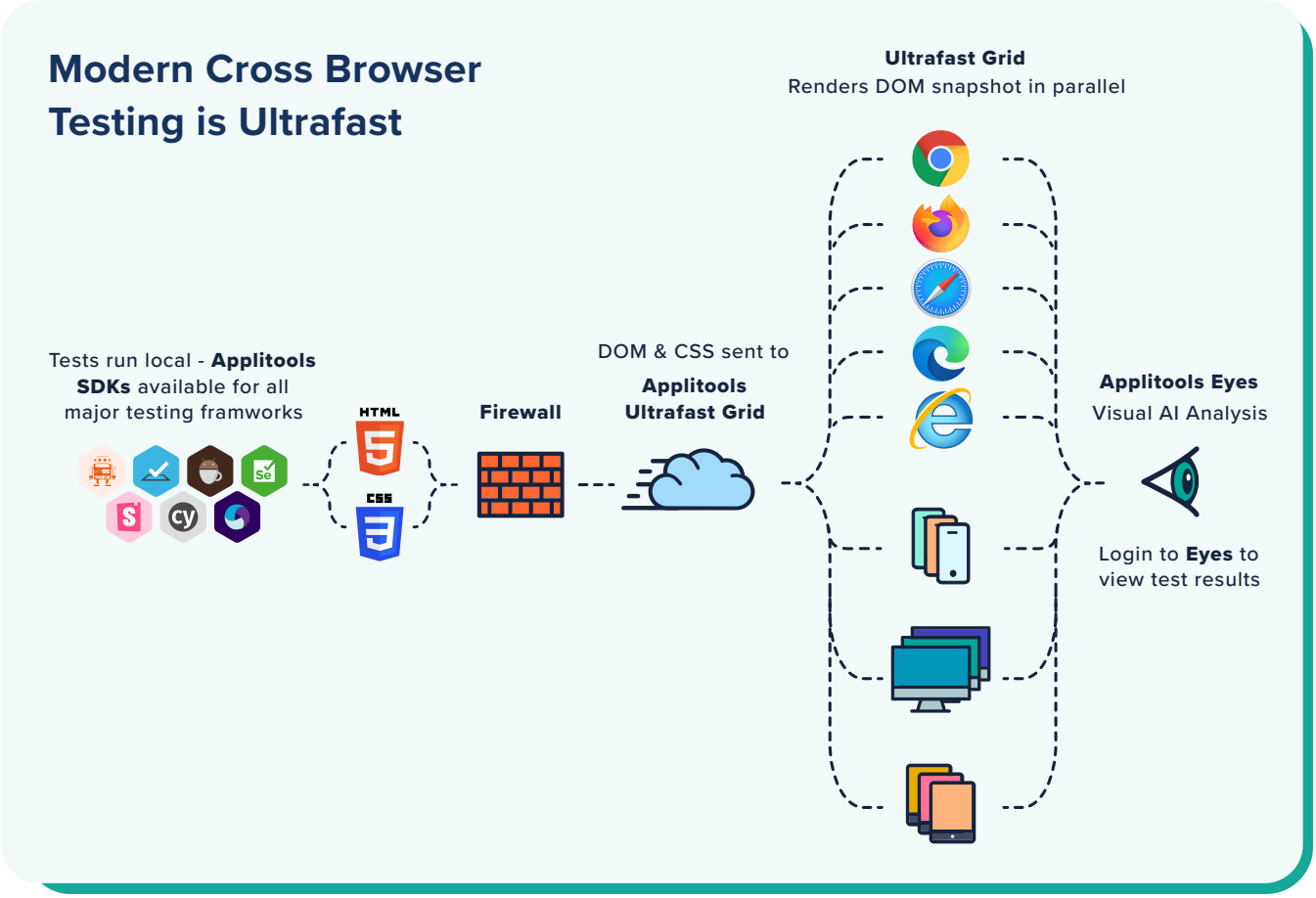
Modern Cross Browser Testing Through Visual AI

3,112 Hours of Empirical Evidence From 203 Selenium, Cypress, & Webdriver.IO Engineers

applitools

Why Read This Report?

In just 10 minutes, you will learn how your team can release higher quality apps faster and for less money by modernizing your cross browser, cross device, and cross operating system testing. Sourced from 203 Selenium, Cypress, and Webdriver.IO engineers who spent a combined 3,112 hours — **that's 1.5 years of engineering effort** — writing, running, analyzing, reporting, and maintaining 21 cross environment tests against a real world application, we provide empirical evidence of how modern cross browser testing powered by the Applitools Ultrafast Test Cloud is vastly superior to traditional, old-school techniques that are ill equipped to handle modern responsive apps in today's digitally transformed, remote work world.



