

OpenText Aviator

Use case scenarios: Software development



Software development made simple



Why it's hard today

- Late-stage security issues delay releases
- Wasted hours debugging unclear issues, and delays from poor upfront sprint plans
- Poor test rigor leads to overlooked errors and security flaws
- Difficulty proving compliance or audit trail gaps
- Lack of insights cause release cycle delays



Powered by OpenText

- OpenText™ DevOps Aviator™
- OpenText™ Application Security Aviator™



Wins for everyone

- **Developers:** Deliver secure code faster with AI-driven testing and fixes.
- **Security team:** Spot risks early and respond faster with AI insights.

The big payoffs

► Accelerated time-to-releases

Automate security testing and integrate vulnerability scans, reducing delays.

► Adopt security everywhere

Empower devs to fix issues sooner in the SDLC with in-line remediation guidance, reducing rework.

► Strengthened regulatory compliance

Automatically check code for compliance, providing audit-readiness with no manual overhead.

Let's get started...

Secure development lifecycle (SDLC) management

Help devs catch vulnerabilities and enable early remediation.

Incident response and threat remediation

Speed up triage with AI-driven root cause analysis and recommendations.

Testing and QA automation

Auto-generate tests from requirements, UI designs, and change histories to improve coverage and accuracy.

Compliance and audit readiness

Track code risks and get evidence for compliance frameworks like ISO, SOC 2, and NIST.

Vulnerability and code risk prioritization

Flag high-risk code and rank vulnerabilities by business impact.

Explore software development scenarios



Software test ideation

I'm a developer turning specs into smart test plans without breaking a sweat. With AI spinning up test cases in seconds, I skip the slog and jump straight to better coverage, faster fixes, and fewer late-night bugs.

Let's go



Generate app test steps

I'm a developer turning test cases into clean, repeatable steps without the manual grind. With AI doing the heavy lifting, I save days each release and crank up automation—because testing shouldn't slow down shipping.

Let's go



Create app tests from video

I'm a developer turning screen recordings into smart, reusable app tests. With AI capturing every click and swipe, I save hours, boost test coverage, and let the footage do the heavy lifting.

Let's go



Build ready-to-go user stories

I'm a developer turning ideas into polished, ready-to-go user stories. With AI shaping descriptions into test-aligned gems, I fill the backlog with stories that are clear, accurate, and dev-ready—no digging required.

Let's go



Faster security code fixes

I'm a developer turning compliance alerts into clean, secure code—fast. With AI helping me patch vulnerabilities in record time, I keep releases on track and both teams happy. Fewer flaws, more high-fives!

Let's go



Smarter sprints, faster app delivery

I'm a developer turning sprint chaos into clear, automated workflows. With AI clarifying features, predicting timelines, and breaking down tasks, I deliver results faster—with fewer rewrites, sharper plans, and more time for innovation.

Let's go

Generate test cases in seconds, not sprints

I'm a software developer turning product specs into test plans.

1 Start with the spec

You're handed a product spec, like "Display transactions in a checking account." Time to turn it into tests.

Action: OpenText Aviator analyzes the spec and identifies what needs to be tested.

Impact: Skip the blank page. Go from feature to test outline in a click.

4 Tweak it 'til it's tight

Need variations? More edge cases? Just ask again.

Impact: Customize fast, iterate freely, and never slow down delivery.

See it in action!

2 Let the tests roll in

Action: Aviator autogenerates a list of AI-suggested tests.

Impact: No guesswork, no gaps, just smart, spec-driven coverage.

3 Plug and play

Pick your favorites and drop them straight into your test suite.

Action: Aviator adds your chosen tests to the plan.

Impact: What used to take hours now takes seconds.

Ship great software, every time

Unlock the possibilities:

- **Save days per feature** by eliminating manual test creation.
- **Achieve your highest levels of automation** across UA and integration testing.
- **Boost test coverage** in ways you'd never think of with AI-suggested test cases.

Generate test steps in seconds, not spreadsheets

I'm a software developer turning test cases into repeatable testing steps.

1 Pick your test case

You've selected a case like "Verify transaction history display." Time to break it into steps.

Action: OpenText Aviator analyzes the test case and prepares to generate structured test steps.

Impact: Jumpstart your testing with zero formatting friction.

4 Refine and reuse

Review the list, edit if needed, and reuse patterns for similar tests.

Action: Aviator supports refinements and variation generation.

Impact: Streamline future testing with consistent, reusable step structures.

2 Auto-generate test steps

Action: Aviator autogenerates structured test steps tailored to the case.

Impact: Say goodbye to manual...well, everything. Get steps that make sense the first time.

3 Drop into your test plan

You copy the step list and paste it right into your manual test section.

Action: One click, clean paste—ready to execute or tweak.

Impact: Go from case to plan in under a minute.

Great test cases, every time

Unlock the possibilities:

- **Save days per release** by reducing manual test writing.
- **Achieve your highest level of test step automation** across test suites.
- **Boost test consistency** with standardized steps.

See it in action!

Turn videos into test cases

I'm a software developer generating accurate, reusable tests from video.

1 Drop in your evidence

You've recorded a walkthrough and attached a feature description. Time to let OpenText Aviator take it from here.

Sample prompt: "Create manual tests using this video and description."

Action: Aviator scans the video and text to auto-generate manual tests..

