curiosity

www.curiositysoftware.ie

Case Study:

How Eljin Achieved 8x Faster Test Creation for Their Business-Critical Payment System

eljin productions

The largest payer of union audiobook talent in North America was rebuilding its business-critical payment platform from the ground up.

eljin

Eljin required a rapid solution for optimised test creation, capable of generating targeted tests from among a vast range of complex payment scenarios.

curiosity

Quality Modeller auto-generates rigorous in-sprint tests from flowcharts that double up as living documentation for the complex payment system.

The Benefits at a Glance

- An initial 8x increase in test creation speed.
- Automated and on demand test generation thereafter.
- Accelerated test maintenance, providing sprint-over-sprint efficiency gains.
- Optimised testing and targeted coverage, catching bugs in-sprint.
- A move from unmeasured, unknown coverage, to testing that is as “exhaustive” as needed.
- A reduction in wasteful over-testing, focusing on logically distinct test scenarios.
- Several critical production bugs found during the first three test runs.
- A significant reduction in time spent on bug remediation and debugging tests.
- Synthetic test data generation avoids data privacy risks in testing.
- Living documentation maintains knowledge and boosts collaboration.
- Up-to-date tests and documentation avoid technical debt and “future proof” development.

“Our payment application lets publishers and producers pay union performers in the US and Canada. If transactions go wrong, there’s trouble from all directions—from actors and publishers, talent agents and the unions themselves. CSI’s models tested our business logic aggressively. They ensured the reliability of complex connections between user types, so payments stay on target and private data remains secure. The CSI team dove deep into complicated calculations, report generation, and automated payment components. They identified minor and major bugs, so we could bring our product to market quickly—and guarantee our users a simple, worry-free experience. CSI’s contribution made all the difference.”

John McElroy

President of Eljin Productions

curiosity

Bringing a First-of-Kind Payment Platform to Market

Eljin, the largest payer of union audiobook talent in North America, was rebuilding its pioneering platform for paying and tracking audiobook talent. It was rebuilding the platform from the ground up, requiring robust and continuous testing before and after its launch.

The first-of-its-kind payment platform significantly simplifies the payment of fees to audiobook narrators. Enterprises, narrators, agents and publishers use the platform to pay and receive money for performances in the US and Canada, including for cross-border work. Eljin automatically calculates and tracks the complex payments, resolving talent-agency fees, currency conversion, union contributions, taxes, and more.

The platform substantially simplifies the process and effort associated with paying unionized talent. It is Eljin's core offering and unique selling point, while Eljin's revenue is built on making high volumes of accurate payments. The quality of the platform is business critical, demanding rigorous and continuous testing.

Vastly Complex Payment Rules Demand Rigorous Testing

Development work for the new Eljin platform began in June 2020. Aware that rigorous testing must occur in parallel to rapid development, Eljin faced a challenge: How could they test the complex payment calculations rapidly, rigorously, and with an attainable volume of test cases?

Highly Complex Test Logic

The possible test combinations associated with the Eljin platform are vast. This combinatorial complexity is due to the rules associated with calculating talent payments. Each payment must consider factors like pension contributions, taxes, US and Canadian law, union fees, currency conversion, and more.

These factors each contains their own set of highly complex logic. When combined in a payment system, they lead to an explosion in the different types of payment to test.

Under-Testing: Too Great a Risk to The Business

Exhaustively testing every change to the Eljin platform is therefore not a viable test strategy, given the pace of development and the number of possible test cases associated with Eljin's payment system. Yet, under-testing the business-critical platform is simply not an option either.

Under-testing the Eljin platform would introduce a range of commercial, legal, and operational risks. Each would threaten Eljin's bottom line, potentially devastating a fast-growing start-up. The risks of poor testing include:

1. The legal and financial repercussions associated with making incorrect payments.
2. Revenue lost in waived fees due to making inaccurate payments.
3. The time lost in development as production defects take substantially longer to diagnose and fix.
4. Reputational damage, customer churn, and slow platform adoption.
5. Security vulnerabilities on a system that processes the most sensitive types of data. This includes Social Security Numbers, Social Insurance Numbers, and information related to Union Membership.
6. Damage to long-standing relationships between Eljin's CEO and its enterprise customers. This would impact relationships built over decades in the production industry.

