Success Story

EVERFI's 9-month journey

From manual testing to generating new tests in minutes!

Learn more at curiositysoftware.ie

Contents

Test generation in minutes!

EVERFI

An award-winning provider of cloud-based eLearning, that delivers hundreds of courses to millions of learners globally.

cypress

Required scalable course automation, capable of testing vast permutations at speed, across devices, and with visual and accessibility testing built-in.

curiosity

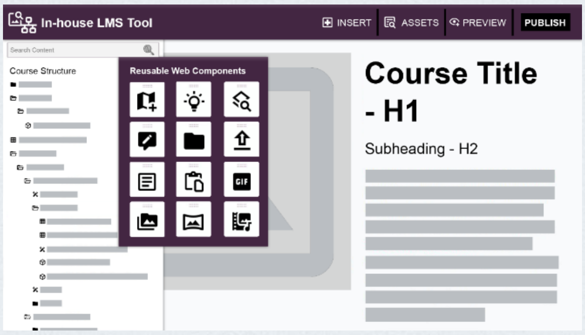
EVERFI's CI/CD pipeline today generates targeted tests within minutes of a course check-in, automatically analysing course schemas to generate flowcharts and tests.

The benefits at a glance

- 85% automation of course testing, saving resources previously spent on outsourced manual testing.
- 100% course permutation coverage, assuring quality, with functional and visual testing of every page.
- Up to ~30,000x faster test creation – from weeks to minutes.
- Tests run in minutes, with concurrency of up to 90 machines.
- Automated test maintenance takes minutes following a new course check-in.
- Close collaboration between Quality Engineering, development, product owners and designers.
- Clear and documented understanding of what's being tested.

Componentisation: A challenge and opportunity for quality engineering

Course development at EVERFI is rapid, leveraging a library of reusable components to create online courses in EVERFI's eLearning platform. Developers use an in-house Learning Management System (LMS) to combine reusable Backbone.js components:



A purpose-built content management platform enables rapid course development at EVERFI.

These components form pages, which in turn create modules that make up 100+ courses. Each module has up to 30 pages, while the average course has 75 pages:

This reusability ensures the scalability of course development. Yet, it creates a "combinatorial explosion" that must be tested at speed. Testing must cover millions of possible user journeys through the courses. It must further validate the journeys across different types of network (corporate or K-12), learner (supervised or unsupervised), locations (state and country), learning management system (EVERFI's or third party), devices, and resolutions.

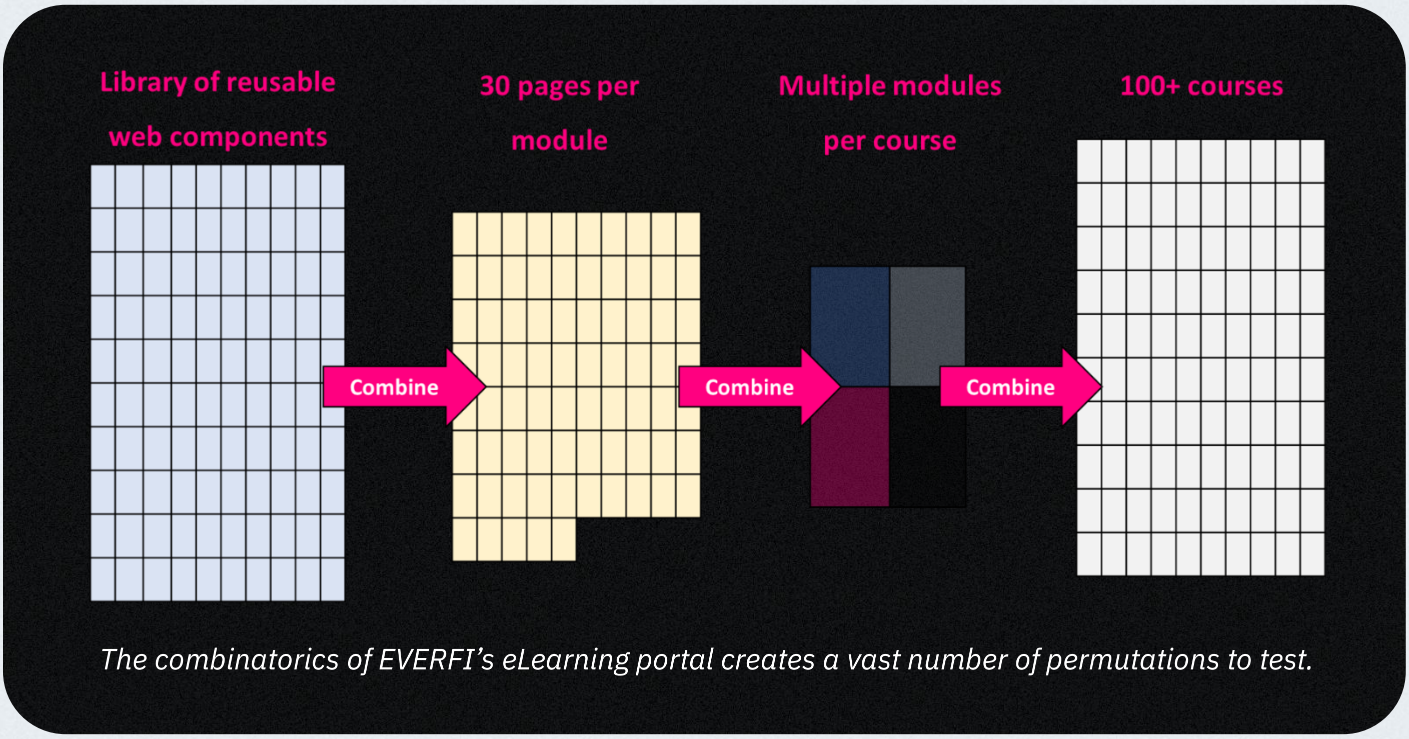
> **"As we embarked on a journey to scale test automation across the whole enterprise, we knew we needed to match the pace of course development, while fostering close collaboration with developers and product owners. Curiosity platform's intuitive flowcharts break our system down into reusable chunks that generate the functional and visual tests we need. We can further assemble these reusable building blocks automatically as new courses are checked in, auto-generating rigorous end-to-end tests and providing a collaborative picture of what's being tested."**