18x
Faster to Complete a Full Test Cycle

81x
More Code Efficient

77%
Increase In Engineer Satisfaction

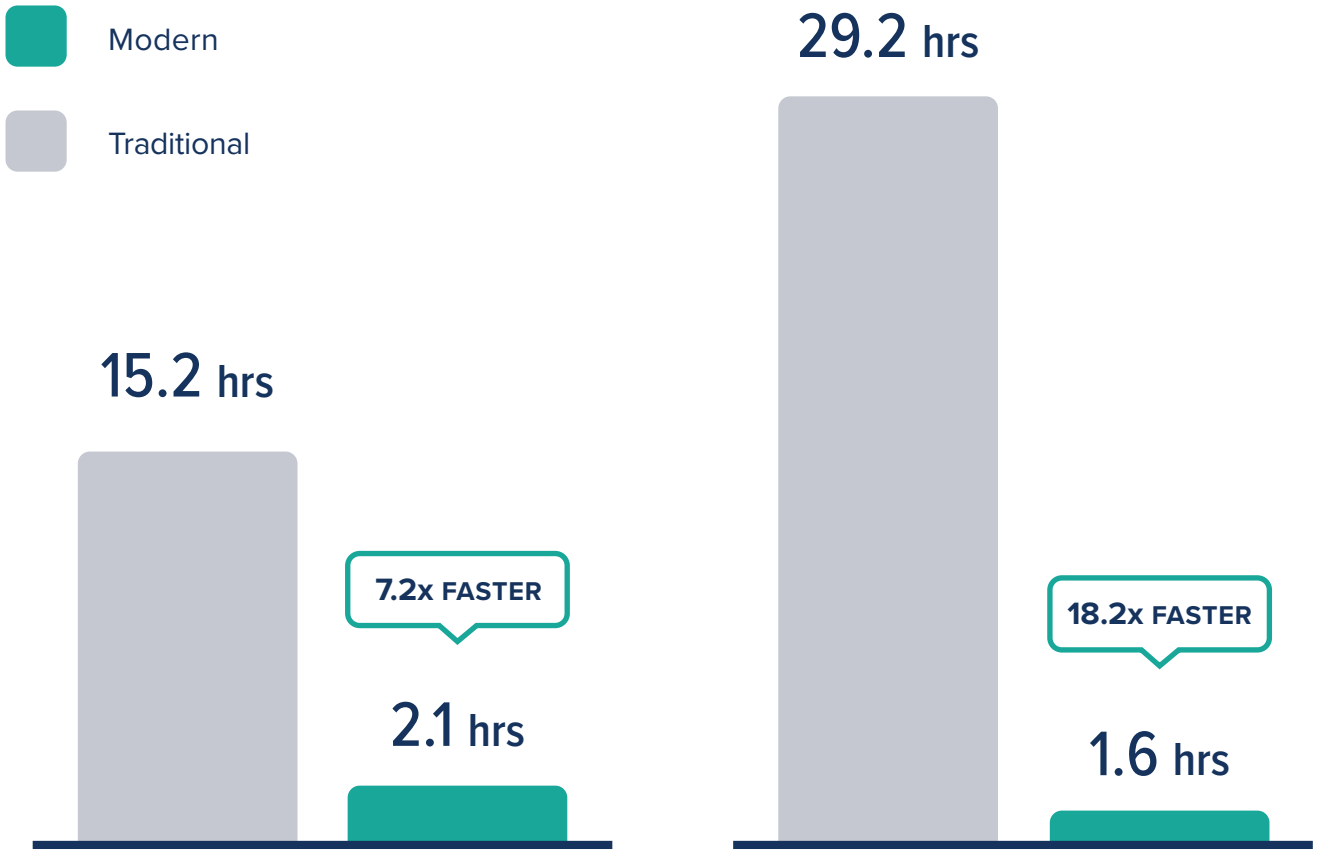
What is Modern Cross Browser Testing?

Today's JavaScript, HTML, and CSS standards are strictly defined by W3C and WHATWG standards bringing deep consistency across modern browsers. Additionally, modern JavaScript solutions like jQuery & Babel have smoothed out browser support for JavaScript APIs, making the testing of browser engines themselves relatively binary. In a nutshell, the modern browser engines have been tested to near perfection opening the door for a better approach to cross browser testing.

USING TRADITIONAL CROSS BROWSER TESTING APPROACHES...	USING MODERN CROSS BROWSER TESTING APPROACHES THROUGH APPLITOOLS...
We Test Every Browser Combination. Prior to modern standards, applications behaved differently across every browser, so we tested every combination. The rise of mobile and the responsive web multiplied the number of combinations we had to test, making this old-school approach expensive, slow, and non-scalable.	**One Functional Test Run Achieves 99% Coverage.** With standards bringing consistency across modern browsers we can now identify 99% of issues from a single functional test run on a single browser.
Visual Testing Is Neglected. Ten years ago we focused almost exclusively on functional bugs on a few browsers. Visual testing was either done manually, or not done at all. In an agile and CI/CD world, teams operating this way today see a lot of late stage bugs and waste time on false positives, because cross environment bugs are almost always visual and pixel based diff tools are mostly ineffective.	**Cross Browser Bugs Today are Mostly Visual.** Today we have mobile and IoT device proliferation, complex responsive design viewport requirements, and dynamic content. Since rendering the UI is subjective, the majority of cross-browser defects are visual in nature requiring the 99.9999% accuracy of Visual AI to automate testing at scale.
Testing Is Done Sequentially on Real Devices Most teams still rely on manual testing or budget permitting, connect to device farms in the cloud. Test runs are queued up and run on a first-come, first-serve basis. This approach is slow, unstable, and expensive. Teams testing this way today are often an expensive QA bottleneck blocking faster, higher quality releases.	**Fast, Parallel Rendering Across All Browsers and Viewport Combinations.** Tests run faster using a single local functional run that captures the DOM & CSS rules for every browser state rendering it in parallel across all browsers and viewports. Screenshots are then instantly analyzed by Visual AI to find functional and visual bugs.
Traditional Methods Introduce Security Risk. Cloud testing requires reverse proxy or an IPSEC VPN. Both of these techniques increase our penetration risk by opening the door to outside factors. Mitigating this risk is impossible and getting approval from IT team is a slow, painful and expensive.	**Security Concerns Are Eliminated.** There are no inbound connections needed with a modern cross browser testing approach. This means there are no additional vulnerabilities from outside your network.

18.2x Faster to Complete a Full Test Cycle

applitools



What It Means to the Engineer?

Engineers now have Ultrafast Grid and Visual AI technology capable of comprehensively testing large, modern apps across dozens of environments in minutes, not days.

What It Means for the Team?

The QA bottleneck holding back release velocity is eliminated enabling true CI/CD; Application quality simultaneously improves dramatically.

Where Did The Time Go?

applitools

One-Time Install :06 Minutes

applitools

One-Time Install :49 Minutes

BrowserStack CrossBrowserTesting A SMARTBEAR COMPANY SAUCELABS

Marie Drake | Principal Test Engineer | News UK

"Cross browser testing is ultra fast using Applitools Ultrafast Test Cloud. Bugs that I missed (and there were lots!) on different browsers and viewports were easily caught without affecting the time it takes to run my tests." comments Marie Drake, Principal Test Engineer at News UK. "Functional testing frameworks automate browsers, but they are not built to catch the visual changes now incredibly common cross environment. Applitools Ultrafast Test Cloud helps engineers succeed when doing modern cross browser testing."

