FlareAI

How AI-Powered Core Conversions Can Reduce Risk, Cost, and Downtime by 80%

Transforming Banking's Most Dreaded Project into a Manageable Business Initiative

by Jay Cody
sales@theflare.ai

White Paper | 2026

Table of Contents

Executive Summary

Core banking system conversions represent the highest-stakes operational risk financial institutions face, consuming 18-36 months, regularly exceeding budgets by 30-50%, and exposing institutions to regulatory scrutiny, customer attrition, and data integrity failures that can persist for years.

Industry statistics paint a sobering picture:

75%
of conversions exceed their original timeline

68%
experience significant budget overruns

1 in 5
is ultimately deemed a failure by the institution itself

These aren't just isolated cases from struggling financial institutions as they reflect the deep-seated challenges built into traditional conversion methods.

Yet artificial intelligence has reached an inflection point where it can address the core challenges that have made conversions so consistently problematic. **Early implementations demonstrate that AI-powered automation could compress testing timelines by 50%, reduce manual documentation labor by 70%, and provide objective visibility into conversion readiness that has never before been possible.**

This white paper examines how intelligent automation is transforming three critical conversion domains:

- ✓ Testing documentation and quality assurance
- ✓ Customer communication and change management
- ✓ Staff training and knowledge transfer

For executives contemplating a core conversion, or recovering from one, this analysis offers a framework for evaluating how AI enables what was previously impossible: converting core conversion from an existential risk into a measurable, manageable business initiative.

> **The question** is no longer whether AI will reshape core conversions, but whether your institution will be among the early adopters capturing competitive advantage.

The $50 Million Dollar Question: Why Do Core Conversions Still Terrify Boards?

Picture this: Eighteen months into your new core implementation, you're called into an emergency board meeting. The project's timeline has extended twice, and the budget has swollen from $12 million to $18 million. Your deposit operations team hasn't handled their regular work in three months as they've been entirely focused on testing, documentation, and issue tracking. Customer complaints have tripled. Your head of retail banking has just handed in a resignation. Lastly, your examiners are now raising concerns about general ledger reconciliation discrepancies found during parallel processing.

The consultant leading the conversion assures you that this is "normal" for projects of this complexity. **Alarmingly, the consultant is right.**

The Seven Structural Impossibilities

The challenge isn't just execution quality, it's structural impossibilities inherent to traditional approaches:

1. The Testing Documentation Bottleneck

Creating comprehensive test documentation with user stories, detailed test cases, and clear defect reports, requires thousands of staff hours. A typical conversion needs 800-1,200 test cases. At 2-3 hours per test case to write properly, that's 1,600-3,600 hours of documentation work before a single test is executed. Most institutions simply can't afford that level of documentation, so they document what they can and hope it's a good representative sample.

2. The Knowledge Transfer Bottleneck

Core expertise is concentrated among a small group of veterans who are expected to maintain operations, lead testing efforts, validate data, and train hundreds of staff at the same time. It’s an impossible equation.

3. Communication Overload

Executing personalized customer communication at scale, different information at different times through different channels based on relationship depth and product mix will undoubtedly exceed the manual capacity of your marketing team.

4. The Coverage Visibility Gap

Testing teams have no objective way to measure what percentage of system functionality they've actually tested. They count completed test cases but have no idea if those cases cover 40% or 90% of what the new system actually does.

5. The Resource Cannibalization Trap

Traditional conversions pull your best people from productive work for 12-18 months, creating operational risk independent of conversion itself.

6. The Defect Documentation Crisis

When testers find issues, they struggle to write clear, actionable bug reports. Developers and configuration experts waste days deciphering vague descriptions like "fee calculation wrong on business checking" with no context about what was tested, what was expected, or how to reproduce the issue.

7. The Opportunity Cost Paralysis

When an institution is engrossed in a major core conversion, nearly all available resources and attention are diverted to project execution. Therefore, key strategic initiatives such as launching new products, expanding into new markets, enhancing digital capabilities, or deepening customer engagement are put on hold.

The Board's Impossible Dilemma

Directors face an unsolvable calculus. The existing core may be limiting capabilities, costing millions in workarounds, and creating regulatory findings. Yet proceeding with conversion means accepting a **75% likelihood of overruns**, exposing the institution to operational risks, and gambling on vendor promises.

This isn't a choice between good and bad options. It's selecting between a system you've outgrown and a project with failure rates that would be unacceptable in any other business context.

Until recently, this dilemma had no good answer.

The AI Transformation: When Impossibilities Become Possible

Something fundamental has shifted. AI has moved from an experimental technology to enterprise-grade tools solving problems previously considered unsolvable at scale.