Greg Sypolt

VP of Quality Engineering

Sneha Viswalingham

Director of Quality Engineering


Library of reusable web components
30 pages per module
Multiple modules per course
100+ courses
Combine
Combine
Combine

The combinatorics of EVERFI's eLearning portal creates a vast number of permutations to test.

Only rigorous testing of all of these journeys can support EVERFI's mission of addressing education's missing learning layer, providing courses for financial education, workplace training, community education, K-12 learning, and beyond.

Moving from manual to automated testing

Before the introduction of a Quality Engineering (QE) mindset, course testing at EVERFI was made up wholly of manual testing. A team of manual testers worked alongside manual exploratory and crowd testing, which was outsourced to a range of vendors. However, this was too time-consuming and costly. Suboptimal testing further risked hitting the same logic repeatedly, exposing EVERFI's eLearning platform to bugs, while there was no way of communicating clearly what was being tested to developers or product owners.

The need for scalable automation

EVERFI therefore built and now maintain a best-of-breed functional test automation framework, created using leading test automation technology. The Cypress framework is built using best practices for reusability like page object architecture.

Course automation required a scalable approach to creating tests for this Cypress framework, capable of testing the vast permutations created by course creation.

The Quality Engineering team decided that hand-scripting every test would be too slow to test the modularised system architecture. They sought a rapid and automated approach to reusing the high-quality Cypress automation, avoiding the manual effort of copying, pasting and editing boilerplate code. The automated test generation would further need to guarantee sufficient test coverage, while avoiding copious test maintenance and communicating clearly what's being tested to stakeholders without coding skills.

Digital tranformation: Key project goals

To match the pace and complexity of course development, QE embarked on a digital transformation project to remove manual testing from 100+ courses. The project sought to scale rigorous test automation across the whole organization. There were two core goals:

1. **Move faster.** The initial goal was generating course automation as quickly as possible, removing testing bottlenecks to match the pace of development.
2. **Optimise and measure coverage.** To assure quality at speed, the automated testing would further need to achieve sufficient coverage. QE set out to achieve optimal functional test coverage of the course permutations, including navigating UI components by mouse and keyboard when testing the components in isolation. Visual testing would further validate how users experience these different pathways, while A11y testing would be required to ensure accessibility. All of these quality factors would furthermore need testing across different devices and resolutions.

EVERFI identified two supporting goals as paramount to achieving this quality at speed:

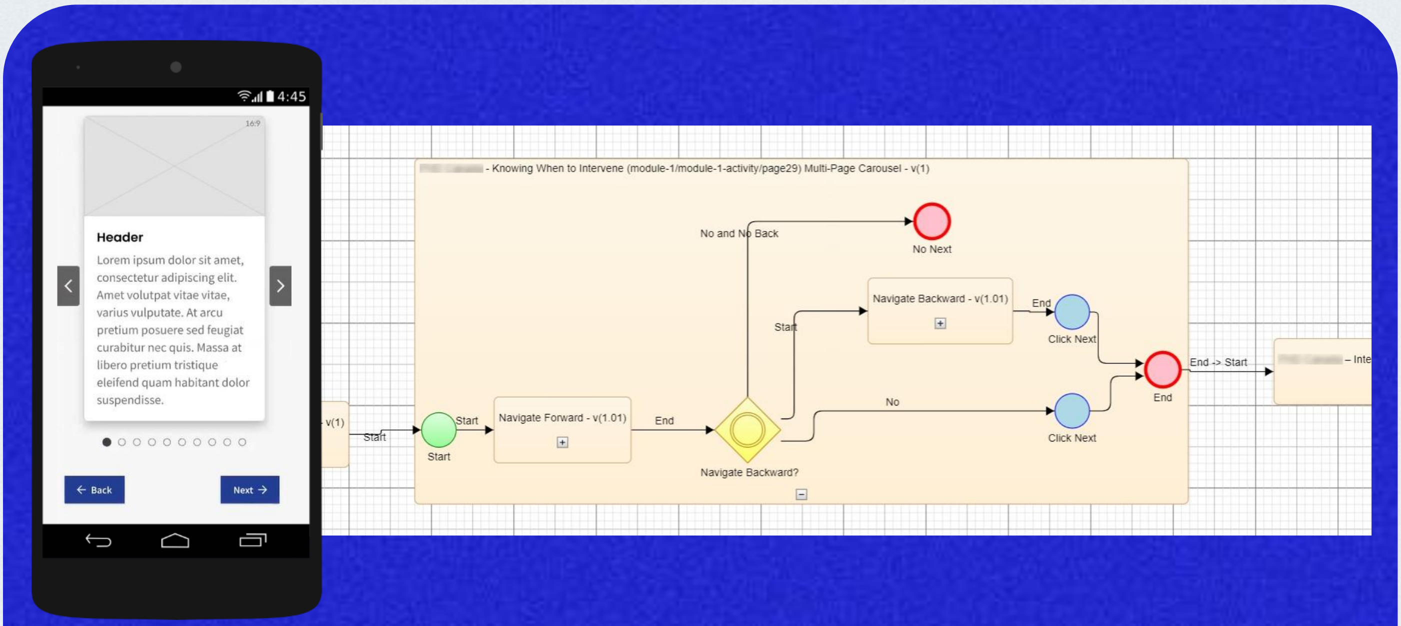
3. **A closer relationship with product owners and development.** QE needed to shift left, while communicating clearly what is being tested.
4. **Upfront thinking about quality.** Quality needed to become everyone's responsibility, with buy-in across teams at every stage of course development.

