPLIANCE CONSIDERATIONS

Applying Good Machine Learning Practice (GMLP)

Although GMLP guidelines are geared toward medical devices using AI, the principles are instructive for any GxP use of AI. FDA and other regulators have identified 10 guiding principles for GMLP to promote safe, effective, and high-quality AI (Good Machine Learning Practice for Medical Device Development: Guiding Principles | FDA) (Good Machine Learning Practice for Medical Device Development: Guiding Principles | FDA). Life Science Organizations should incorporate these principles in their use of Aviator. Key considerations include:

- **Quality and relevance of data:** Ensure that any data used to fine-tune or prompt Aviator (such as Life Science Organizations' own test cases or product requirements) is accurate and representative, to avoid AI bias.
- **Model transparency:** Life Science Organizations should seek as much understanding as possible from OpenText about how Aviator's models make decisions or what data they were trained on. While the internal algorithms might be a "black box," OpenText might provide assurance statements on data governance. Life Science Organizations' governance could include requiring vendor documentation on their AI development practices (e.g., bias testing, algorithm validation).
- **Monitoring performance:** Once deployed, continuously monitor the performance of the AI in Life Science Organizations' context – for example, track if the test cases generated by Aviator are catching defects or if any significant misses occur.
- If the AI's suggestions are declining in quality or drifting, that might indicate a need for retraining or vendor fix, aligning with the GMLP concept of monitoring for performance degradation.
- **Accountability and expertise:** GMLP emphasizes leveraging multidisciplinary expertise. Life Science Organizations should involve IT, QA, and domain experts in evaluating Aviator's output – this cross-functional oversight ensures the AI's suggestions are assessed from technical and process perspectives (Artificial Intelligence Governance in GxP Environments | Pharmaceutical Engineering).
- **Bias and fairness:** While less directly relevant to test cases than to patient data, Life Science Organizations should be mindful that any AI can have blind spots. Perhaps ensure the AI is tested on various project types (e.g., simple vs. complex systems) to see if it consistently performs. If any bias in output is noticed (like always focusing on certain modules), that feedback should be given to the vendor.

In summary, by following GMLP guiding principles (data quality, transparency, performance monitoring, etc.), Life Science Organizations will demonstrate that it is using Aviator responsibly and in line with regulatory expectations for AI quality. This might be documented in an internal guideline or addendum to the validation plan that explicitly maps how Life Science Organizations' implementation of Aviator addresses each of the GMLP principles to the extent applicable (even if Aviator is not part of a medical device, it's good practice to show commitment to these quality principles).

Organizational Readiness Requirements

Successfully adopting Aviator will not only be a technical endeavor but also an organizational change. Life Science Organizations should prepare their people, policies, and processes to maximize benefits and minimize risks:

Training & Upskilling the Team

Train all teams on OpenText Aviator's features and generative AI basics to build comfort and proficiency.

Prompt Engineering and Testing

Optimize AI output by practicing "prompt engineering" and creating a library of effective prompts.

Oversight and AI Stewardship

Establish clear oversight for AI usage to monitor compliance and keep leadership informed.

LLM Usage Policy and Data Governance

Develop a strict policy for LLM usage, covering data input, use cases, and security.

Change Management and Culture

Prepare the organization culturally for AI adoption through communication, pilots, and process updates.

ORGANIZATIONAL READINESS REQUIREMENTS

Training & Upskilling the Team

Invest in comprehensive training for all stakeholders (testers, QA leads, developers, IT compliance) on the capabilities and limitations of OpenText Aviator. This training should cover how to use the new AI features (e.g. how to prompt the Smart Assistant, how to review AI-generated tests) as well as foundational knowledge about generative AI. The goal is to make teams comfortable with AI as a collaborator. For example, test engineers should be trained in interpreting and refining Aviator's output – if a generated test script isn't perfect, how to quickly adjust it. Likewise, QA/compliance staff should learn how to audit AI outputs.

OpenText will likely provide documentation and perhaps sandbox environments; Life Science Organizations could run internal workshops or "lunch and learn" sessions where teams practice using Aviator on non-critical projects first. Since generative AI might be new to many, include some basic AI literacy education: explain that the model may sometimes produce incorrect or irrelevant results ("hallucinations"), and teach how to identify and correct these. By building skill and confidence, Life Science Organizations' teams will trust the tool and know how to get the best out of it, rather than either blindly trusting it or avoiding it out of fear.