User Satisfaction Increases 77% vs. Traditional Approaches

applitools

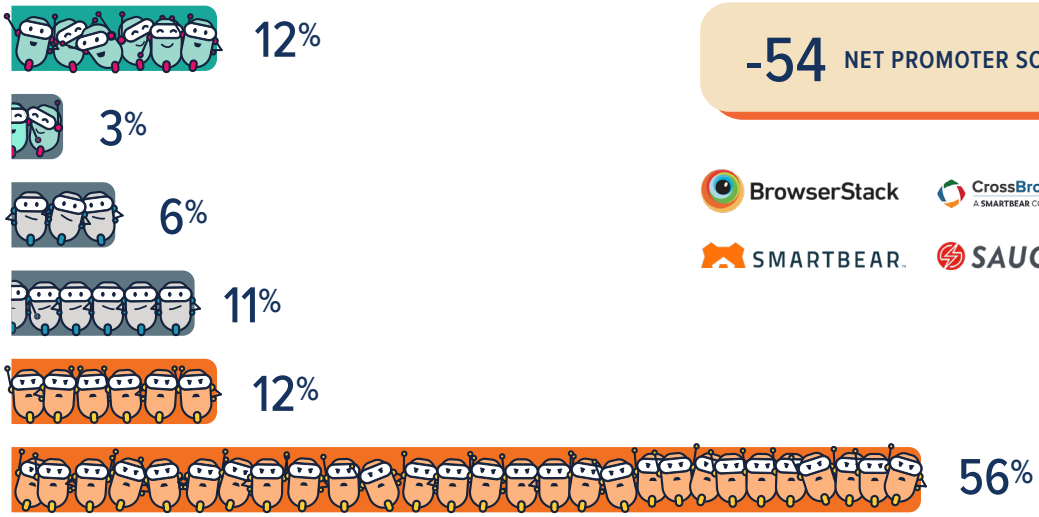
MODERN

10 Promoter	60%
9 Promoter	15%
8 Passive	17%
7 Passive	3%
6 Detractor	2%
5 or less Detractor	3%

applitools

70 NET PROMOTER SCORE

TRADITIONAL



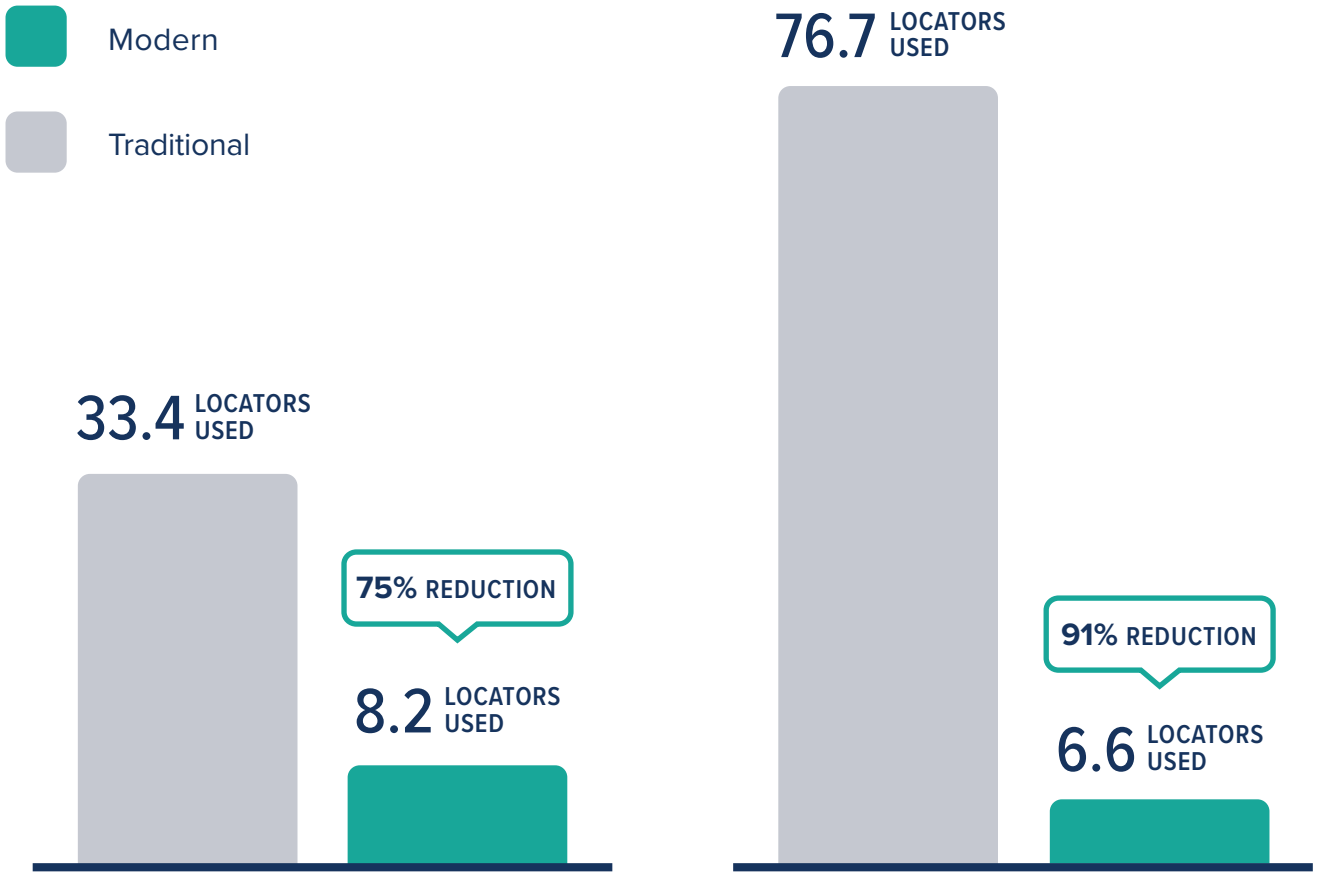
10 Promoter	12%
9 Promoter	3%
8 Passive	6%
7 Passive	11%
6 Detractor	12%
5 or less Detractor	56%

-54 NET PROMOTER SCORE

BrowserStack CrossBrowserTesting A SMARTBEAR COMPANY SMARTBEAR SAUCELABS

*Net Promoter Score ranges from +100 (high satisfaction) to -100 (low satisfaction). NPS is calculated by ignoring the passive responses and subtracting the % of detractors from the % of promoters. Products with an NPS over 50 are considered world-class while products below 0 are in decline and losing users.

91% Reduction In Locator Instability Vs. Traditional Approaches



What It Means to the Engineer?

Reducing reliance on locators means tests break less often and throw far fewer false positives. This allows engineers to expand coverage and spend more time managing quality.

What It Means for the Team?

Teams with stable test code can test more or test faster or some combination of both depending on their needs and goals.