Generating optimised tests at speed

The project selected model-based test generation to create course code as quickly as possible, while optimizing and clearly communicating test coverage. QE chose Quality Modeller, part of Curiosity's platform, as it would break EVERFI's courses down into reusable flowcharts, combining them to generate course automation quickly. Its visual flows would further communicate clearly what is being tested, while the intuitive BPMN-style modelling would enable rapid training.

Modelling course components in Quality Modeller

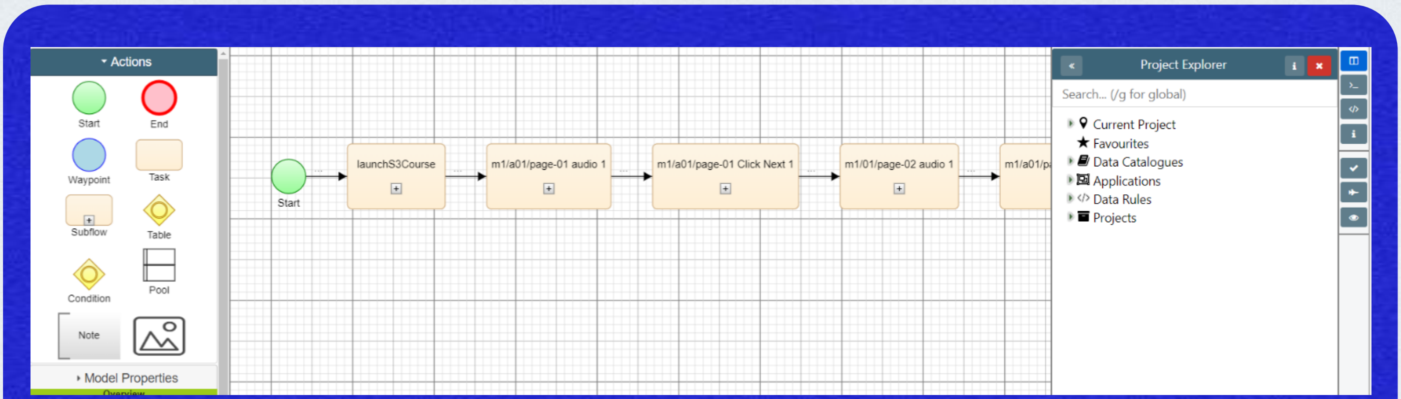
The QE team's first job was to create reusable models for the EVERFI course's library of reusable web components. They dragged-and-dropped task and decision blocks to Quality Modeller's flowcharts, linking them with arrows. The flowcharts then act as cause-and-effect models, mapping different routes through the logic of each course component:


4:45
16:9
Header
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Amet volutpat vitae vitae, varius vulputate. At arcu pretium posuere sed feugiat curabitur nec quis. Massa at libero pretium tristique eleifend quam habitant dolor suspendisse.
Back
Next
- Knowing When to Intervene (module-1/module-1-activity/page29) Multi-Page Carousel - v(1)
No and No Back
No Next
Navigate Backward - v(1.01)
End
Start
Click Next
End -> Start
– Inte
End
v(1)
Start
Start
Start
Navigate Forward - v(1.01)
End
No
Click Next
Navigate Backward?

A "Carousel" component (left) modelled as a flowchart in Quality Modeller. The user can use the "next" and "back" buttons to navigate forwards and backwards through the slides. Clicking "next" on the last slide will end the user's interaction with the carousel, as shown in the red "End" block in the flowchart. Alternatively, the user's journey might end with not clicking "next".