TAKEAWAYS

- **Comprehensive training on OpenText Aviator's AI features is crucial for all team members.**
- **Training should include foundational generative AI knowledge, like identifying "hallucinations."**
- **The goal is to build comfort and confidence in using AI as a collaborative tool.**
- **Teams need to learn how to interpret, refine, and audit AI-generated outputs effectively.**

ProcellaRX
PRESCRIPTION FOR SOFTWARE QUALITY

ORGANIZATIONAL READINESS REQUIREMENTS

Prompt Engineering and Testing

Using an AI assistant effectively often comes down to crafting good prompts or queries. Life Science Organizations should encourage a practice of “prompt engineering” – testing different ways of asking Aviator to ensure the best results. Teams can develop a library of proven prompts for common tasks. For instance, a structured prompt template for generating test cases (including background, assumptions, input format) might yield more consistent results. During the initial rollout, Life Science Organizations can have a small group of “power users” experiment with various prompt styles and share what works.

Consider formalizing this by adding a Prompt Testing phase in the implementation plan: e.g., take a representative sample of requirements from past projects and see how different phrasings in the prompt to Aviator change the output. Evaluate these outputs for accuracy and completeness. The findings can become guidelines (like “When asking Aviator to summarize a validation report, always specify the sections to include”). Over time, as users interact, they will naturally refine their interactions, but capturing best practices early will flatten the learning curve. This effort turns the unpredictable nature of generative AI into a more controlled, recipe-driven use – crucial in a regulated setting.

TAKEAWAYS

- **Effective AI use requires good prompts.**
- **Encourage "prompt engineering" to optimize results.**
- **Develop a prompt library and templates for consistency.**
- **Formalize prompt testing to create best practices and guidelines.**

ORGANIZATIONAL READINESS REQUIREMENTS

Oversight and AI Stewardship

Establish clear oversight mechanisms for AI usage. Life Science Organizations may consider forming an “AI steering committee” or at least assigning AI liaisons/champions within the IT Quality organization (ITCSV). These individuals would be responsible for monitoring how Aviator is being used, ensuring compliance with procedures, and serving as a bridge with the vendor on AI-related issues. For example, an AI champion in QA could periodically review a random sample of AI-generated test cases to ensure they remain high quality, as a quality audit measure.

Additionally, ensure management (Life Science Organizations leadership) is receiving regular updates on the AI adoption – e.g., quarterly reports on the types of tasks Aviator is performing, the efficiency gains realized, and any issues encountered. This keeps leadership engaged and allows them to make informed decisions about scaling usage or addressing concerns. Oversight also means making sure that if the AI is used in a new way (outside the originally validated use case), that triggers a review. Essentially, treat the AI’s “role” as something that is itself subject to performance review and continuous improvement. Having dedicated oversight will also reassure any regulatory auditors that Life Science Organizations has control over the AI and it’s not a “black box” running unchecked.

TAKEAWAYS

- **Establish clear AI oversight mechanisms, like a steering committee or champions.**
- **Assign individuals to monitor usage and ensure compliance.**
- **Provide regular updates to leadership on AI adoption and performance.**
- **Implement a review process for new AI use cases and continuous improvement.**

ProcellaRX
PRESCRIPTION FOR SOFTWARE QUALITY

ORGANIZATIONAL READINESS REQUIREMENTS

LLM Usage Policy and Data Governance

Develop a clear policy for LLM usage within Life Science Organizations' environment. This policy should detail what types of data or prompts are permitted with Aviator and what they are not. For instance, if Aviator's backend LLM is hosted in the cloud by OpenText, Life Science Organizations' policy might forbid inputting any patient-identifiable information or proprietary molecular structures into the prompt, to prevent sensitive data from leaving its controlled environment. The policy can also specify acceptable use cases (e.g., "Aviator may be used to draft test cases, but final approval must be by a human" as a rule, which echoes our validation points). Another aspect is intellectual property: if Aviator generates content, who owns it and where can it be stored? The policy might clarify that all AI-generated artifacts are Life Science Organizations property and must reside in Life Science Organizations' validated systems (e.g., if the AI provides an answer, it should be captured in ALM or a document, not just left in the AI chat interface).