81x More Test Code and Screenshot Efficient

applitools

1,694

LINES OF CODE OR SCREENSHOTS

Traditional

21

SCREENSHOTS

Modern

What It Means to the Engineer?

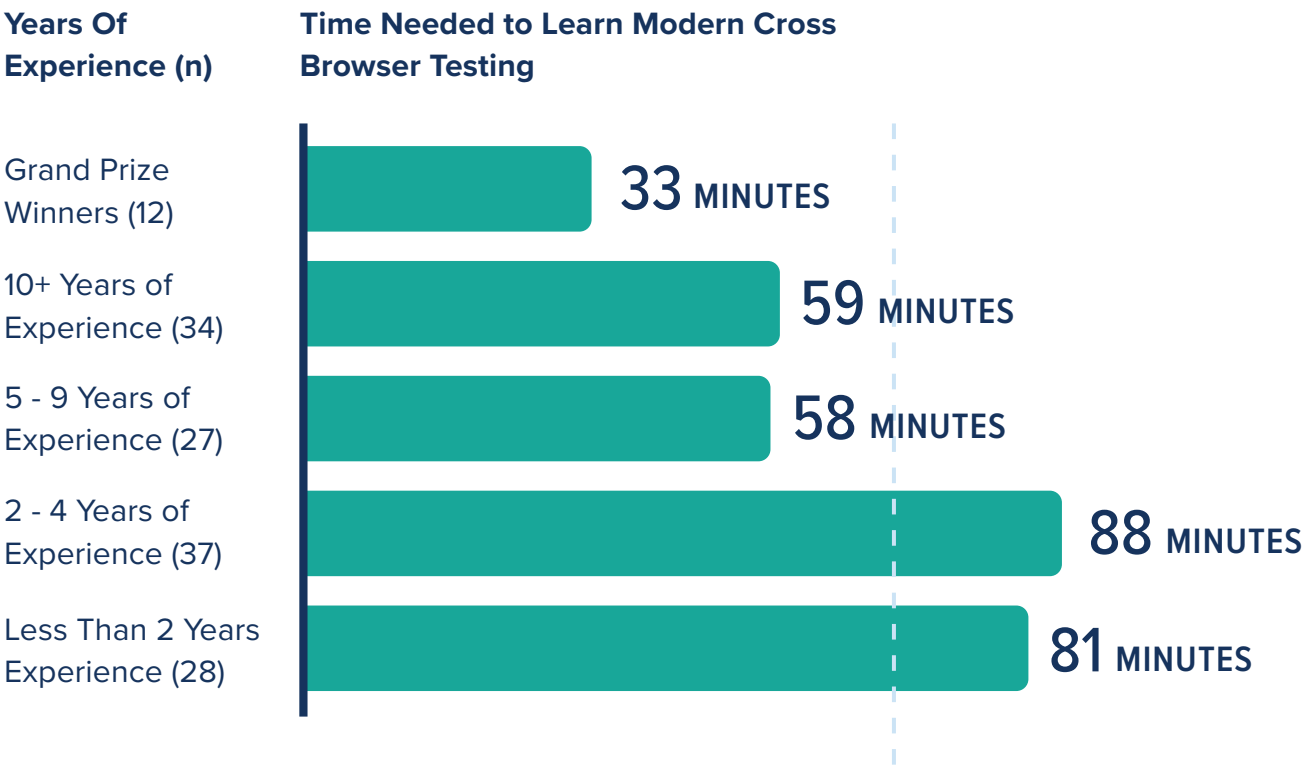
Screenshots analyzed by 99.9999% accurate Visual AI dramatically reduce false positives and allow visual inspection in seconds of any bugs found.

What It Means for the Team?

Testing speeds and coverage increase with almost zero added maintenance. This enables dev teams to improve release velicity and push to true Ci/CD more easily.

Engineers of Any Skill Level Learned It in 88 Minutes or Less

applitools



The average engineer (n=186) had 5 years of experience and learned modern cross browser testing using Applitools Ultrafast Test Cloud in 72 minutes.

What It Means to the Engineer?

It's easy for engineers of any experience level to learn Applitools Ultrafast Test Cloud and upskill to modern cross browser testing.

What It Means for the Team?

Teams operate together better with this modern testing approach and have a team member (or members) they can rely on to help release faster with more confidence.

Gathering the Empirical Data

The Ultrafast Cross Browser Testing Hackathon

To execute the study, we built an application representative of common, modern app cross environment testing use cases. In June 2020 a challenge was issued to testers all over the world to compete, and learn, by managing three real-world cross environment testing scenarios using their preferred traditional cloud testing solutions or running locally using Selenium, Cypress, WebdriverIO, and others. These same quality engineers then repeated the process for the same three test scenarios using Ultrafast Test Cloud from Applitools. Testers competed for 100 prizes worth over $50,000 and were judged on the following criteria:

1. **Test Creation and Execution Time**
   How code-efficient and time-efficient is test creation? How quickly do tests run?

2. **Test Coverage and Analysis**
   How easily and quickly are results analyzed? How effective are tests in catching all the bugs?

3. **Team Analysis and Collaboration**
   How easily and quickly are bugs logged and resolved across the team?

4. **Test Stability and Auto-Maintenance**
   How easy and quick is it to maintain tests? Do all tests execute successfully?

The Challenge. Test a Modern, Responsive App Against Three Real World Scenarios Across Seven Browser and Viewport Combinations.

In the Hackathon, our contestants played the role of test automation engineer for “AppliFashion”, a high profile e-commerce company that sells fancy shoes. The AppliFashion web app is used by millions of people, using various devices and browsers to buy shoes. In our Hackathon scenario, the 1st version of the app (V1) is already built and is “bug-free”, but the dev team is now coming up with a newer version(V2) that may be full of bugs. The challenge is to build the automation suite for the first version of the app and use it to find bugs in the second version (V2) of the app. Our contestants need to automate three (3) main tasks across seven (7) different combinations of browsers and screen resolutions (viewports). Further, they need to automate the tasks using both a traditional approach to cross browser testing and the modern approach using Ultrafast Test Cloud for both V1 and V2 versions of the app.

Empirical Evidence Sourced From 3,112 Hours of Data

2,224 Participants. 203 Qualified Submissions. Over 100 Winners.