Assembling end-to-end flows

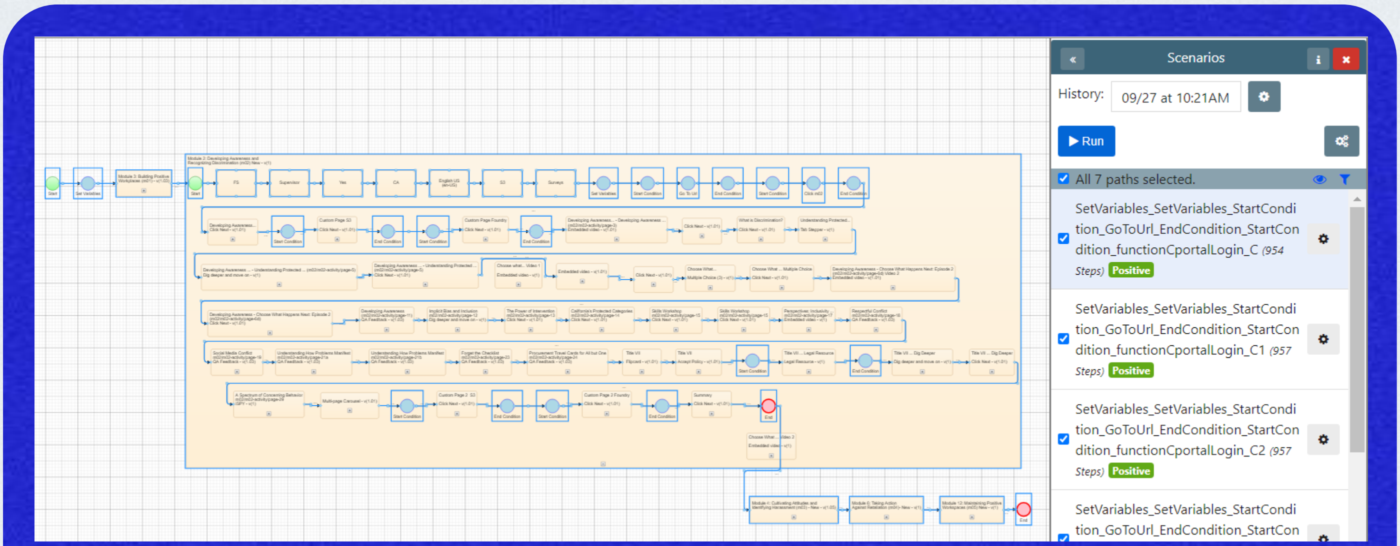
Each flowchart in Quality Modeller further becomes a reusable component, which can be dragged, dropped, and assembled to form end-to-end flows. Master models pass parameters and data attributes from one subflow to the next, making it quick and easy to test different course components in combination:



Each model in Quality Modeller becomes a reusable asset, which can be dragged-and-dropped to end-to-end flows.

Automated test generation

Once EVERFI have an end-to-end flow, the logical precision of Quality Modeller's flowcharts enables automated and optimised test generation. The models acts as a directed graph, to which mathematical algorithms can be applied. These identify logically distinct paths through the flowchart, which are equivalent to test cases:



09/27 at 10:21AM

☑ All 7 paths selected.

☑ SetVariables_SetVariables_StartCondition_GoToUrl_EndCondition_StartCondition_functionCportalLogin_C *(954 Steps)* Positive

☑ SetVariables_SetVariables_StartCondition_GoToUrl_EndCondition_StartCondition_functionCportalLogin_C1 *(957 Steps)* Positive

☑ SetVariables_SetVariables_StartCondition_GoToUrl_EndCondition_StartCondition_functionCportalLogin_C2 *(957 Steps)* Positive

☑ SetVariables_SetVariables_StartCondition_GoToUrl_EndCondition_StartCon

Mathematical algorithms automatically identify logically distinct paths through combined subflows.

Overcoming vast logical complexity

The Curiosity platform further provides optimisation algorithms to reduce test volume and duplication. This might, for example, identify a set of logically distinct tests that cover every block or arrow in the flowchart, reducing the number of tests without compromising functional test coverage.

Automated test generation in turn enables EVERFI to test rigorously from among millions of possible course permutations. Meanwhile, the 1:1 relationship between web component and model avoids repetition during test creation, as each test step is defined as a one-off task in a reusable model. Test maintenance is further accelerated, as QE can update reusable flowcharts to update all master models and test artifacts generated from them.

Framework-agnostic script generation

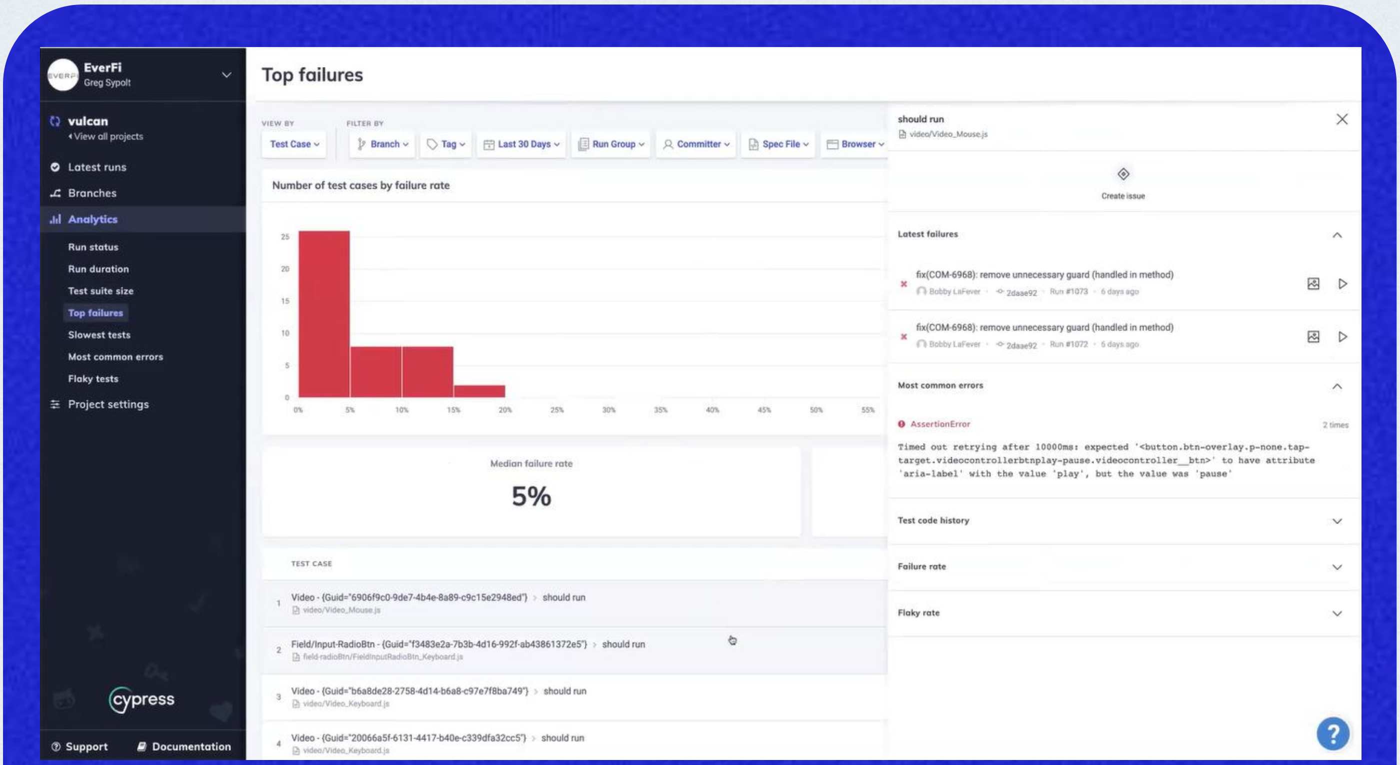
At the same time as optimised test cases, Curiosity's platform generates automated test scripts.

