Moving on from qTest

How to select new tools and migrate your data

NEIL CRICHTON
Environment & QA Lead

November 2025

Mercury

THE PROBLEM: WHY WE HAD TO LEAVE QTEST

⏰ **Hard Deadline: 2 months to select a new tool and migrate**
License not being renewed due to cost

> The Pain Points

- 💰 **Expensive:** High licensing costs that weren't justified by the value
- 🐌 **Slow Innovation:** New features took forever to reach the market
- 🤖 **Poor AI Integration:** Limited to JIRA story descriptions only
- 🗂️ **Clunky UI:** Navigating through layers of tree structures
- 😞 **Team Sentiment:** Large negative sentiment across the organization

The Stakes

20,000 Test Cases

14,000 Executions

Projects In-Flight

TOOL EVALUATION: FAST & FOCUSED

> Our Shortlist

- **BrowserStack Test Management** - Full evaluation
- **TestRails** - Full evaluation
- **Katalon TestOps** - Eliminated (wouldn't meet deadline)

> Evaluation Approach

- **Decision Maker:** Me (with team input)
- **Method:** Criteria list + analyst trials + informal feedback
- **Key Factors:**
    - Cost and licensing model clarity
    - User interface intuitiveness
    - AI capabilities
    - Team buy-in

> Why BrowserStack Won

- 🎨 **User-Friendly UI:** Intuitive, easy to find test executions
- 🤖 **Superior AI:** Conversational AI with supporting documentation
- ✅ **Team Approval:** Minimal resistance
- 💰 **Cost-Effective:** Clear licensing at better price

THE MIGRATION CHALLENGE

> What Auto-Imported

✅ **Test Cases**
✅ **Test Case Attachments**

> What Didn't Auto-Import

❌ **Test Execution History**
❌ **Test Execution Attachments**
❌ **Test Plan Structures**
Note: Added to auto-import months later - a con of early adoption!

> The Real Challenge

Data Structure Mismatch: qTest's complex 3-tier structure had to map to BrowserStack's clean hierarchy

THE "AHA MOMENT"

Every qTest element could be reduced to 6 scenarios based on:

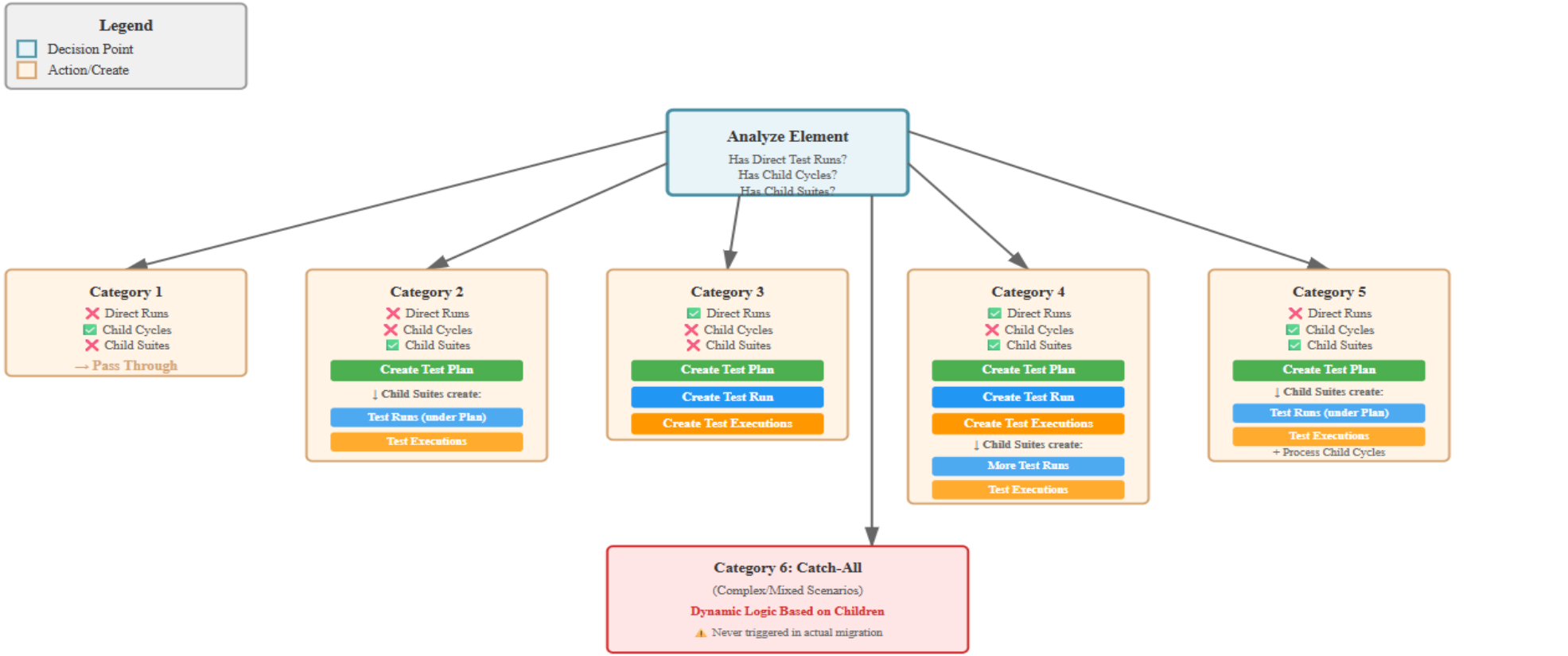
- Does it have direct test executions?
- Does it have child test cycles?
- Does it have child test suites?

