DINÉ DEVELOPMENT CORPORATION

# Closing the Test Gap

## AI-generated Test Plans, Traceability, and Automation for Release Validation

**Michael Peterson**
Principal Solutions Architect
Diné Development Corporation

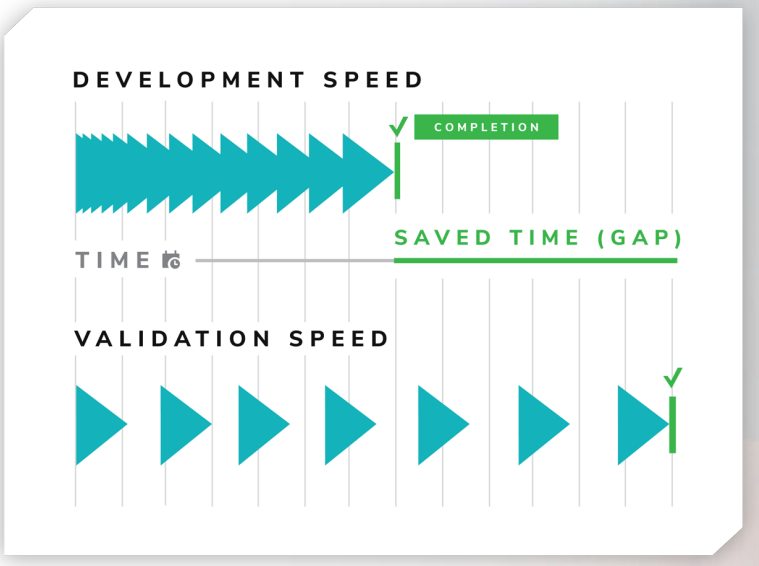
# Agenda

- The Validation Gap
- The Federal Burden of Proof
- Where Manual Validation Breaks
- Generating Validation from Source
- Agentic TestOps
- The Governed Lifecycle
- Traceability as a Control Surface
- Tester-in-the-Loop Governance
- Live Demonstration
- What is Next?


© DDC - Proprietary and Confidential Business Information

# AI Is Accelerating Development, Not Validation

- AI is accelerating how teams generate and modernize software
- Test planning, traceability, and evidence collection stay manual
- Faster code without faster validation moves the bottleneck downstream
- The gap widens between "the software is complete" and "the release is proven"
- AI-generated automated tests built against code —not requirements—risk validating implementation errors instead of intended behavior





© DDC - Proprietary and Confidential Business Information

# Federal Release Validation - Higher Burden of Proof

| Delivery Pressure | Assurance Requirement |
|---|---|
| Shorter release cycles | Requirements traceability |
| Continuous modernization | Repeatable regression coverage |
| Limited tester capacity | Documented test evidence |
| Changing requirements | Review and approval history |
| AI-generated code | Human accountability |