The Challenge of Manual Test Design

Eljin required a rapid and optimised approach to test creation, rigorously validating a wide range of payments in the smallest possible volume of tests. This is why they approached Curiosity Software.

Before Curiosity joined the project, test creation was being performed manually by Eljin's development teams. Engineers used an Excel spreadsheet to formulate data-driven tests for the highly complex payment scenarios.

The Excel spreadsheet contained a range of interrelated sheets and formulae. These intricate formulae were responsible for calculating the expected results for different payments, based on the test data entered by the engineers.

The spreadsheet was hard-to-

understand and hard-to-maintain. During development, there was also little time to formulate new scenarios manually in the complex spreadsheet. Consequently, the spreadsheet contained just 12 different scenarios, falling far short of the number of combinations associated with the complex payment system.

Manual, Unsystematic Test Design Hindered Development

Relying on manual, unsystematic test design in Excel created several testing challenges. Each introduced unacceptable risk as Eljin brought its new platform to market. These challenges included:

1. **Inaccurate Test Creation.** It was impossible to define test cases and expected results accurately in the unwieldy spreadsheet.
2. **Unmeasurable Testing.** It was impossible to measure the coverage of testing, or to know with confidence when testing was "done" before an update.
3. **Under-Testing.** Testing lacked sufficient rigour. The total possible test scenarios contained in the spreadsheet fell far short of the total number of possible payment scenarios. Testing therefore had no way of creating all the tests that might be needed for the Eljin platform.
4. **False negatives.** If testing generated an unexpected result, developers struggled to know if the defect was genuine, or a flaw in the hard-to-manage spreadsheet.
5. **Growing technical debt.** Generating new scenarios in the sheet was simply too complex and laborious. This undermined the creation of new test scenarios, risking growing technical debt.

Models Generate Rigorous and Maintainable Tests

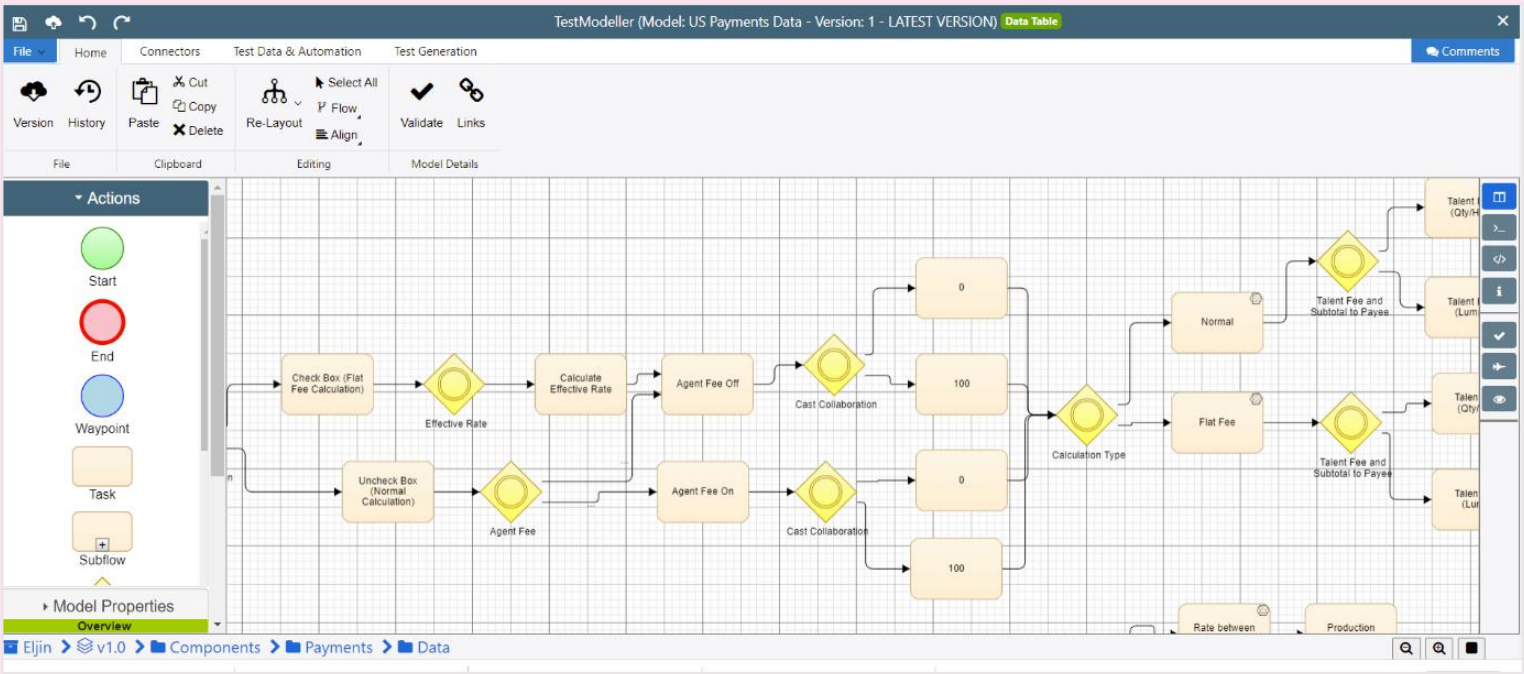
Eljin brought Curiosity into the testing process early in the project, testing before the first version of the platform went live.