> Development Timeline

- **Week 1:** Analyzed qTest projects to identify all data combinations
- **Week 2:** Built and tested migration script to handle all 6 scenarios
- **Result:** Successfully migrated 20,000 test cases and 14,000 test executions

Once categorized, each element mapped cleanly to BrowserStack's Plan → Run → Execution hierarchy

qTest Element → BrowserStack Structure Mapping



BrowserStack Target Structure

Test Plan
Top-level container
Groups related test runs
e.g., "Sprint 5 Regression"

contains

Test Run
Execution session
Contains test executions
e.g., "Login Tests - Build 123"

contains

Test Execution
Individual test result
Pass/Fail/Blocked status
e.g., "TC-001: Valid Login"

The "Aha Moment"

qTest's 3-tier structure (Releases/Cycles/Suites) with complex nesting patterns could always be reduced to 6 decision scenarios based on:

Does it have direct test runs? • Child cycles? • Child suites?

Once categorized, each element maps cleanly to BrowserStack's Plan → Run → Execution hierarchy

EARLY ADOPTION: RISK VS REWARD

BrowserStack Test Management had been on the market for only a few months when we migrated.
We already used other BrowserStack products.

⚠️ The Risks

- **Missing Features:** Some reporting capabilities weren't built yet
- **Manual Migration:** Built scripts for data that later auto-imported
- **Minor Bugs:** Small issues in accelerated development
- **Workarounds:** Had to pull data via API for custom reports

🎯 The Rewards

- **Direct Line to Product:** Feature requests prioritized
- **Rapid Bug Fixes:** Issues resolved within days
- **Influence Direction:** Helped mold the tool for our needs
- **Ecosystem Access:** Access to PMs on other BrowserStack tools

Bottom Line: Being an early adopter gave us a seat at the table to build the tool we needed.

LESSONS LEARNED: WHAT I'D DO DIFFERENTLY

1. Engage Vendor Developers Earlier

The Issue: Built complex migration logic the vendor might have accelerated.
Next Time: Discuss complex challenges with dev team early.

2. Be Ruthless with Data Culling

The Reality: ~40% of migrated test cases were outdated.
The Mistake: People panicked about losing data despite backups.
Next Time: Set clear criteria. Archive, don't migrate old data.

3. Evaluate More Options

Next Time: Add one or two more tools to evaluation list for better perspective and negotiating leverage.

KEY TAKEAWAYS

When Evaluating Tools:

- Get hands-on feedback from actual users
- Cost matters, but so does licensing clarity
- User experience drives adoption
- Consider vendor's innovation velocity

When Migrating Data:

- Understand both data structures completely
- Look for patterns to reduce complexity
- Be ruthless about what needs migrating
- Build in validation and testing time

On Early Adoption:

- Sometimes "risk" is actually opportunity
- Direct product team access is invaluable
- Your input can shape the tool's future
- Have backup plans for missing features

Q & A

Discussion Point:
“How does your team balance innovation risk vs. stability when adopting new tools?”