**Release is proven by evidence**

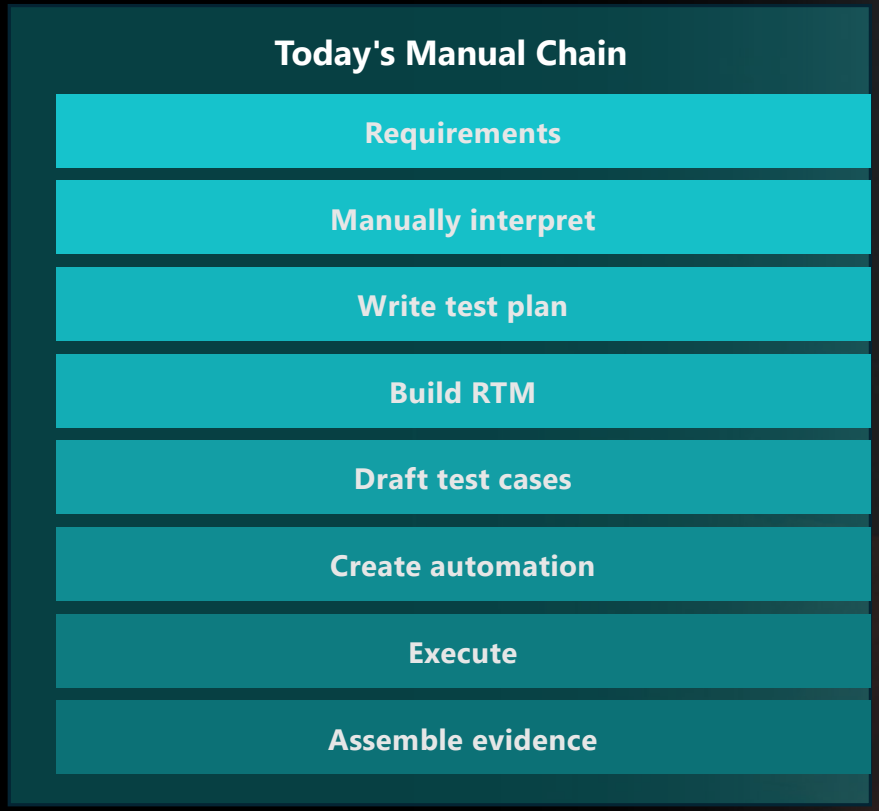
- What was required
- What was tested
- What passed
- Who approved it


© DDC - Proprietary and Confidential Business Information

# Manual Validation Does Not Scale with Release Velocity

- Requirements and test cases drift apart
- Regression suites no longer reflect the current release
- Automation is created after manual testing, not alongside it
- Evidence scatters across documents, screenshots, and tools
- Reviewers cannot reconstruct why a release was approved

**Test engineers spend their time producing documents instead of analyzing risk**



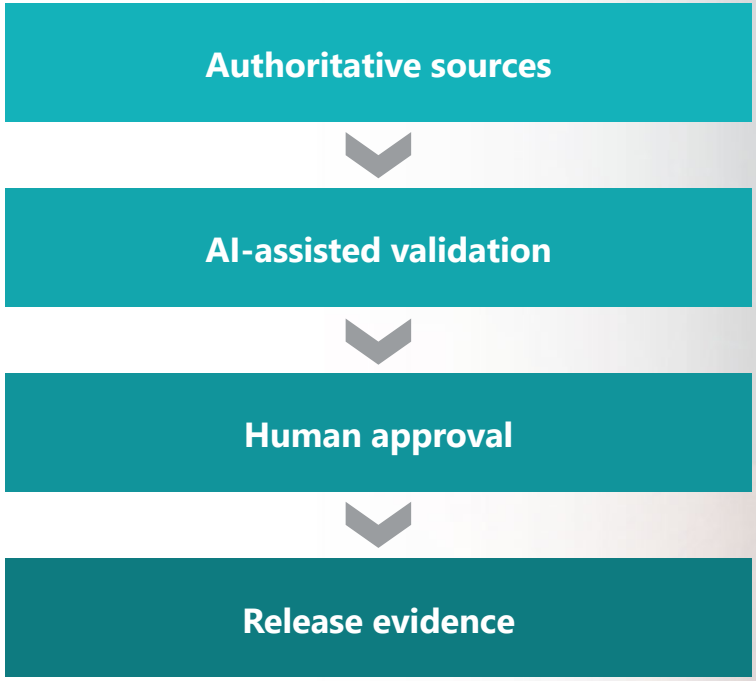


© DDC - Proprietary and Confidential Business Information

# Generate Validation from the Sources that Define the Release

- Ingest application definitions, requirements, stories, or existing test cases
- Identify changes between releases
- Generate a connected validation package
- Review and refine every artifact with qualified testers
- Execute approved cases through browser agents

**Validate against functional requirements and use cases—not developed code, which can drift from intended behavior**





© DDC - Proprietary and Confidential Business Information

# Human-Governed Release Validation Platform

Tester-in-the-loop platform that turns authoritative requirements into traceable tests, automation, evidence, and release decisions