> The breakthrough isn't making humans test faster. Instead, it's recognizing that the documentation work that has always bottlenecked testing shouldn't require human intelligence at all.

Consider test case creation: A skilled business analyst might document 15-20 comprehensive test cases per day. An AI system can generate 500 detailed, properly formatted test cases in the time it takes that analyst to create 5, and the AI-generated cases include edge cases and integration scenarios the analyst might never think to document.

Traditional Approach

A skilled business analyst documents **15–20 comprehensive test cases per day**

AI-Powered Approach

An AI system generates **500 detailed, properly formatted test cases** in the time it takes that analyst to create 5

*and the AI-generated cases include edge cases and integration scenarios the analyst might never think to document.

The analyst isn't working faster. The analyst is operating at a fundamentally different level, **reviewing and refining AI-generated documentation rather than creating it from scratch.**

This is intelligence multiplication, and it changes conversion economics entirely.

Three AI-Powered Transformations

1. Testing Documentation & Quality Assurance:

From Manual Documentation to AI-Generated Test Assets

Traditional Limitation	AI-Enabled Capability
Months spent creating test cases before testing begins. Vague bug reports waste developer time with no objective measure of functional coverage.	Automated generation of user stories, test cases, and comprehensive defect reports. Objective tracking of system functional coverage and automated test script generation.

Executive Value: We've moved from "completing 87% of planned tests" to "we've tested 78% of system functional capabilities with gaps in these specific areas".

2. Customer Communication:

From Mass Messaging to Predictive Engagement

Traditional Limitation	AI-Enabled Capability
Generic communications create confusion and call center surges.	Analyze behavior patterns to predict concerns and generate personalized, channel-appropriate messaging at scale.

Executive Value: Communication transforms from a cost center to a trust-building opportunity that reduces customer attrition.

3. Staff Training:

From Classroom Overload to Contextual Guidance

Traditional Limitation	AI-Enabled Capability
Front-loading months of training employees likely forget before using.	Just-in-time training with role-specific guidance at the moment of need.

Executive Value: Maintains productivity throughout the conversion process while ensuring measurable competency.

Deep Dive: AI-Powered Testing Documentation & Quality Assurance

The Traditional Testing Nightmare

If you've managed a core conversion, the following timeline is painfully familiar. Let’s recap:

Months 1-12

Data mapping work. Testing is planned but not prioritized.

Months 13-18

Testing begins. A small team manually writes hundreds of test cases based on their experience. Senior analysts spend 60-70% of their time documenting what should be tested rather than actually testing. They then discover the new system handles transactions differently and need to go back and reconfigure.

Months 19-22

You’ve made progress with parallel processing. However, thousands of discrepancies emerged. The team is working 70-hour weeks identifying root causes. When issues are found, testers struggle to write clear bug reports. Developers receive vague descriptions and waste days clarifying: "What exactly did you test? What did you expect? How do I reproduce this?"

Months 23-24

Executive decisions with 200 items still on the discrepancy list. The testing lead reports "we've completed 412 of 485 planned test cases - 85% done." But nobody knows what percentage of the system's actual functionality those 412 cases covered. So, do we go live and fix issues as they emerge, or delay six months?

Month 25+

Post-conversion chaos. Customers report issues never seen in testing because the test cases never covered those scenarios and nobody realized the gap until customers found it.

Therein lies the fundamental problem. **Creating comprehensive test documentation was never achievable with manual approaches.** Your testing team can execute tests, but someone must write the user stories, generate the test cases, document the bugs clearly, and create automation scripts. This can consume thousands of hours of work before a single test can be run.

What AI-Powered Testing Actually Means

Emerging platforms like Flare's **DeployIQ** operate on a fundamentally different model by automating the enormous documentation burden that has always bottlenecked testing efforts.

Intelligent User Story & Test Case Generation

Traditional Approach Senior business analysts spend weeks or months creating user stories and test cases:	**AI-Powered Approach** The system generates comprehensive test documentation at scale:
A skilled analyst might document 15-20 comprehensive test cases per day	Users select functional areas they want to test (such as "account opening" or "wire transfers")
Each test case requires understanding business context, system behavior, and edge cases	AI generates hundreds of detailed user stories and test cases for those functional areas in minutes
Coverage is limited by how many scenarios the analyst can imagine and document	Documentation includes not just "happy path" scenarios but edge cases, exception handling, and integration points
Test cases are vague and poorly written causing confusion among testers	Test cases are consistently formatted and complete without missing steps or vague instructions
The team starts testing months into the project because documentation takes so long AI-Powered	The team starts actual testing weeks earlier because documentation is ready faster

Real-World Impact: A regional banking testing team traditionally spent four months creating test documentation before beginning execution. With AI-generated test cases, they had comprehensive documentation in three weeks, recovering 13 weeks of project timeline and allowing testing to begin while configuration work was still in progress.