We were blown away by the enthusiastic response from the testing community. 2,224 engineers signed up to participate with 203 ultimately submitting results after spending ~17 hours to complete the challenge. That's a total of 3,112 engineer hours. Participants were geographically, firmographically, and technologically dispersed, providing us with the industry's largest, highest quality, and freely available data set for understanding the impact of modern cross browser testing on test automation speed, coverage, and stability.

Tarun Narula
Naukri.com

"It was a wonderful experience which was challenging in multiple aspects and offered a great opportunity to learn modern cross browser testing. It's really astounding to realize the coding time and effort which can be saved for tasks that can be so easily done using Ultrafast Grid and Visual AI. Hands down, Applitools Ultrafast Grid is the tool to go for while making a shift to modern cross browser testing. Cheers to the team that made this event possible.

Oluseun Orebajo
FamsITSolutions

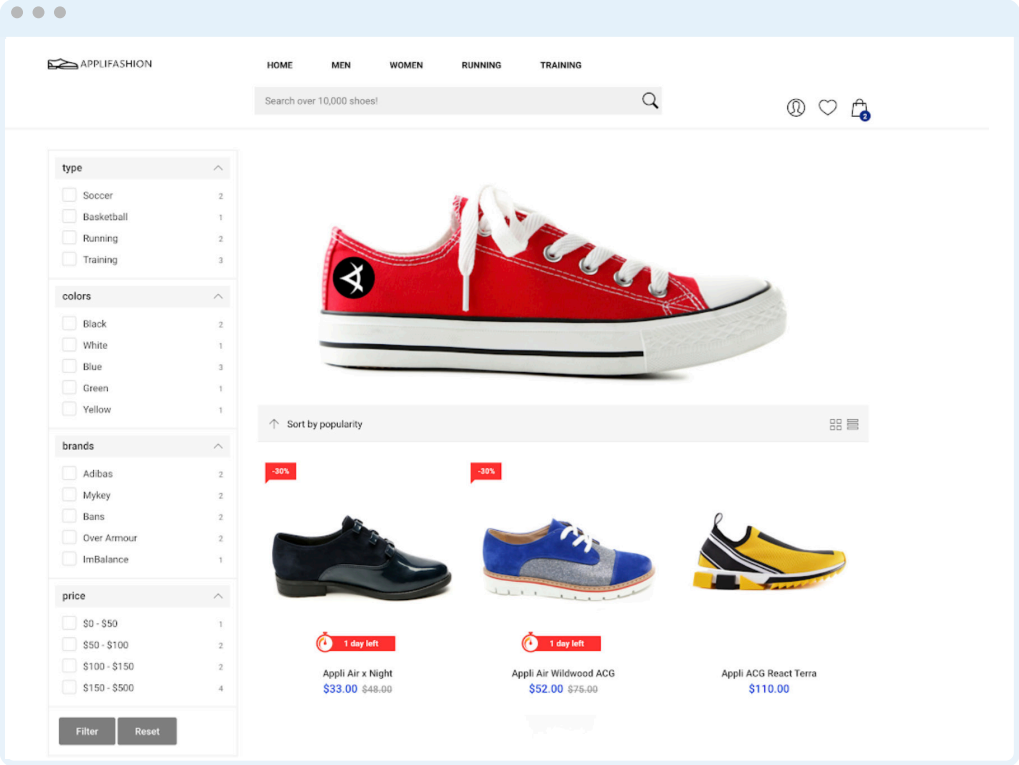
"The effort to implement a comprehensive cross environment testing strategy using traditional approaches are astronomical. Applitools has totally changed the game with the Ultrafast Grid. What took me days of work with other approaches only took minutes with the Ultrafast Grid! Not only was it easier, it's smarter, faster, and provides more coverage than any other solution out there. I'll be recommending the Ultrafast Grid to all of the clients I work with from now on."

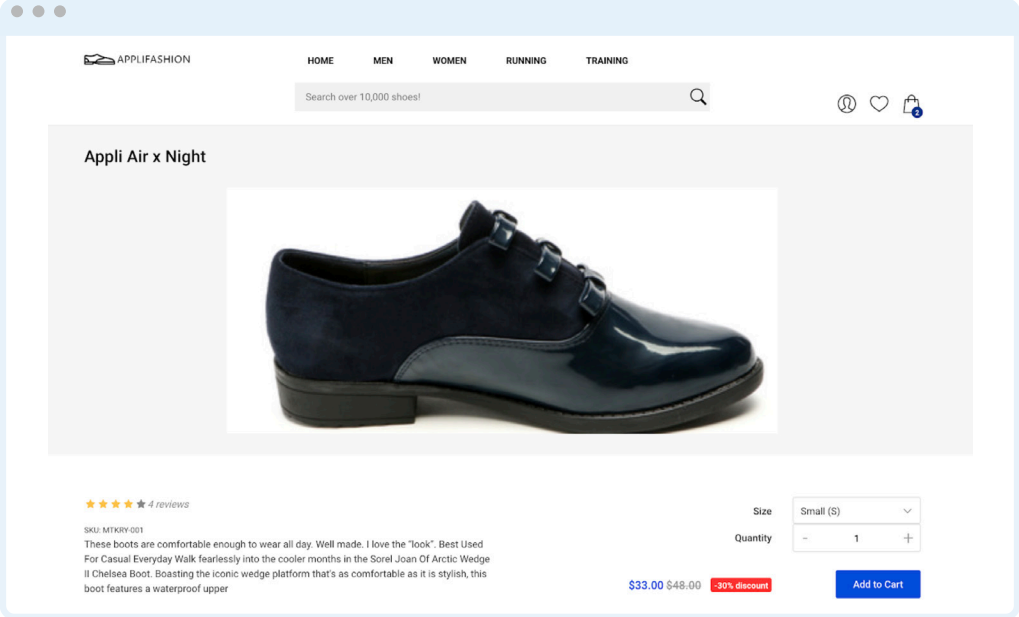
Arjun Blok
Mendix

"The effort to implement a comprehensive cross environment testing strategy using traditional approaches are astronomical. Applitools has totally changed the game with the Ultrafast Grid. What took me days of work with other approaches only took minutes with the Ultrafast Grid! Not only was it easier, it's smarter, faster, and provides more coverage than any other solution out there. I'll be recommending the Ultrafast Grid to all of the clients I work with from now on."