Two Curiosity consultants now use Quality Modeller to generate rows of "covered" test data for the Eljin platform. They apply automated optimisation algorithms to intuitive visual flowcharts, identifying the smallest set of test cases needed in testing. This optimised test creation generates "covered" sets of data to run in testing, rapidly and rigorously testing the fast-evolving Eljin system.

Automated and Optimised Test Creation

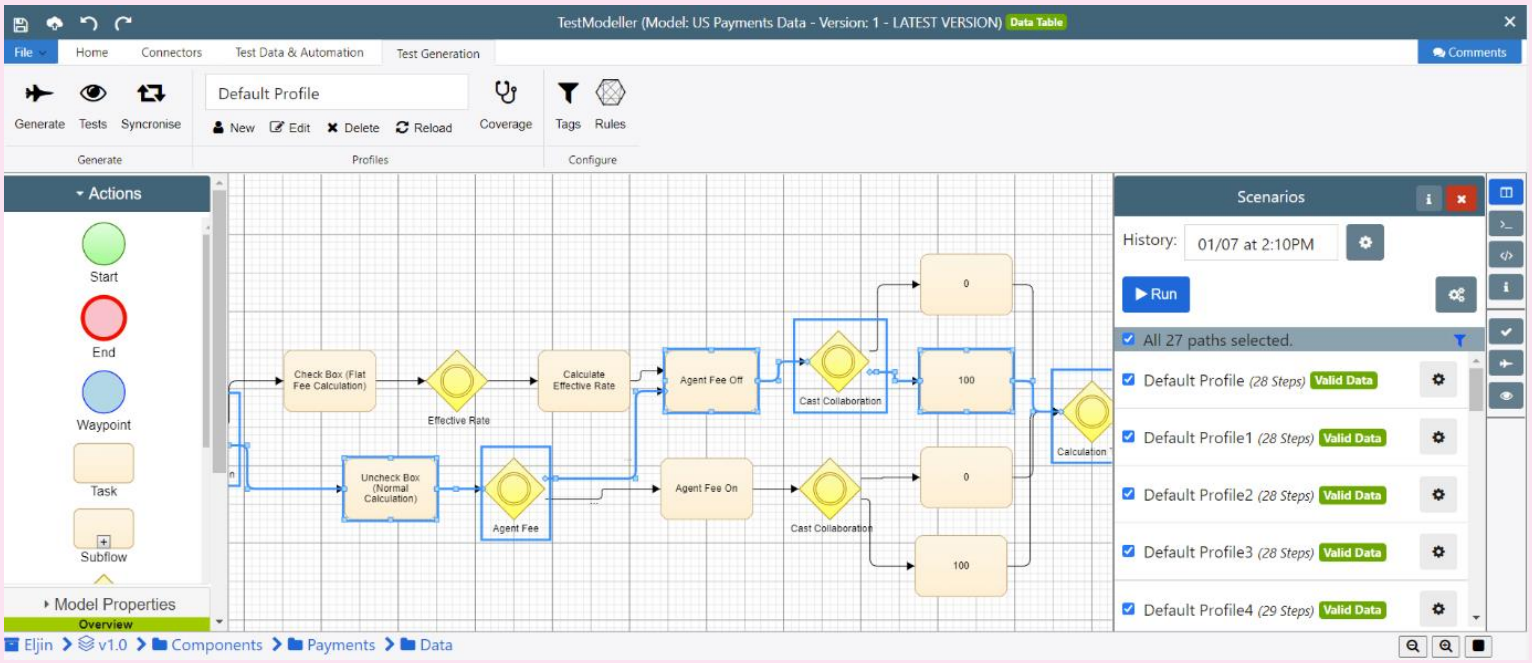
The optimised tests are generated from intuitive flowchart models in Quality Modeller.