Impact: From recording to repeatable steps with no notes and no guesswork.

4 Zoom in. Clean up.

Explore test logic, relationships, and inputs. Everything's ready to refine or run.

Action: Aviator highlights how manual and codeless tests align and fills in test parameters.

Impact: Test smarter, not harder. Your test suite is clear, consistent, and complete.

2 Pick what matters

Aviator gives you a smart list of suggestions. You choose what fits.

Sample prompt: "Add these tests to the feature."

Action: Aviator pulls the most relevant tests and inserts them into the right section.

Impact: One click, multiple wins.

5 Record. Generate. Done.

With Aviator, you can generate and validate tests straight from feature walk-throughs, video recordings, and specs.

Impact: Skip the typing. Let AI turn videos into tests—fast.

3 Switch to codeless mode

Want more? Time to generate codeless tests too.

Sample prompt: "Generate codeless tests using the same video."

Action: Aviator reuses your inputs to auto-create complementary no-code test steps.

Impact: Double the value from a single source.

Generate tests from analyzing video and data sources

Unlock the possibilities:

- **Save hours of work** by creating manual and codeless test cases in minutes.
- **Achieve your highest levels** of automation for test development.
- **Boost test coverage** faster than ever.

See it in action!

Turn descriptions into user stories

I'm a software developer writing accurate, ready-to-go user stories.

1 Upload your inputs

Sample prompt: "Generate user stories from this video and description."

Action: OpenText Aviator scans the attachments and generates story suggestions.

Impact: From backlog blank page to stories in seconds.

4 Dig for depth

Sample prompt: "Add test parameters and connect to codeless tests."

Action: Aviator enriches stories with technical details, mapping logic to reusable tests.

Impact: Stories that do more—fueling the whole sprint lifecycle.

2 Pick your winners

Choose the most relevant user stories and drop them straight into the feature.

Action: Aviator's ready to go with the stories you've selected.

Impact: One click and you're writing less but delivering more.

5 Send it to the board

Action: Automatically drop everything into the backlog—stories, test cases, and descriptions.

Impact: Clearer backlog, smarter sprints, faster delivery.

3 Add helpful context

Want more? Time to generate codeless tests too.

Action: Aviator connects each user story to feature details, tests, and any required inputs.

Impact: Launch development and testing with user stories that do more.

See it in action!

Generate tests from analyzing video and data sources

Unlock the possibilities:

- **Save hours of work** by creating manual and codeless test cases in minutes.
- **Achieve your highest levels** of automation for test development.
- **Boost test coverage** faster than ever.

Fix code flaws fast with compliance and dev in sync

I'm a developer squashing vulnerabilities sent from the compliance officer.

1 Get the alert

A security scan flagged an SQL injection in an online shopping app. The compliance officer assigned it to you for review.

Action: OpenText Aviator sends an issue summary and recommends immediate review.

Impact: Get clear, immediate insight—no need to sift through logs.

4 Ship the fix

Using Aviator's guidance, you fix the code. The next scan will verify resolution.

Impact: The issue gets resolved quickly, with AI as a shared guide between security, dev, and compliance.

2 Jump into the code

Open "My Open Issues" and dive into the flagged code.

Action: Aviator helps analyze the source code and identify the root cause of the issue.

Impact: Skip the wild goose chase and go straight to the problem.

5 Drive proactive response

The collaboration between dev and compliance—with help from Aviator—keeps the app secure and audit-ready.

Impact: Faster fixes, fewer compliance gaps, and a stronger security posture.

3 Just ask OpenText Aviator

The issue's complex, so you pull up the AI-generated guidance.

Action: Aviator explains the risk and suggests a fix, right down to the code level.

Impact: Know what to do and why.

Bridge compliance and code

Unlock the possibilities:

- **Reduce friction** between dev and compliance teams with shared visibility and AI summaries.
- **Detect and fix** vulnerabilities faster.
- **Address flagged issues** with clear compliance context.

See it in action!

Smarter sprints, faster application delivery

I'm a software developer delivering top results on my next sprint.

1 Drop in new improvements

Sample prompt: "Identify elements that could improve the [XYZ feature] description."

Impact: Reviews descriptions, highlights gaps, and suggests improvements for clarity and user-friendliness.

2 Forecast the finish line

Sample prompt: "What is the predicted delivery time for [XYZ feature]?"

Impact: Analyzes historical project data and current progress to provide more accurate forecasts for the completion of features and projects.

3 Learn from user stories

Sample prompt: "Suggest a breakdown of user stories for [XYZ feature]."

Impact: Auto-generates a list of structured user stories to guide development and improve sprint outcomes.

4 Tasks untangled

Action: Generate a breakdown of suggested tasks for [feature].

Impact: Dissects features into actionable tasks, helping teams stay focused and organized.

5 Sprint like a pro

Optimize software sprint processes and automate tasks with a few clicks to get the job done right every time – OpenText Aviator automates faster, high-quality software delivery!

See it in action!

Optimize software sprints with AI-powered workflows.

Unlock the possibilities:

- **Boost Dev velocity**—Clarify features and break down tasks for faster builds.
- **Speed time-to-market**—Reduce rework and speed delivery with smarter plans.
- **Improve flow efficiency**—Predictive insights for better sprint execution.
- **Enhance quality**—Generate user stories and test support for max results.



✦ Limitless