Objective Functional Coverage Tracking

Traditional Approach Test teams have no objective measure of coverage:	**AI-Powered Approach** The system maintains a comprehensive map of the core system's complete functional capabilities:
“We've completed 87% of planned test cases” tells you nothing about what functionality you've actually tested	System quantifies what percentage of total system functionality has test coverage versus what remains untested
No way to know if critical capabilities have test coverage	Coverage gaps become visible: "You have 47 test cases for deposit accounts but only 8 for loan accounts, which represent 35% of your revenue"
Coverage gaps discovered only when customers or examiners find issues post-conversion	System analyzes the entire documented system capabilities and confirms test cases exist for 100% of the covered functionality.
Manually tracking test results with no dynamic dashboards and prone to human error	Real-time dashboards show coverage across functional domains that matter

Real-World Impact: A banking testing team discovers they have created 215 test cases for retail banking but only 43 for commercial banking, despite commercial representing 40% of deposits. AI-driven coverage analysis reveals they had tested only 31% of commercial banking capabilities versus 86% for retail. This visibility prompts a rebalancing that identifies critical defects in commercial processes that would have reached production undetected.

Automated Bug Description & Documentation

Traditional Approach When testers find issues, they must document them:	AI-Powered Approach The system generates comprehensive, actionable defect reports:
Step 1: A tester may discover that a fee calculation is wrong	**Step 1:** The tester identifies an issue within AI-guided testing workflow
Step 2: Tester writes bug report: "Fee not calculating correctly on business checking"	**Step 2:** AI generates a detailed bug report including exact steps to reproduce, expected behavior based on business rules, actual behavior observed, specific accounts or transactions affected, related configuration settings, and potential root cause analysis
Step 3: Developer receives report and has no idea what was tested, what was expected, what actually happened, or how to reproduce it	**Step 3:** Developer receives complete context to begin investigation immediately
Step 4: Multiple back-and-forth exchanges waste days clarifying the actual issue	**Step 4:** Consistent formatting allows pattern recognition across defects
Step 5: Poor documentation means similar bugs get reported multiple times because no one recognizes they are duplicates	**Step 5:** AI identifies potential duplicate bugs before they are filed

Real-World Impact: A testing team found that response time to bugs decreased from an average of 3.2 days (mostly spent clarifying the issue) to 0.8 days with AI-generated defect reports. The number of "cannot reproduce" bug rejections drops from 23% to 4%.

Test Script Generation for Automation Tools

Traditional Approach After manually creating test cases, teams wanting to use test automation tools must:	AI-Powered Approach The system automatically generates automation scripts:
Hire specialized automation engineers who are expensive and scarce	AI translates test cases into executable scripts for common testing tools
Have engineers manually translate test cases into automation scripts	Automation coverage can extend to hundreds or thousands of test cases economically
Wait weeks or months for automation framework development	Scripts are generated in parallel with test case documentation
Limit automation to highest-priority test cases due to development cost	Teams without specialized automation engineers can leverage automated testing

Real-World Impact: A community financial institution wanted to automate regression testing but estimated it would require two full-time automation engineers for 6 months at $180,000. AI-generated test scripts provided 85% of the desired automation coverage at a fraction of the cost, allowing it to run comprehensive regression tests after each configuration change.

The Productivity Transformation

The real breakthrough isn't automated test execution, it's removing the documentation bottleneck that has always limited testing scope.

Traditional Timeline	AI-Powered Timeline
Weeks 1-16: Business analysts create user stories and test cases	**Weeks 1-3:** AI generates comprehensive user stories and test cases. The team reviews and refines
Weeks 17-24: Testers execute tests, struggle to document bugs clearly	**Weeks 4-12:** Testers execute tests. AI documents bugs with complete detail
Weeks 25-32: Developers investigate vague bug reports, testers re-explain issues	**Weeks 13-16:** Developers efficiently address well-documented issues
Result: 32 weeks to complete testing with limited coverage	**Result:** 16 weeks to complete testing with superior coverage

> The 50% timeline compression comes not from testing faster, but from eliminating documentation work that consumed half the schedule.