The visual flows clearly map out the data variables and equivalence classes associated with the different payment scenarios. They replace the unwieldy Excel spreadsheet previously used by Eljin's developers with an intuitive approach to rigorous test creation, while also providing a collaborative resource for understanding the complex system logic:



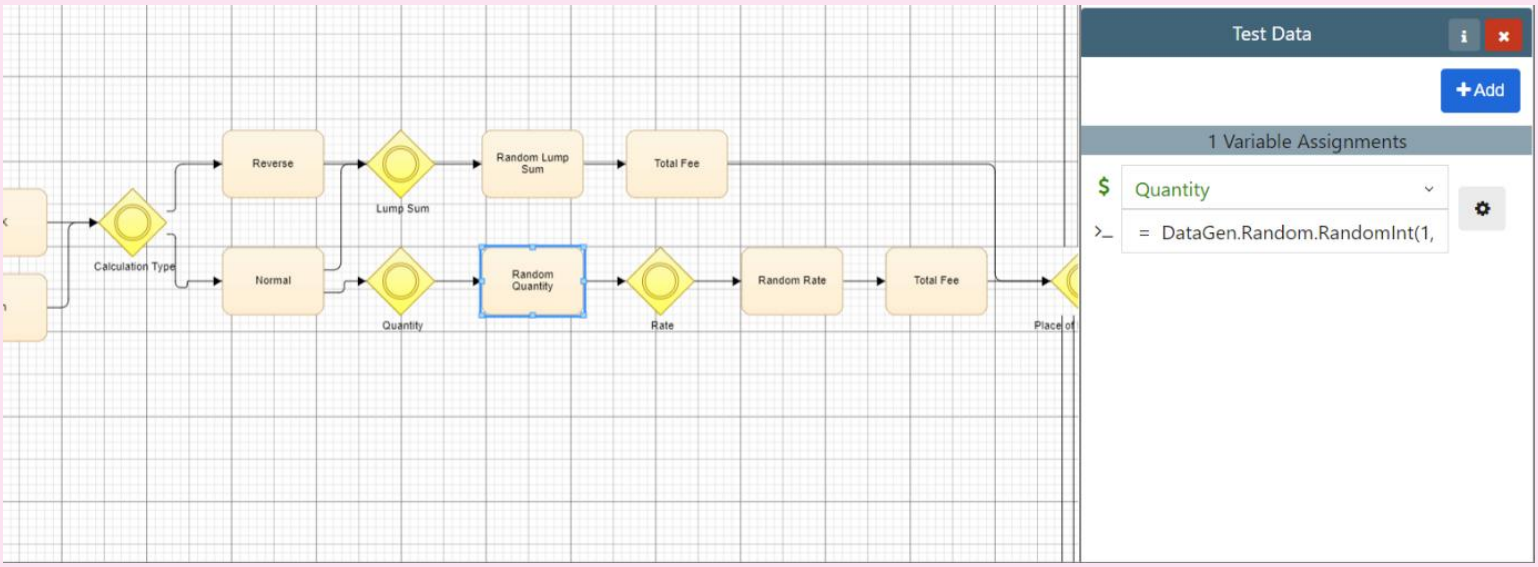
Intuitive flowcharts maintain a clear understanding of complex payments logic and generate optimised data for rigorous testing.