The LLM policy should also address security and access control – ensure only authorized personnel can use the AI features (maybe managed through OpenText's permissions). Additionally, consider incorporating relevant external guidance: for example, reference FDA's stance on AI or the upcoming EU AI Act if Life Science Organizations operate internationally, to ensure the policy aligns with future regulatory requirements. By having a formal policy, Life Science Organizations sets expectations for all staff and creates a compliance artifact that can be shown during audits or inspections to demonstrate proactive control of AI technology.

TAKEAWAYS

- **Establish a clear LLM usage policy for Aviator.**
- **Define permitted data and use cases, especially for sensitive information.**
- **Clarify intellectual property ownership and artifact storage.**
- **Incorporate security, access control, and external regulatory guidance into the policy.**

ORGANIZATIONAL READINESS REQUIREMENTS

Change Management and Culture

Finally, prepare the organization culturally for AI adoption. Some team members may be skeptical or concerned (about job impact or reliability), so leadership should communicate the vision that Aviator is meant to augment human work, not replace it, enabling people to focus on higher-value activities. Highlight success stories or pilot results (e.g., "Using Aviator, we cut test script development time by 50%" or "AI analysis found a performance issue 2 days faster"). Engage end-users early by running a pilot project or proof-of-concept in a non-critical application to gather feedback and refine the approach before wider rollout. This pilot can act as a blueprint for scaling up.

Ensure that ITCSV (IT Compliance, Security, Validation) stakeholders are involved at every stage, so they become comfortable with the technology – this includes updating any internal validation templates or quality documents to mention AI usage as applicable. As part of organizational readiness, also review and update vendor qualification processes: OpenText providing Aviator as SaaS means Life Science Organizations should have on record vendor assurances (like SOC reports, if any, or a quality agreement) that cover the AI service. In sum, treat the introduction of Aviator not just as a tool upgrade, but as a change initiative – with training, communication, policy, and continuous improvement components – to fully realize its benefits in a compliant manner.

TAKEAWAYS

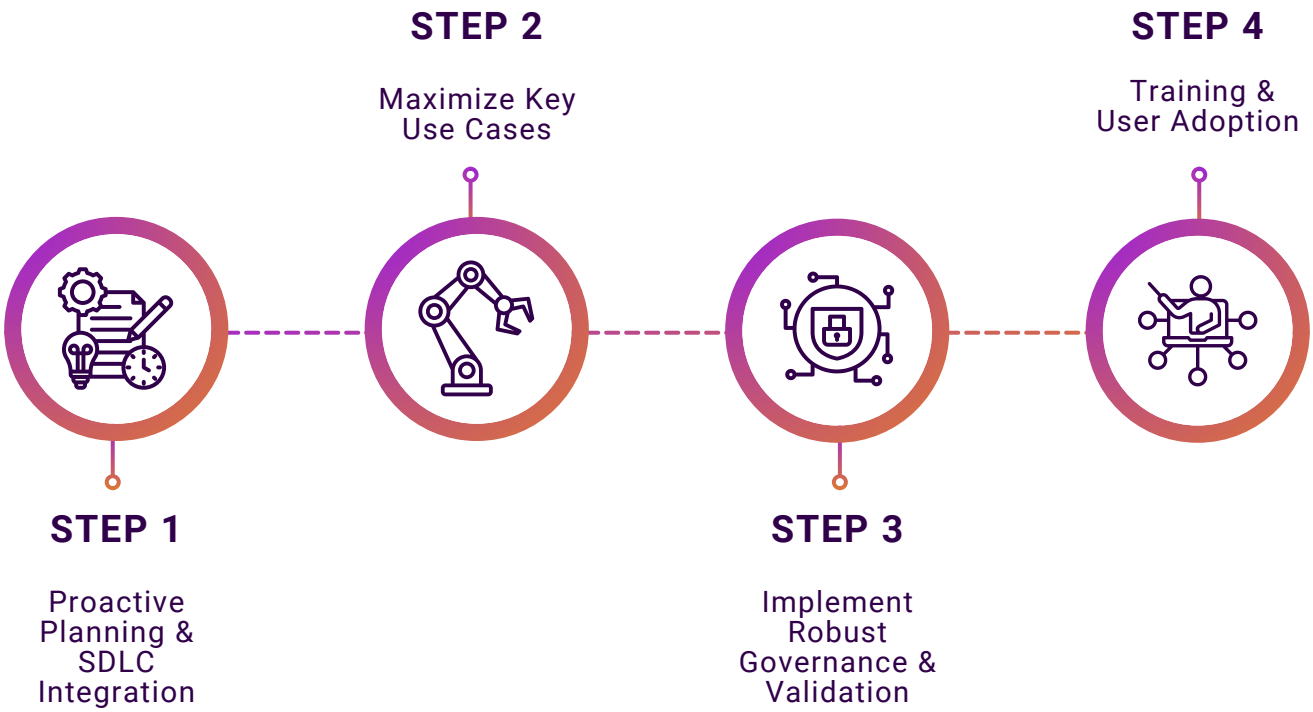
- **Prepare the organization culturally for AI adoption, addressing skepticism.**
- **Communicate AI's role as augmentation, not replacement, highlighting success stories.**
- **Engage end-users early through pilot projects to gather feedback and refine.**
- **Treat AI introduction as a change initiative, involving all stakeholders and updating processes.**