The framework- and language-agnostic script creation uses configurable templates. Automation objects and actions are defined once, before being dragged and dropped onto automation "waypoints" in flowcharts. The modelled logic then passes parameters into each function during test generation, providing a scalable approach to framework development.

At EVERFI, Quality Modeller interacts with EVERFI's command line tool, Modello, and with Javascript code templates in a GitHub repository. Test generation compiles scripts complete with Setup/Teardown, rapidly generating tests for an optimal set of course permutations.

Functional test execution in Cypress

EVERFI selected Cypress for their functional test execution, populating Cypress "it blocks" during test generation. They chose Cypress as it offered a range of benefits for the project:



Median failure rate
5%

TEST CASE

1 Video - {Guid="6906f9c0-9de7-4b4e-8a89-c9c15e2948ed"} › should run
video/Video_Mouse.js

2 Field/Input-RadioBtn - {Guid="f3483e2a-7b3b-4d16-992f-ab43861372e5"} › should run
field-radioBtn/FieldInputRadioBtn_Keyboard.js

3 Video - {Guid="b6a8de28-2758-4d14-b6a8-c97e7f8ba749"} › should run
video/Video_Keyboard.js

4 Video - {Guid="20066a5f-6131-4417-b40e-c339dfa32cc5"} › should run
video/Video_Keyboard.js

should run
video/Video_Mouse.js

fix(COM-6968): remove unnecessary guard (handled in method)
Bobby LaFever · 2daae92 · Run #1073 · 6 days ago

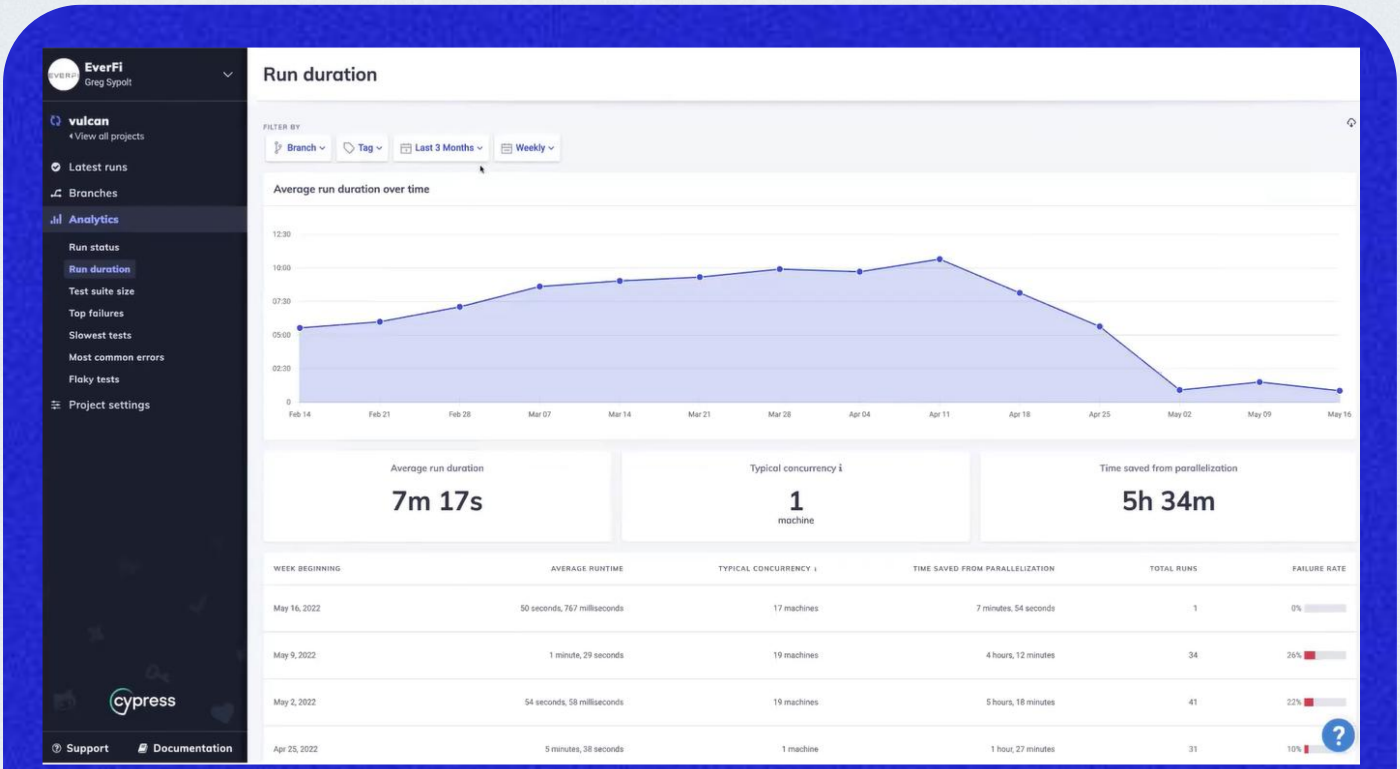
fix(COM-6968): remove unnecessary guard (handled in method)
Bobby LaFever · 2daae92 · Run #1072 · 6 days ago