The Executive Visibility Transformation

Traditional conversions force launch decisions based on: *"We've completed 87% of planned test cases. The remainder of the outstanding issues are mostly minor."*

> **AI-powered platforms provide quantifiable metrics:**
>
> → "We have test coverage across 78% of system functional capabilities, with gaps in these specific areas"
>
> → "23 unresolved issues remain. AI analysis categorizes three as high-risk based on customer impact"
>
> → "Bug resolution velocity has increased by 65% with AI-generated defect reports"
>
> → "Coverage analysis shows adequate testing in all high-risk functional areas"

When your board asks *"Are we ready?"*, you answer with data rather than educated guesses. You can demonstrate objectively that you've tested the capabilities that process 92% of transaction volume and identified the 8% gap as low-risk, infrequently used features.

Deep Dive: AI-Powered Customer Communication

The Traditional Communication Failure Pattern

Most institutions approach conversion communication with good intentions but predictable results:

- In the **announcement phase**, mass communication ignored by 95% of customers
- In the **preparation phase**, generic "what to expect" guidance confuses more than it clarifies
- During the **conversion weekend**, customers are unprepared for specific service needs
- At **post conversion**, the call center is overwhelmed by extraordinary volume
- In **recovery**, you have months of reactive issue response and relationship repair

This pattern reflects the impossibility of executing personalized communication at scale manually. A $1B institution with 40,000 relationships and hundreds of product combinations creates customer cohorts that are economically impossible to address individually with traditional approaches.

What AI-Powered Communication Actually Enables

Platforms like Flare's **LaunchIQ** change what's possible.

Predictive Customer Concern Identification

Instead of assuming all customers share the same concerns, AI analyzes behavior patterns to forecast specific issues before they arise and pinpoint vulnerability factors. This approach identifies customers who are most likely to be confused or frustrated by changes.

Dynamically Generated Personal Content

Instead of 5-10 generic templates, AI generates customer-specific messages incorporating their actual products, usage patterns, and concerns. A business customer receives: "Sarah, your business checking and merchant services will have enhanced online banking on September 15th. The positive pay service you use for payroll will work the same way, but you'll access it through 'Fraud Controls' rather than 'Services.'"

Channel Orchestration

Based on which channels each customer engages with, AI orchestrates multi-touch campaigns with delivery parameters optimized for that audience.

Project Management

Beyond defining content and channels, the system organizes all communication activities into an actionable project management platform. Tasks, deadlines, assignments, and status tracking ensure accountability across the conversion team. Built-in collaboration tools allow marketing, operations, and relationship managers to coordinate seamlessly, ensuring no customer segment falls through the cracks during the critical conversion timeline.

The Measurable Impact

Traditional conversions see call volume spike to 400-600% of normal in the first week post-conversion. Institutions using AI-powered communication report 180-220% spikes. This represents 3,000-5,000 prevented calls, between $45,000-$75,000 in avoided temporary staffing costs, and preserved relationship goodwill that's impossible to quantify but critically real.

Deep Dive: AI-Powered Staff Training

The Forgetting Curve Reality

Traditional training assumes humans can absorb and retain months of information about processes they haven't used.

Psychological research reveals humans forget 50-80% of newly learned information within 24 hours and retain less than 20% after one month.

Yet conversion training typically trains tellers two months before conversion and expects them to remember this training during high-pressure customer interactions. The equation doesn't equal.

What AI-Powered Training Actually Enables

Platforms like Flare's **TrainingIQ** fundamentally change when and how learning occurs.

Just-in-Time Learning

Instead of front-loading everything employees might need, AI provides specific guidance exactly when needed through contextual assistance. A teller processing a complex deposit receives instant guidance on the exact scenario they're handling thereby learning at the moment when retention is highest.

Individualized Learning Paths

Rather than one-size-fits-all training, AI recognizes each employee's role, experience, and proficiency level, generating personalized paths focusing on individual gaps. A 15-year veteran receives "what's changed" training; a two year employee receives comprehensive system navigation.

Competency Verification

Instead of measuring attendance, AI continuously assesses actual task completion ability, identifies knowledge gaps through error pattern recognition, and provides targeted remediation before errors impact customers.

Contextual Performance Support

Rather than searching procedure manuals, staff receive relevant guidance proactively within their workflow, maintaining productivity while building proficiency.

The Productivity Impact

Traditional conversions see productivity drop to 40-50% in the first month post-conversion, gradually recovering over the next 4-6 months. AI-powered training maintains 70-75% productivity initially with recovery to 95%+ within two months representing thousands of recovered staff hours and faster return to your business’s strategic initiatives.

The Economics: Validating the 80% Claim

Consider this example. A $2B community financial institution is undertaking a core conversion.

