



Software Development

OpenText™ Aviator Use Case Scenarios





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!](#)

Software Development made simple



Why it's hard today

- Late-stage security issues delay releases
- Wasted hours debugging unclear issues
- Poor test rigor leads to overlooked errors and security flaws
- Difficulty proving compliance or audit trail gaps
- Customers disengages due to lack of value or clarity



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.
- ▶ **Shift “everywhere” security adoption**
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 early remediation.

Incident Response & Threat Remediation: Speed up triage with AI-driven root cause analysis and recommendations.

Testing & QA Automation: Auto-generate tests from requirements, UI designs, and change histories to improve coverage and accuracy.

Compliance & Audit Readiness: Track code risks and provides evidence for compliance frameworks like ISO, SOC 2, and NIST.

Vulnerability & Code Risk Prioritization: Flag high-risk code and ranks vulnerabilities by business impact.



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.

2 Let the tests roll in

Action: OpenText Aviator auto-generates 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: OpenText Aviator adds your chosen tests to the plan.

Impact: What used to take hours now takes seconds.

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!](#)

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.

2 Auto-generate test steps

Action: OpenText Aviator auto-generates 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.

4 Refine and reuse

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

Action: OpenText Aviator supports refinements and variation generation.

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

[See it in action!](#)

Test great 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.



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: OpenText Aviator scans the video and text to auto-generate Manual Tests.

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

2 Pick what matters

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

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

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

Impact: One click, multiple wins.

3 Switch to codeless mode

Want more? Time to generate codeless tests too.

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

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

Impact: Double the value from a single source.

4 Zoom in. Clean up.

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

Action: OpenText 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.

5 Record. Generate. Done.

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

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

[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.



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.

2 Pick your winners

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

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

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

3 Add helpful context

Want more? Time to generate codeless tests too.

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

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

4 Dig for depth

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

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

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

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.

[See it in action!](#)

Deliver backlog gold—instantly usable, test-aligned stories.

Unlock the possibilities:

- **Save hours per feature** with automated user story creation.
- **Align dev and test** with deeply integrated story-to-test links.
- **Deliver sprint-ready backlog items** with full feature context.



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 a SQL injection in an online shopping app. The compliance officer assigned it to me for review.

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

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

2 Jump into the code

I open “My Open Issues” and dive into the flagged code.

Action: OpenText 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.

3 Just ask OpenText Aviator

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

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

Impact: Know what to do and why.

4 Ship the fix

Using OpenText Aviator's guidance, I 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.

5 Drive proactive response

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

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

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!](#)



 **Limitless**