AssertionError 2 times

```
Timed out retrying after 10000ms: expected '<button.btn-overlay.p-none.tap-
target.videocontrollerbtnplay-pause.videocontroller__btn>' to have attribute
'aria-label' with the value 'play', but the value was 'pause'
```

Cypress' dashboards drill down into the cause of failures for individual tests, in this instance an AssertionError caused by a timeout.

1. **Cypress' reporting and dashboards** enable easy optimization and troubleshooting. Root cause analysis of failing or flaky tests helps the QE team maintain a healthy framework, analysing individual test failures, median failure rates, common causes of failure, screenshots, videos, and more. Cypress further reports on test duration at different levels of concurrency, as well as on the slowest tests and how tests run in parallel across nodes on EVERFI's test grid. This lets QE fine-tune their Cypress testing for performance and scalability.
2. **Cypress' parallelization and performance** further supports the scalability of EVERFI's course automation. Today, for example, setting test concurrency to 17 machines has reduced run times from 7 minutes 17 seconds to just 50 seconds.



Run duration

Average run duration: 7m 17s

Typical concurrency: 1 machine

Time saved from parallelization: 5h 34m

WEEK BEGINNING	AVERAGE RUNTIME	TYPICAL CONCURRENCY	TIME SAVED FROM PARALLELIZATION	TOTAL RUNS	FAILURE RATE
May 16, 2022	50 seconds, 767 milliseconds	17 machines	7 minutes, 54 seconds	1	0%
May 9, 2022	1 minute, 29 seconds	19 machines	4 hours, 12 minutes	34	26%
May 2, 2022	54 seconds, 58 milliseconds	19 machines	5 hours, 18 minutes	41	22%
Apr 25, 2022	5 minutes, 38 seconds	1 machine	1 hour, 27 minutes	31	10%

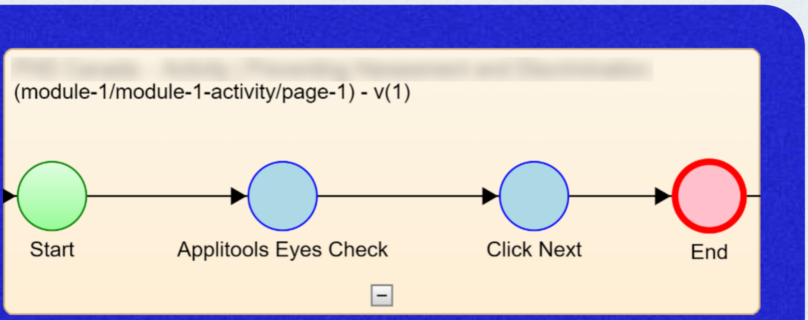
Setting test concurrency to 17 machines has reduced run times from 7 minutes 17 seconds to just 50 seconds.

3. **Cypress' ability to intercept API calls** allows EVERFI to perform API testing at the same time as UI testing. This uses cy.intercept() to check whether a known state of a page is finished loading, sometimes mixing in verification of responses. For example, as testing moves between pages in a course, Cypress tests whether the progress record has been updated.
4. **Cypress' clear and comprehensive documentation** further enabled the quick upskilling of EVERFI's QE team, who were at the time mostly manual testers.
5. **Cypress came equipped with integrations** into technologies used at EVERFI, including GitHub, Slack, Jira, CircleCI and Quality Modeller.

Visual and accessible testing built in

In addition to UI tests with API listeners, Curiosity's platform generates visual test assertions to validate how users see and experience EVERFI's courses. This is as simple as adding a blue automation waypoint into a reusable flowchart. A step in the generated Cypress tests then makes an API call to Applitools Eyes, which can be switched on and off by passing an option into EVERFI's CI pipeline.

During a proof of concept, EVERFI similarly used Quality Modeller to generate a Cypress step that triggers accessibility testing of course components. This used Deque's Axe-Core, achieving test generation that fulfils every facets of EVERFI's test coverage. goals.