Targeted test generation then creates optimised tests automatically for the Eljin platform. The automated coverage algorithms generate the smallest set of combinations needed to "cover" the modelled logic, creating a "rightsized" set of test data for de-risking system change.



A set of optimized "paths" have been generated through a model of the Eljin system. Each path represents a test case, generating data to a test a complex payment scenario.

The optimised test data is generated from static values defined in the flowchart, combined with dynamic data generated by synthetic test data functions. Eljin today leverage combinable data generation functions to create rich synthetic data for their tests. The intuitive, easy-to-define functions resolve as tests are generated, creating consistent and compliant data sets:



Synthetic data generation functions defined at the flowchart level combine to create consistent test data for the complex payment platform.

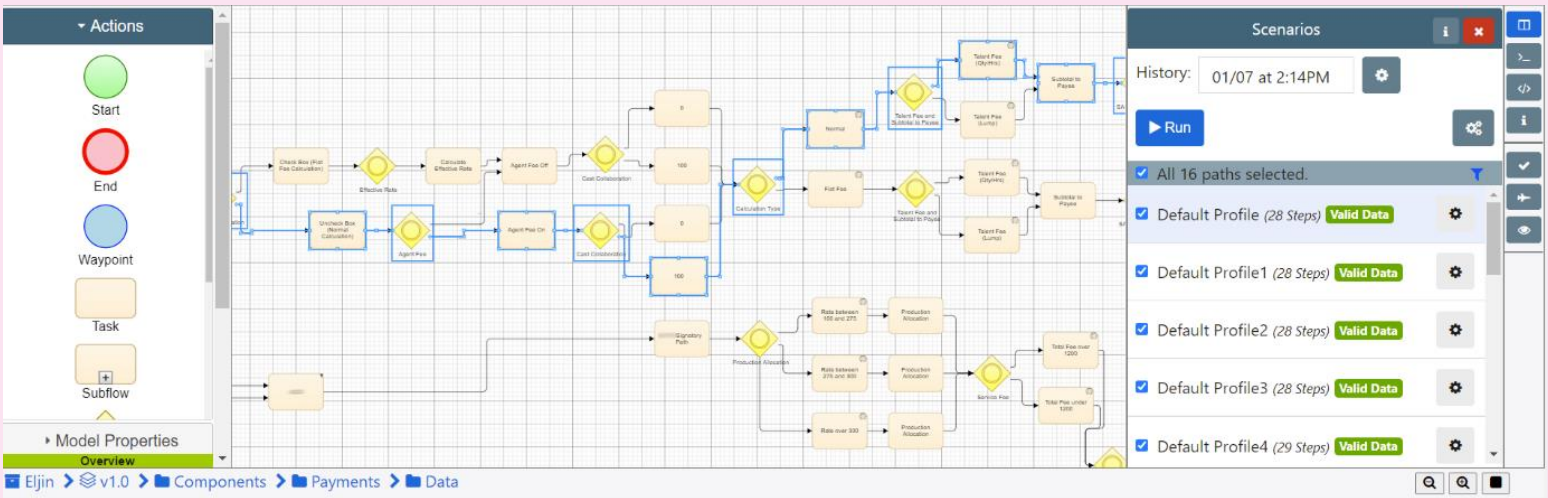
Targeted Test Coverage Based on Time and Risk

For a typical test run today, the automated generation creates around 100 targeted rows of data. Each row represents a distinct test combination, reflecting a substantial advance on what Eljin's developers were using previously. Quality Modeller has accordingly increased test coverage substantially, reducing the negative risk associated with under-testing.