**1**

## Understand

**Ingest application and release information**

**2**

## Plan

**Generate the plan, diff, RTM, and cases**

**3**

## Execute

**Browser agents run approved procedures**

**4**

## Prove

**Preserve evidence, approvals, and audit history**

**Input → Generate → Review → Approve → Export**


© DDC - Proprietary and Confidential Business Information

# One Governed Workflow from Intent to Evidence

1. Create release session
2. Ingest authoritative sources
3. Compute semantic difference
4. Generate test artifacts
5. Review and approve
6. Execute via browser agents
7. Generate automation
8. Capture evidence
9. Export validation package

**Tester control and audit trail — continuous across all nine stages**


© DDC - Proprietary and Confidential Business Information

# Meet Programs Where Test Knowledge Already Exists

**Supported Starting Points**

- AppDNA and structured app definitions
- Requirements documents
- Agile epics and user stories
- Existing test cases
- Previous and current release artifacts
- Supporting technical documentation

- Programs do not need perfect documentation to begin
- Existing test cases are parsed to infer the requirements they cover
- Teams that start with test assets are fully supported
- Agent-driven intelligent document processing extracts structure and intent from diverse source artifacts
- The pipeline transforms fragmented inputs into traceable test knowledge for human review


© DDC - Proprietary and Confidential Business Information

# Semantic Release Diffing Makes Regression Risk-Based

**Previous + Current Release** › **Semantic Difference** › **Targeted Validation Scope**

- Textual difference: word or file changes
- Semantic difference: behavior, rule, interface, or acceptance condition changes
- Use release context to concentrate validation where risk moved

- New test cases
- Modified test cases
- Required regression coverage
- Retired or obsolete tests
- Areas requiring tester investigation


© DDC - Proprietary and Confidential Business Information

# Artifacts Form a Traceable System, Not Separate Documents




## Generated Artifacts

- Release Difference Report
- Test Plan
- Requirements Traceability Matrix
- Coverage metrics
- Manual and automation-ready test cases
- Exportable, CI/CD-ready automation project, which integrates with existing pipelines and runs without a TestOps dependency
- Review and decision history


© DDC - Proprietary and Confidential Business Information

# The RTM Becomes Operational View of Release Readiness

| Requirement | Test Coverage | Execution | Evidence | Approval |
|---|---|---|---|---|
| R-101 | Full | Passed | Available | Approved |
| R-102 | Partial | Pending | Missing | Review |
| R-103 | None | Not run | None | Blocked |

- Live requirement-to-test linkage
- Uncovered and partially covered requirements surface immediately
- Positive, negative, and boundary coverage
- Reviewer visibility into unresolved gaps

**Coverage is visible before release approval, not reconstructed afterward**


© DDC - Proprietary and Confidential Business Information

# Agentic Does Not Mean Autonomous

## AI proposes

- Requirement interpretations
- Test plans and traceability links
- Test cases
- Automation drafts
- Potential coverage gaps

## Test engineers decide

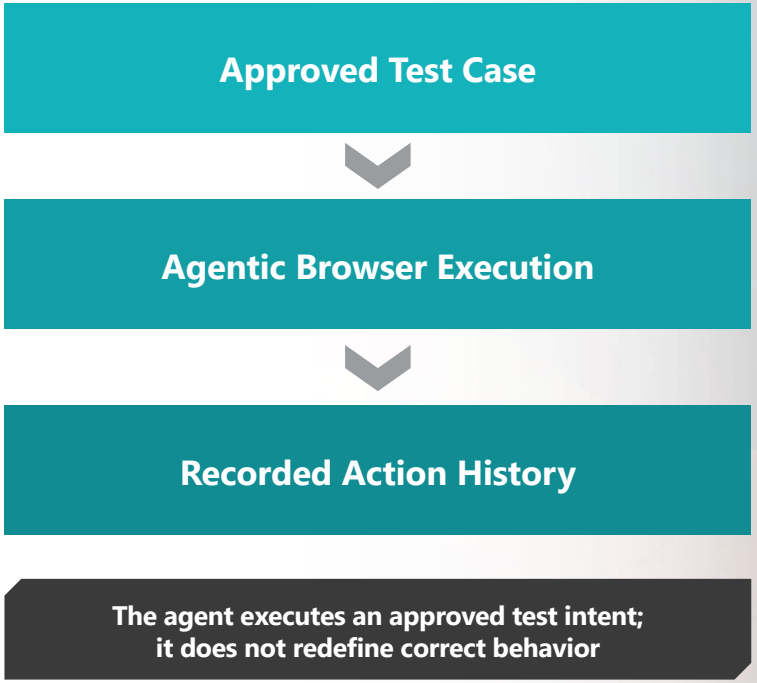
- Whether interpretations are correct
- Whether coverage is sufficient
- Which tests may execute
- Whether generated automation is acceptable
- Whether the release is approved