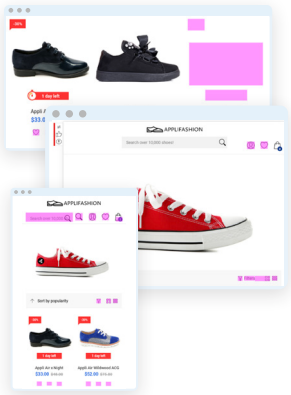
Introducing AppliFashion: Our Hackathon App

Imagine you are a test automation engineer for AppliFashion, a high profile e-commerce company that sells fancy shoes. The AppliFashion web app is used by millions of people, using various devices and browsers to buy shoes. It's a modern, responsive app that includes many of the sorting, filtering, and selection functionality that you'd expect from a well designed ecommerce site today.



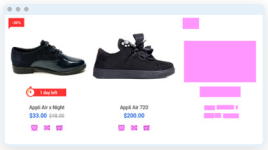


The Challenge: Three Test Scenarios



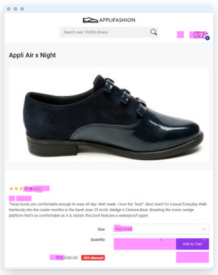
Test Scenario One: Cross-Device Elements Test

The AppliFashion app is responsive, so when testing on various viewports, the elements of your application change. Some are hidden, some are made visible, some are repositioned to accommodate the space available. For example, in the app, the Search field is displayed on laptops and tablets, but is hidden on mobile devices. Your job is to find these changed elements and ensure they are properly hidden or displayed in all viewports and browsers.



Test Scenario Two: Shopping Experience Test

Like any good ecommerce app, there is a lot of important filter functionality vital to conversion. In this scenario we replicate this testing challenge by having applicants filter for “Black” shoes, then ensure the filter results and all functionality is as expected in the application.



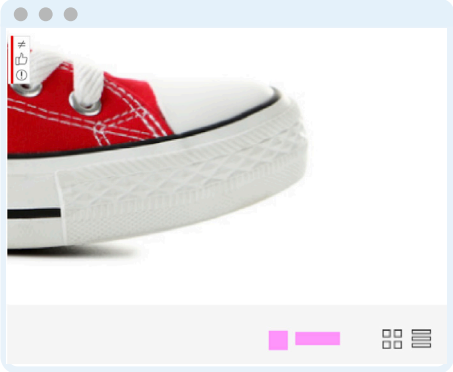
Test Scenario Three: Product Details Test

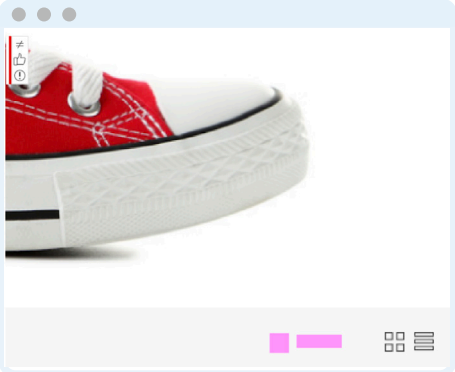
Once the customer has found what they want, it’s time to look more closely at the shoe. This is often the final step before purchase, so you must navigate to the product details page and check to see that everything is functionally and functionally accurate.