The visual test models contain far more logical combinations than the 100 targeted rows of data generated for a typical test run. Eljin can today dial up or dial down the desired test coverage based on time and risk.

This targeted test generation in turn maximises the impact of testing, while testing rapidly and iteratively with a lean set of tests. Logically repetitive scenarios are avoided during the optimised test creation, reducing wasteful over-testing in favour of testing an optimised set of distinct scenarios.

Eljin can now switch between coverage profiles in a model to generate an optimal set of tests for different developer needs. For instance, manual test execution might not want to run large test suites for every change made to the Eljin platform. Smoke testing might similarly generate a smaller set of tests, for instance those that pass through each node in the model once.



16 paths have been generated from a logically complex model, applying automated coverage algorithms to create an optimized set of tests.

At other times, testing can focus on at-risk functionality, increasing the number of paths that pass through nodes associated with test failures or recently updated system logic.

This ability to generate targeted tests automatically saves Eljin significant time during testing and development. Meanwhile, the optimised testing has found several critical bugs that might have devastated production systems. It has moved Eljin from unmeasured, unknown coverage, to testing that is as "exhaustive" as they need it. This catches potentially costly bugs while they are quick and easy to fix, de-risking new releases.

Benefits of Using Quality Modeller at Eljin

8x Faster Test Creation

The automated, model-based test generation is not only rigorous, measurable, and risk-based; it is also far faster than the manual test design used previously by Eljin.

It took Eljin developers one day to create the initial Excel spreadsheet used to create tests. This spreadsheet produced data for just 12 different test scenarios.

It took two Curiosity consultants just 1 day to spreadsheet in a Quality Modeller flowchart. This created an initial model that could generate over 100 distinct test scenarios, representing an eightfold increase in test creation speed.

Growing Time Savings Sprint-Over Sprint

While the initial 8x speed gain might seem impressive, the automated test creation is saving even more time sprint-over-sprint. This stems from time saved during ongoing test creation and maintenance, as well as during test execution.

Rapid Test Creation

Instead of entering data to create tests for each new test run, visual flowcharts now generate targeted test cases for each test suite. This automated process is far faster than entering data manually into a spreadsheet to calculate expected results. Test creation occurs at the click of a button, and can further be exposed to CI/CD processes.

Accelerated Test Maintenance

When something changes in the Eljin platform, test maintenance is likewise accelerated with Quality

Modeller. It is far faster and more reliable to update a logically precise visual flowchart at Eljin, than it was to edit an intricate spreadsheet.

For instance, if an upstream variable like a payment type changes, testing only needs to update this in a single node or part of the models. Automated test generation then creates all the tests and downstream logic needed to reflect this wide-reaching change, saving the time otherwise spent maintaining complex tests one-by-one.

This rapid maintenance and automated test creation unlocks substantial sprint-over-sprint time savings, while achieving the test coverage needed for the Eljin platform. The rapid test maintenance further future proofs the Eljin platform. If something changes, such as a tax rate in a Canadian province, the model can react quickly to this change to generate new tests. Rigorous testing does not then present a barrier to the rapid development of the Eljin platform.

Reduced Test Run Times

Dialling test coverage up and down based on risk saves further time during test execution, as it rightsizes testing based on time and risk. Eljin can reduce overall test volume with each test run, all while increasing test coverage, reducing negative risk, and catching potentially costly bugs early.

Early Bug Detection Avoids Devastating Costs to The Business

Using Quality Modeller, Eljin have found several bugs that might have otherwise devastated their business-critical payment platform. Functional bugs have been found during rigorous test execution, while the process of modelling the Eljin platform has uncovered additional defects.

Modelling Identifies Missing or Erroneous Logic

Visually modelling the Eljin platform has identified gaps and errors in logic that were previously reflected in Eljin's pre-production systems.