**Inline editing** | **Scoped regeneration** | **Artifact locking** | **Approval gates** | **No silent changes** | **Full decision history**


© DDC - Proprietary and Confidential Business Information

# Browser-Use Agents Turn Procedures into Recorded Execution

- Interprets the approved test procedure
- Navigates the application and performs each step
- Records actions and observed outcomes
- Captures execution evidence
- Flags significant workflow differences for tester review
- Reasons over page context to select the next approved action
- Adapts to minor UI changes without changing test intent





© DDC - Proprietary and Confidential Business Information

# Record Once, Produce Maintainable Automation

**Recorded Execution → Action Ledger → Framework Project**

**Selenium**

**Playwright**

**Cypress**

- Preserve linkage to the requirement and test case
- Engineers can inspect and modify generated code
- Export a complete automation project
- No proprietary playback format lock-in


© DDC - Proprietary and Confidential Business Information

# Trust Requires More Than a Passing Test

## Execution Visibility

- Application and server monitoring
- Logs and request behavior
- Resource and runtime observations
- Screenshots, recordings, and execution details

## Credential Protection

- Store secret references, not values
- Retrieve credentials only at execution time
- Scope access to session and runner
- Keep secrets out of tests and evidence

## Accountability

- Execution identity
- Artifact version
- Test and requirement linkage
- Result and evidence
- Reviewer and approval history


© DDC - Proprietary and Confidential Business Information

# Live Demo

From requirements to runnable test — one connected path

- Session and ingested sources
- Semantic release difference
- Test plan and RTM coverage gap
- Agent executes an approved case
- Automation, evidence, and audit history


© DDC - Proprietary and Confidential Business Information

# Build the Evidence Package as Testing Happens

| TestOps Output | Assurance Value |
|---|---|
| Requirements traceability matrix | Shows requirement-to-validation linkage |
| Approved test plan | Documents validation scope and strategy |
| Test cases and automation | Demonstrate repeatable verification |
| Execution results | Record observed outcomes |
| Screenshots and recordings | Support evidence review |
| Coverage metrics | Expose unresolved validation gaps |
| Audit history | Preserves review and approval lineage |

**Agentic TestOps does not grant an ATO; it produces traceable, reviewable test evidence that supports RMF assessment, authorization, and reauthorization**


© DDC - Proprietary and Confidential Business Information

# What We Demonstrated and What Still Needs Judgment

## Demonstrated

- AI-assisted generation of test artifacts
- Requirements-to-test traceability
- Automation generation, execution, and browser replay
- Evidence export
- Tester-in-the-loop review in a federal test context

## Requires Continued Discipline

- Environment and test-data readiness
- Stable application access
- Validation of generated artifacts
- Long or highly conditional workflows
- Security approval per deployment environment

**Maturation of maintenance as business behavior changes and scaling execution across larger suites**


© DDC - Proprietary and Confidential Business Information

# From Artifact Production to Assurance Engineering

| Less Time Spent On: | More Time Spent On: |
|---|---|
| Reformatting requirements | Challenging requirement quality |
| Building RTMs manually | Evaluating coverage sufficiency |
| Repeating browser steps | Designing high-risk scenarios |
| Translating steps into scripts | Reviewing automation behavior |
| Collecting evidence after testing | Making release-risk decisions |