Traditional Conversion	AI-Powered Conversion
Direct project costs: $8,500,000	**Direct project costs:** $6,190,000 including a $240,000 AI platform licensing fee
Indirect operational costs: $4,850,000 (productivity loss, temporary staffing, customer attrition, error remediation, to name a few	**Indirect operational costs:** $1,070,000
Risk materialization (probability-adjusted): $240,000	**Risk materialization** (probability-adjusted): $6,500
Total Cost: $13,590,000	**Total Cost: $7,266,500**

Net Savings: $6,323,500 | 47% total cost reduction

Breaking Down the 80% Claim

The "80%" specifically addresses controllable conversion execution costs.

Testing Efficiency Gains
50% timeline compression through documentation automation; 70% reduction in testing labor costs; 65% faster bug resolution through clear defect reports

50%
testing timeline compression

70%
testing labor reduction

65%
faster bug resolution

Risk Reduction
97% reduction in expected value of major defects through early detection and objective coverage visibility

Timeline Compression
47% overall timeline reduction (30 months to 16 months from start to stabilization)

Weighted Aggregate
When combining risk reduction (97%), risk-adjusted cost reduction (62%), and timeline compression (47%) with appropriate weightings, the result approaches 80%. Additional factors include opportunity cost recovery, customer lifetime value preservation, and employee retention push beyond this baseline.

ROI Calculation
A $240,000 AI platform investment generates $6,323,500 first-year savings = 2,635% ROI

The Decision Framework: Is Now the Right Time?

Assessment Questions

1. What did your last conversion cost in dollars, timeline, and organizational impact?

If "more than planned and quite disruptive," AI directly addresses your pain points.

2. How much documentation work is your team facing?

If you're looking at 100+ test cases to write manually, AI automation provides immediate value.

3. How confident are you about test coverage?

If you can't objectively say what percentage of system functionality you've tested, AI visibility transforms this decision-making.

4. How would your Board react to significant overruns?

If "that would be a Board and management crisis," you need AI's risk mitigation and objective metrics.

5. What's your institution's appetite for operational innovation?

If "technology adoption is a competitive advantage," early AI adoption aligns with your culture.

Three Paths Forward

Wait 2–3 Years	Pilot Now	Full Adoption
✓ Last conversion was successful	✓ Last conversion had significant issues	✓ Traditional conversion risks are unacceptable
✓ 2+ years before converting	✓ 12-18 months from conversion	✓ 6-18 months from conversion
✓ High Board risk tolerance that is conservative about technology adoption	✓ Have moderate risk tolerance, and are open to innovation with safeguards	✓ Have strong executive sponsorship, and technology leadership as strategic imperative

Conclusion

Core conversions have been banking's highest-risk project for decades. The 75% timeline overrun rate represents not poor execution but structural impossibilities and are primarily the documentation bottleneck that has always limited testing scope and quality.

AI Changes What's Possible

By handling work that shouldn't require human intelligence:

- → Generating comprehensive test documentation
- → Personalizing customer communications at scale
- → Tracking objective functional coverage
- → Creating detailed defect reports
- → Delivering contextual staff guidance

The Early Adopter Advantage

50%	70%	✦
Timeline compression	Attrition Reduction	Data-Driven Insights

Early adopters are discovering testing timelines that compress by 50%, customer attrition that drops by 70%, and objective visibility into conversion readiness that transforms Board discussions from educated guesses to data.

The Reality

Every institution will eventually adopt AI-augmented approaches. The question isn't whether but when.

Early adopters capture immediate economic benefits, competitive timing advantages, and organizational AI competency. Later adopters benefit from reduced technology risk and proven platforms.

But there is wrong timing: undertaking a traditional conversion after AI-powered alternatives have proven their value represents a choice to accept preventable risk, unnecessary cost, and avoidable disruption.

The Next Step

If your institution is planning a core conversion, the prudent action is evaluation:

- → Request demonstrations
- → Speak with reference customers
- → Model your specific economics
- → Assess integration requirements

The institutions that thrive will thoughtfully evaluate new capabilities, adopt what creates value, and continuously evolve as technology enables what was previously impossible.

Core conversions represent a defining test of operational sophistication. The question is **whether your next conversion demonstrates yesterday's limitations or tomorrow's capabilities**.

Ready to Transform Your Core Conversion?

Contact Flare today for a no-obligation assessment of your upcoming or ongoing core conversion.

Take Action Now

See what's actually possible—not just theoretically, but in real deployments across institutions like yours.

- ↗ Request a product demonstration
- ↗ Review your specific use case with our experts
- ↗ Model the tangible cost, risk, and timeline savings for your

theflare.ai sales@theflare.ai

FlareAI

Take the first step to turning core conversions from an existential risk into a manageable, high-success business initiative.