Conclusion

Adopting OpenText Aviator in 2026 can significantly elevate Life Science Organizations' QA productivity and insight, from automatically generating test cases and scripts to intelligently analyzing results and predicting risks. By proactively planning for integration and compliance, Life Science Organizations can harness this AI power while remaining firmly in control of quality and regulatory obligations. These recommendations provide a roadmap to integrate Aviator into the SDLC, capitalize on its use cases (test generation, automation, summarization, optimization, predictive analysis), and implement the necessary validation, governance, and training "guardrails" (Artificial Intelligence Governance in GxP Environments | Pharmaceutical Engineering).

With strong cross-functional collaboration between IT, QA, and compliance teams, Life Science Organizations' leadership can confidently deploy generative AI technology to streamline software testing – accelerating validation cycles and supporting the delivery of high-quality, safe products in a GxP-regulated world. The key is to embrace innovation with diligence: leveraging Aviator's capabilities to the fullest, but with clear oversight, thus ensuring that human expertise and AI innovation work hand-in-hand to achieve Life Science Organizations' objectives.

THE ROADMAP TO HARNESS AI POWER WITH CONTROL



Sources

The recommendations above are informed by OpenText's product roadmap and AI capability announcements, FDA guidelines, and industry best practices for AI in regulated environments. Key references include:

- ***OpenText's DevOps Aviator blogs detailing generative AI in testing (10 Ways Generative AI Will Revolutionize DevOps | OpenText Community) (10 Ways Generative AI Will Revolutionize DevOps | OpenText Community)***
- ***The ValueEdge Functional Test 24.3 release notes on AI-driven test conversion (Discover the Future of Functional Testing with ValueEdge 24.3 | OpenText Community)***
- ***FDA's Computer Software Assurance guidance emphasizing risk-based validation (Decoding The FDAs Draft Guidance On Computer Software Assurance For Medical Devices BioPharma)***
- ***ISPE guidance on AI governance in GxP environments (Artificial Intelligence Governance in GxP Environments | Pharmaceutical Engineering).***

These provide confidence that the strategies recommended align with both the technology's capabilities and the compliance frameworks governing its use.

About the Author

Dori Gonzalez-Acevedo
CEO ProcellaRX

Dori Gonzalez-Acevedo has over 20 years' experience in the pharmaceutical industry, specializing in regulatory compliance strategy and computer systems validation. She started her pharmaceutical career developing FDA-approved manufacturing processes. She later established risk-based quality system for operations quality groups in biotechnology firms.

Dori's focus has been on risk-based strategy and compliance since she started consulting at Genilogix, continuing through to her role as Director of Quality Systems for Healthcare and Life Sciences at Avnet / Tech Data. Most recently, as the VP of Strategic Solutions at Tx3 Services, Dori expanded Tx3's services with a focus on increasing technology adoption and advancing client understanding of compliance, automation, testing, and digital validation.

Lets Keep the Conversation Going

AI in life sciences is evolving fast—and so are the standards that will define its future. Whether you're ready to collaborate, want to learn more about our position, or are exploring how to responsibly adopt AI in your own organization—we'd love to connect.

Scan to Start a Smarter Conversation