**The goal is not to remove test engineers; it is to give them leverage while preserving their authority over quality and release readiness**


© DDC - Proprietary and Confidential Business Information

# What We Proved — and What We Learned From What Didn't

**A jointly executed DDC Innovation Solution Accelerator proof of concept measurably reduced the ACES testing bottleneck, and the tests that did not automate proved just as informative as the eight that did.**

## TODAY — THE MANUAL BASELINE

- **Single test execution** — 30–60 minutes
- **Full regression cycle** — 10–15 business days
- **Test authoring** — 2–3 days per test (Dev → PMO → FMO)
- **Test repository** — 1,444 manual · 0 automated

## DEMONSTRATED IN THE PROOF OF CONCEPT

- **Single test execution** — 1–5 minutes (~93% faster)
- **Automation start-up** — no long ramp vs. traditional scripting
- **Evidence** — full artifact package captured every run
- **Workforce effect** — burden reduced, regression coverage up

**8** **of 10 targeted tests automated**
ARIS 2/4 · FES-IMS 4/4 · Security 2/2 (access control)

**2** **blocked — application workflow, not the tool**
T11690 · T11706 → ACES improvement candidates

**3** **removed & replaced — tool capability gaps**
T11682 · T11689 · T11705 → enhancements planned

| ~93% | 30–60 min → 1–5 min | 1,444 / 0 |
|---|---|---|
| less time per test execution | demonstrated on live ACES tests | manual tests / automated at baseline |

**A diagnostic lens.** The two blocked tests were not tool failures — they exposed ACES application workflow constraints. With three capability gaps now on the enhancement roadmap, the proof of concept returned actionable application and process findings available to the program now, independent of any automation decision.

# Built for the Government Environment — Secure & Adaptable

| | |
|---|---|
| **Authentication & access control** | Integrates CAC/PKI authentication and role-based access control (RBAC). Both RBAC security tests were automated with the generated evidence reviewed and accepted by Cybersecurity Team |
| **Security scope** | Automated coverage is access-control (RBAC) only. Input-validation testing — malicious file type and injection (T11705) — was NOT automated in this proof of concept and is on the enhancement roadmap |
| **Accreditation support** | IA-ready artifacts are usable by ISSO/M for control submission — with potential to reduce duplicate ISSO/M testing. Not yet demonstrated; requires PMO and IA concurrence. |
| **Repeatable and auditable** | Runs tests repeatably and automatically captures a complete artifact package for every run — screenshots, video, test case, test script, and test plan, reducing manual variability and transcription error |
| **Adaptable and GFE-friendly** | Enhanced in under 24 hours from user feedback (example: GFE file-type handling, .ts → .txt and .webm → .mp4). Operates within GFE constraints. |

*The ACES Program Management Office and the CE System team co-executed the DDC Agentic TestOps proof of concept. We observed measurable reductions in test execution time and manual testing burden, with security test evidence reviewed and accepted by PMO Cybersecurity. The effort proved equally valuable for what it could not yet automate — surfacing application workflow and process improvements we can act on now.* ***"I want other leaders to be thinking about how their programs can benefit from test automation wholistically and technologies such as the one discussed here and am open to having the discussions about implementation."***

**— Lt. Col. Tyler J. Hough** Materiel Leader, AFLCMC/GBM, ACES Program Management Office


© DDC - Proprietary and Confidential Business Information

# Faster Delivery Requires Faster Proof

1

AI-generated software increases the need for disciplined validation

2

One governed workflow connects requirements, tests, automation, evidence, and approvals

3

The result is not autonomous testing; it is human-accountable release assurance


© DDC - Proprietary and Confidential Business Information

DDC
DINÉ DEVELOPMENT CORPORATION

# Thank you!

**Reduce Test Time. Increase Mission Readiness.**

## SCHEDULE A DEMO WITH DDC

**Michael Peterson**
Principal Solutions Architect
**michael.peterson@ddc-dine.com**