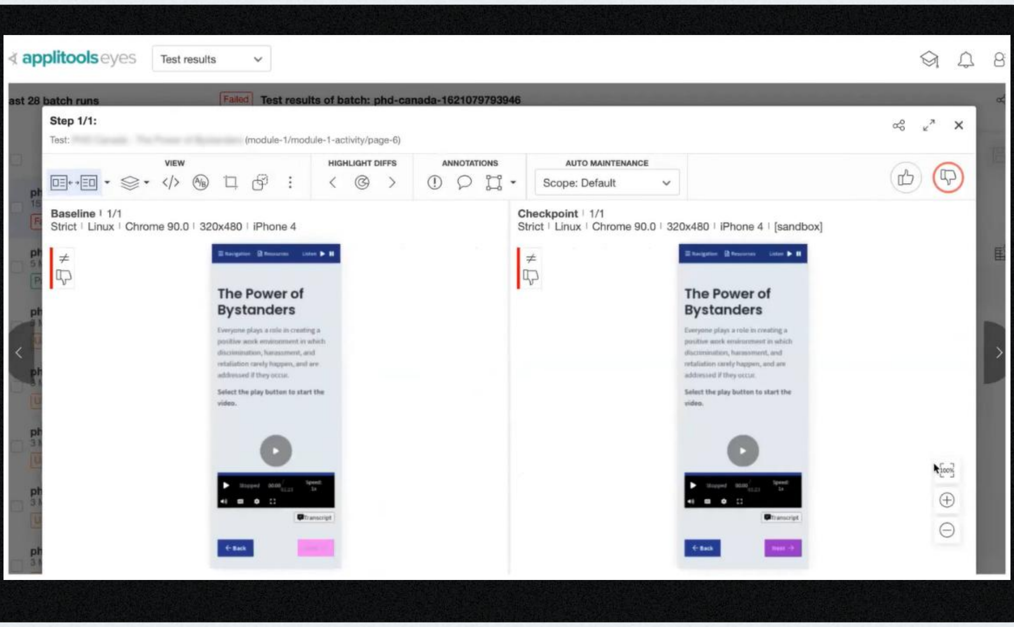


A blue automation waypoint in a model generates an "Applitools Eyes Check" into Cypress tests.

Visual testing

Applitools takes screenshots of the EVERFI courses as Cypress tests run, comparing the images to a baseline for each course. Product Owners and UX designers at EVERFI then work directly in Applitools, quickly approving expected changes to maintain a source of truth for how the system should look. If Applitools flags an unexpected change, a bug report is created.

A visual comparison of the system has identified an unexpected change: The "Next" button should be disabled, but a bug has caused it to be enabled during testing.



ast 28 batch runs

Failed Test results of batch: phd-canada-1621079793946

Step 1/1:

Test: [redacted] (module-1/module-1-activity/page-6)

Scope: Default

Baseline | 1/1
Strict | Linux | Chrome 90.0 | 320x480 | iPhone 4

Checkpoint | 1/1
Strict | Linux | Chrome 90.0 | 320x480 | iPhone 4 | [sandbox]

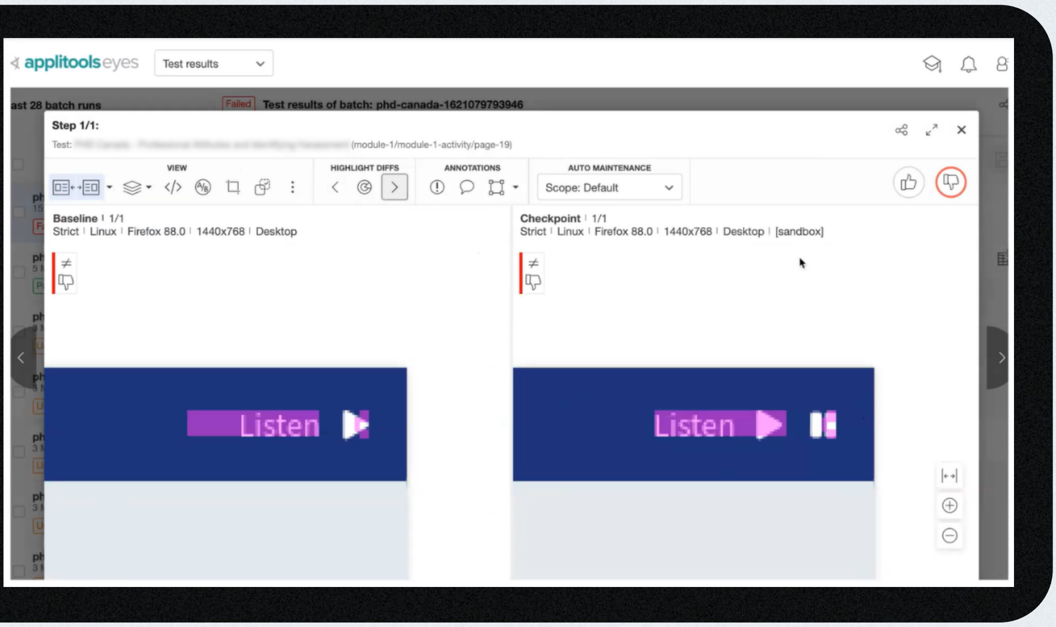
The Power of Bystanders

Everyone plays a role in creating a positive work environment in which discrimination, harassment, and retaliation rarely happen, and are addressed if they occur.

Select the play button to start the video.

Applitools' visual testing enables EVERFI to identify bugs that are otherwise undetectable during functional test automation. Meanwhile, involving product owners and UX designers supports QE's goal of boosting cross-team collaboration and upfront thinking about quality.

A visual comparison of the system has identified an unexpected change: The "Next" button should be enabled, but a bug has caused it to be enabled during testing.



Failed Test results of batch: phd-canada-1621079793946

Step 1/1:

Test: [redacted] (module-1/module-1-activity/page-19)

AUTO MAINTENANCE: Scope: Default

Baseline | 1/1
Strict | Linux | Firefox 88.0 | 1440x768 | Desktop

Checkpoint | 1/1
Strict | Linux | Firefox 88.0 | 1440x768 | Desktop | [sandbox]

Testing across devices and resolutions

Applitools further fulfils EVERFI's goal of testing across devices, browsers and resolutions. Its Ultrafast Grid takes a visual test completed once locally, rendering this across an extensive set of browsers and devices. The cloud-based execution is resource-efficient and scalable, with high levels of parallelisation and concurrency, alongside image caching and DOM Snapshotting (not screenshots).