Today, Quality Modeller's intuitive flowchart models clearly map out the logic associated with payments made by the Eljin platform. Modelling the spreadsheet used previously to create test data made it easy to stop missing

logic. Inaccurate and potentially buggy logic likewise became clear, as it created conflicts that needed resolving during the modelling process.

The migration from the Excel spreadsheet to Quality Modeller flowcharts in turn identified several bugs which had been replicated by developers in Eljin's pre-production systems.

Some of these bugs were critical, representing a serious threat to Eljin's platform and its business. For example, certain differences in US and Canadian tax variables had not been factored into the initial versions of the system. These variables impacted numerous downstream payment calculations, leading to the miscalculation of fees like those paid to a talent agent.

Optimised Testing Finds Even More Bugs First Time Round

As Eljin run the optimised tests generated in-sprint by Quality Modeller, they have found even more critical bugs before they can have a significant impact on the payment platform.

The first three test runs found 5 high severity bugs and 10 minor bugs. Had these bugs been undetected, they would have constituted a significant threat to Eljin's first release. One critical bug found in the test environment involved an agent being paid twice. Another payment combination blocked connection to the accounting software used by Eljin, preventing the submission and processing of payments.

Rapid Defect Remediation

The early bug detection using Quality Modeller has reduced defect remediation time and costs at Eljin, as it catches potentially devastating bugs before they can impact substantial chunks of code.

Defect remediation efforts have been further reduced by Quality Modeller's intuitive flowcharts, which made it quick and easy to locate the root cause of defects. Quality Modeller overlays run results from testing visually onto flowchart models, allowing developers to pinpoint the exact nodes which failing tests pass through.

Engineers at Eljin today can identify the exact columns and variables to focus on when fixing bugs. This is far faster than analysing the unwieldy spreadsheet used previously to design test cases, in which it was unclear if the error lay in the developed system or in the

spreadsheet. Developers can instead locate the exact columns and variables that are associated with failing tests, rapidly locating the payment variables that require immediate attention.

A Collaborative Resource and Shared Understanding

While driving rigorous and optimised testing, the flowcharts also provide up-to-date “living documentation” for testing and development work.

Stakeholders in different roles can work collaboratively from the clear specifications. The flowcharts thereby support close cross-team collaboration and communication, while “future proofing” the Eljin platform. Future development work can leverage the knowledge preserved in the flowchart models, rapidly and reliably evolving the Eljin system.

With Quality Modeller, knowledge of the complex payment rules does not remain in people’s heads. It becomes readily accessible to everyone in the Eljin team, including newly onboarded testers and developers.

A Successful Release – And Many More to Come

The first version of the rebuilt Eljin platform launched successfully in November 2021, and Eljin has since continued processing payments for a wide range of enterprises, narrators, agents and publishes.

The business-critical platform is embarking on a rapid roadmap of development and innovation, supported by rigorous in-sprint testing. This rapid and robust testing is underpinned by Quality Modeller. Quality Modeller has introduced automated and optimised test generation, along with a range of benefits to Eljin’s testing and development. These benefits include:

- An 8x increase in initial test creation speed.
- Automated and on demand test generation
- Accelerated test maintenance, providing sprint-over-sprint speed gains.
- Optimised testing and targeted coverage, catching bugs in-sprint.
- A move from unmeasured, unknown coverage, to testing that is as “exhaustive” needed.
- A reduction in wasteful over-testing, focusing on logically distinct test scenarios.

curiosity

- Several critical production bugs found during the first three test runs.
- A significant reduction in the time spent on bug remediation and debugging tests.
- Synthetic test data generation avoids data privacy risks in testing.
- Living documentation maintains knowledge and boosts collaboration.
- Up-to-date tests and documentation avoid technical debt and “future proof” development.

For more information on Curiosity’s Quality Modeller and how we can help you automate your testing, visit our website or email us.

curiosity

www.curiositysoftware.ie

info@curiosity.software

Curiosity Ireland

Unit 6 The Mill, The Maltings, Bray,

Co. Wicklow, A98 XV40, Ireland

Curiosity USA

4136 Del Rey Ave., Suite 658

Marina Del Rey, CA 90292

USA