Test execution in minutes

As EVERFI generate UI, API and visual tests, CircleCI builds the testing grid, runs the tests, and pushes visual tests to Applitools. Concurrency across nodes achieves optimal performance, with test suites today running on up to 90 machines in as little as 50 seconds. The average test run takes just 15 minutes.

The 6-month mark: Rigorous testing within hours of a check-in

In just 3 months following project kick-off, the Digital Transformation project had reduced test creation time from 1+ months to weeks, including the time to model course components.

The project had further implemented the technologies needed to fulfil EVERFI's multifaceted coverage goals, and to communicate this test coverage to different stakeholders. QE had introduced Quality Modeller, part of Curiosity's Enterprise Test Data platform, to document and generate tests at speed for their Cypress framework, with API, accessibility and visual tests woven seamlessly into the optimised functional tests. By month 6, EVERFI had reduced test creation time for a new course permutation to just 4 hours, and had completely modelled 8 different courses. A range of developments enabled this acceleration:

The QE team had modelled 28 course components and were authoring new models in around 60 minutes.

Around half of the QA Engineers at EVERFI were using this growing library of reusable flows to assemble and test new course permutations at speed.

New best practices for test data allocation further created static data into an S3 bucket, producing a known state of the system for testing.

The QE team had further built a test generation command line tool. This passes options via Quality Modeller's API to trigger test generation. A single command could then generate tests in ~19 seconds, ran locally or via CircleCI, and with visual testing across devices and resolutions.

Yet, QE wanted to move even faster. They sought to automate away the 4 hours spent assembling reusable flows in end-to-end models.

Month 9: Test creation in minutes following a course check-in

QE built a CLI tool to automate model creation, using the newly-added ESM modules in node.js to create “Modello”. Modello leverages changing course schemas to assemble end-to-end flows in Quality Modeller, generating and running tests as courses are checked-in.

When an engineer creates a course, EVERFI’s in-house LMS tool compiles a JSON schema detailing the course components and blocks. Modello parses the JSON, converting it into a usable data structure that maps the course components. It then communicates via REST API to create a model for the course. Modello iterates through each page of the course, searching for reusable flows that match the course components. If it cannot find a matching flow, it adds a subflow to click to the next page in the course. Once the course flow is ready, Modello triggers the generation of Cypress and Applitools tests, which are executed via CircleCI’s test grid, with a standard parallelism of 12.

Shift left testing as courses are checked-in

The process for generating and executing tests in Modello is incredibly simple, enabling rapid “shift left” testing by product owners and developers at EVERFI.

First, an NPM command installs Modello. Next, users create a blank model with a unique tag in Quality Modeller. An init command then parses the course configuration file, and the “model” command assembles a flow in Curiosity’s platform. This can cover the whole course, or can be limited in scope, for instance to test particular modules. Modello’s “generate” command then creates the tests, before the “run” command opens a Cypress sandbox and runs them. This last command is a wrapper for the Cypress “Open” and “Run” commands, with custom logic to resolve dependencies.

Using just 4 commands, testers, developers and product owners can run “shift left” tests just minutes after creating a new course permutation.

Further evolution by month 13

Introducing Modello reduced the time required to create tests for new course permutations to minutes.

The Quality Engineering team have since introduced targeted test generation, further optimising and accelerating course test automation. The targeted test generation is based on GitHub check-ins, focusing testing on impacted pages or modules, instead of re-running end-to-end tests.

The Quality Engineering team is furthermore working to integrate Modello directly into EVERFI's in-house LMS tool. This will generate "shift left" tests within minutes of a course check-in, without needing to run Modello commands manually.

The results so far

The Digital Transformation project has today fulfilled EVERFI's goals of achieving optimal test coverage at speed, with collaboration between stakeholders and upfront thinking about quality. In just 9 months, the project achieved astonishing results for 52 courses:

Test creation and execution time has been reduced from weeks to minutes.

85% of all course testing is now automated, saving the resources previously spent on outsourced crowd and exploratory testing.

Testing hits 100% permutation coverage, with visual test assertions for every page.

Tests run in minutes, with concurrency of up to 90 machines.

Quality is now everyone's responsibility, as course creators trigger "shift left" tests and designers review visual tests.

Test maintenance is as quick as automatically rebuilding tests.

On the roadmap

Yet, EVERFI do not want to stop there. In future, the QE team will:

- Move from 85% to 100% automation for course testing. This will generate tests for scenarios today tested manually, including testing audio, video, and translations.
- Integrate Modello into EVERFI's in-house LMS tool, generating shift left tests without needing to run commands manually.
- Use code coverage to further target testing, adding instrumentation to code to see which lines tests are hitting.
- Use Curiosity platform's OpenAPI importers to model APIs, leveraging framework-agnostic test generation to create tests for a best-of-breed API testing framework.
- Automate test data generation, including database scanning to understand its structure, generating courses in a known state, and adding database snapshotting.
- Cache generated tests to accelerate the build process, re-running existing tests when they are still up-to-date.
- Model requirements to generate shift left tests for new user stories.
- Add performance testing using Lighthouse.

Get started today!

Visit www.curiositysoftware.ie to learn more or book a demo with a Curiosity expert today!

Additionally, you can also email us at info@curiosity.software

Call USA:

+1 914 218 0180

Curiosity Software Ireland

Unit 6 The Mill, The Maltings, Bray, Co. Wicklow, A98 XV40,
Ireland

Curiosity Software USA

4136 Del Ray Ave. Suite 658, Marina Del Rey, CA 90292, USA