Scenario One

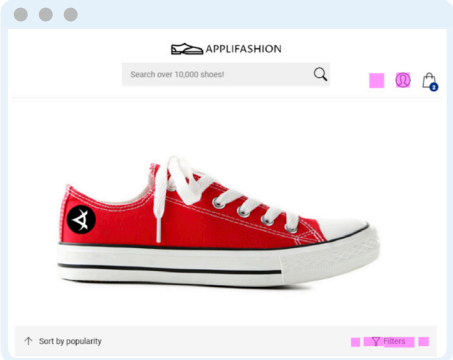
Test For Cross Device Elements

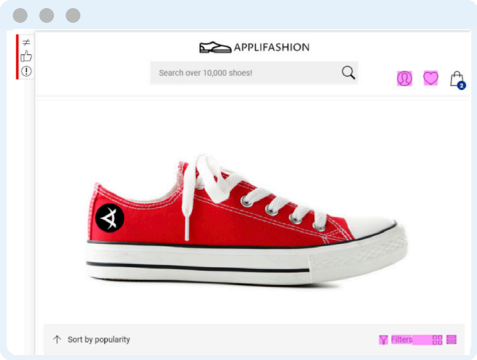
DESKTOP VIEWPORT



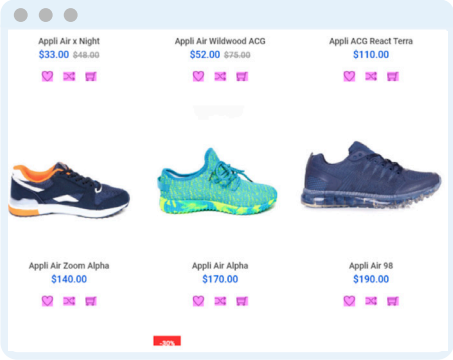


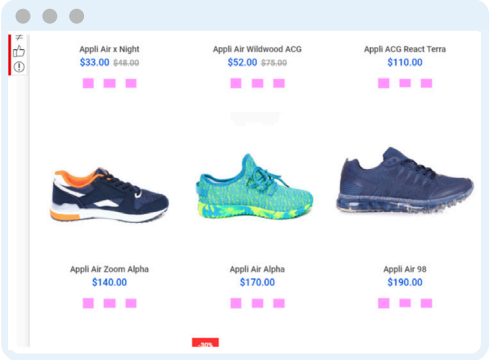
TABLET VIEWPORT ONE





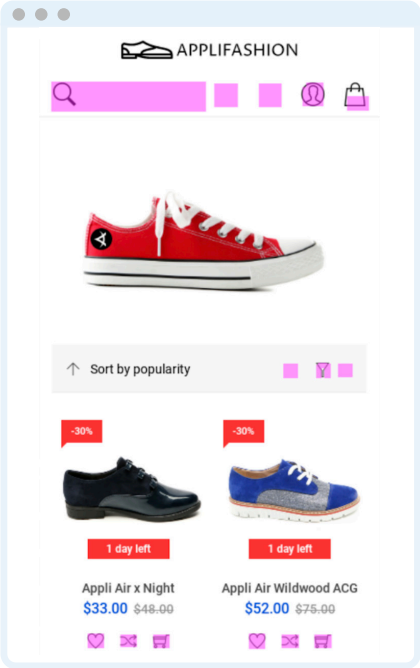
TABLET VIEWPORT TWO

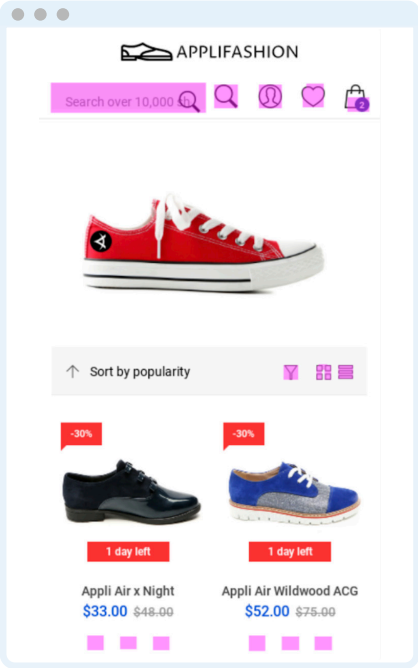




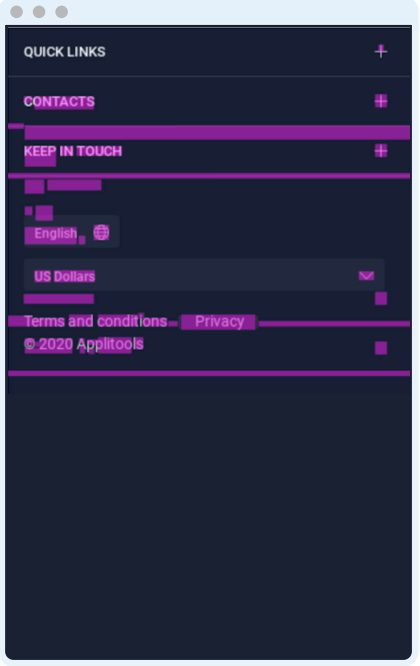
Test Scenario One: Cross-Device Elements

MOBILE VIEWPORT ONE:





MOBILE VIEWPORT TWO:

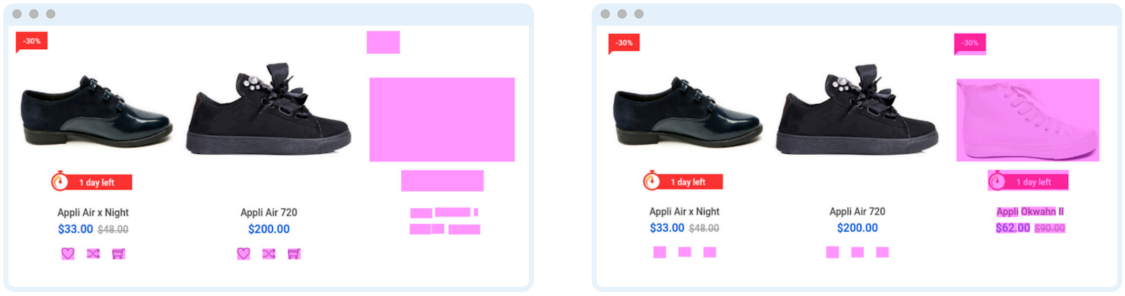


Scenario Two

Test the Shopping Experience For Sorting and Filtering

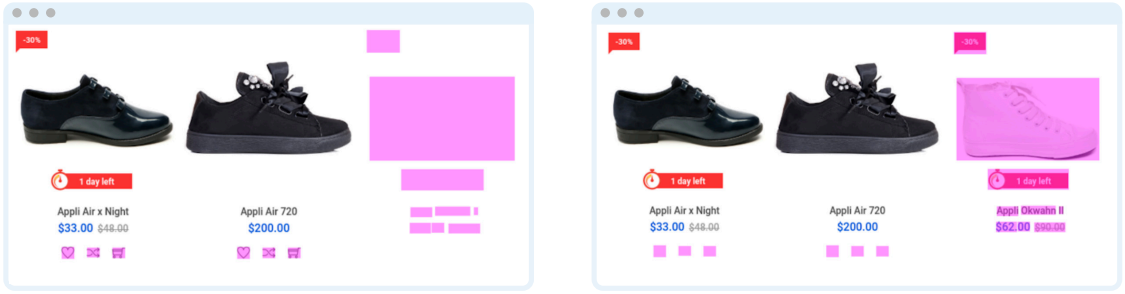
DESKTOP VIEW:

Note this is a functional test where when searched for “Black” shoes instead of 2 shoes, 3 shoes shows up and is highlighted by Applitools Eyes Visual AI



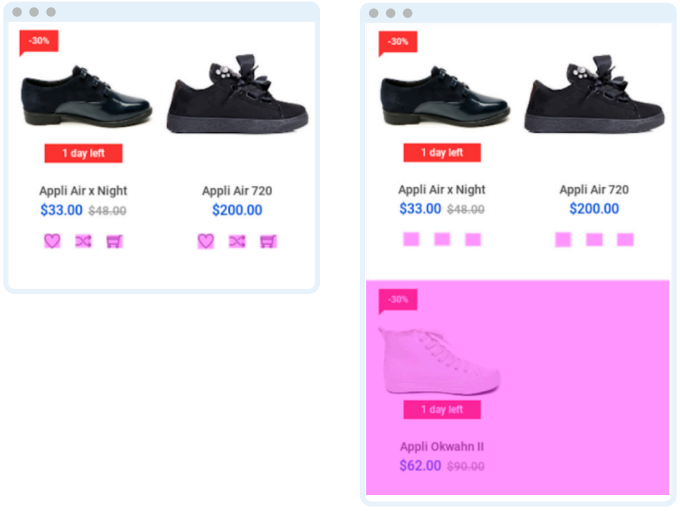
TABLET VIEW:

Note this is a functional test where when searched for “Black” shoes instead of 2 shoes, 3 shoes shows up and is highlighted by Applitools Eyes Visual AI



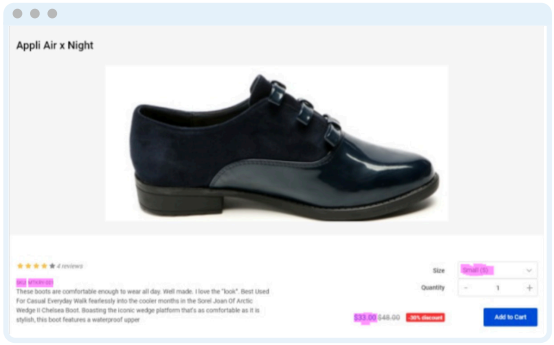
MOBILE VIEW:

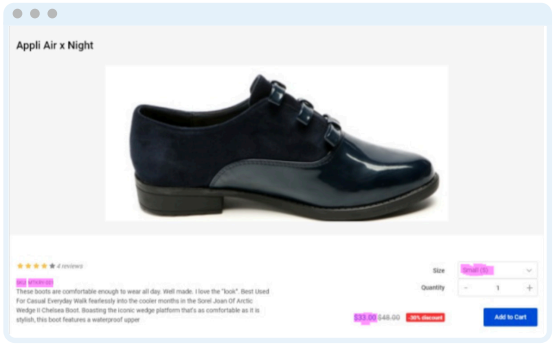
Note this is a functional test where when searched for “Black” shoes instead of 2 shoes, 3 shoes shows up and is highlighted by Applitools Eyes Visual AI



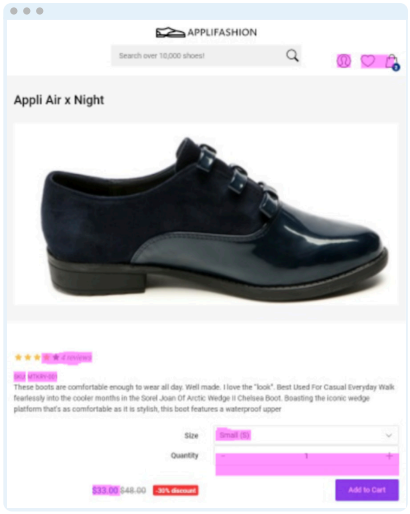
Scenario Three
Product Details Test

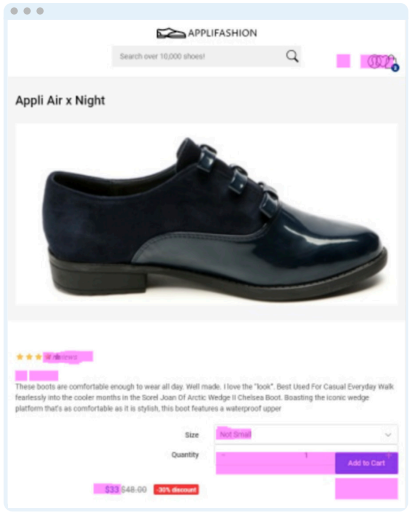
DESKTOP VIEW



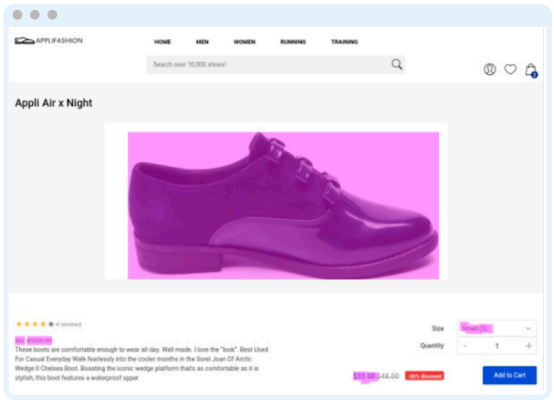


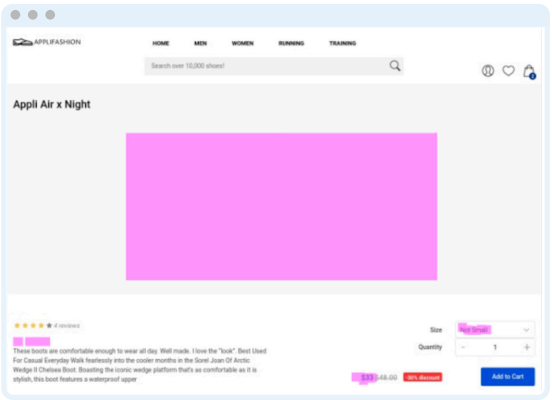
TABLET VIEW





FIREFOX ONLY CROSS BROWSER BUG - ALL DEVICES





Representation From the Global Engineering Community

% OF SUBMISSIONS BY COUNTRY

United States 28%

India 19%

United Kingdom 14%

Brazil 3%

France 3%

Spain 3%

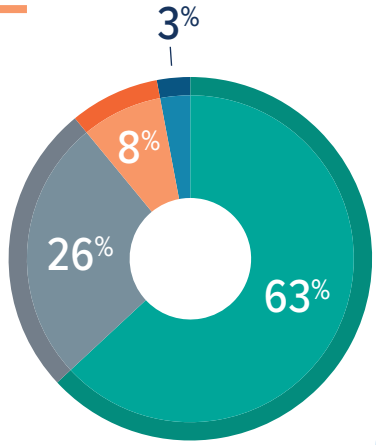
Canada 2%

Kenya 2%

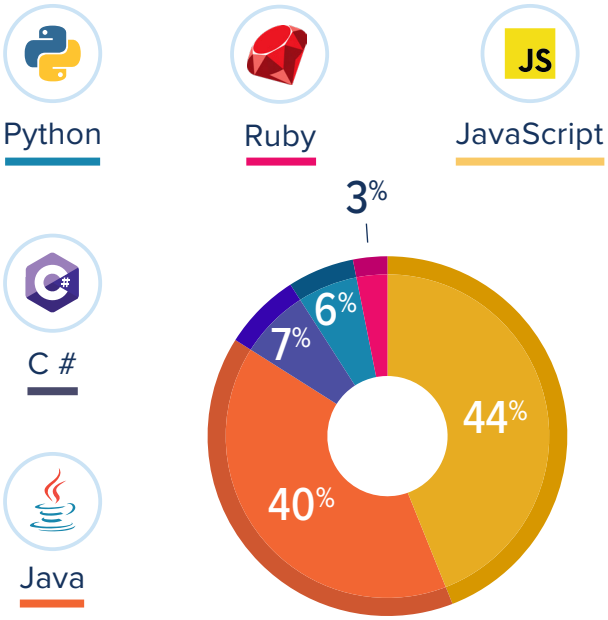
Australia 2%

Plus another 23% from 40 additional countries!

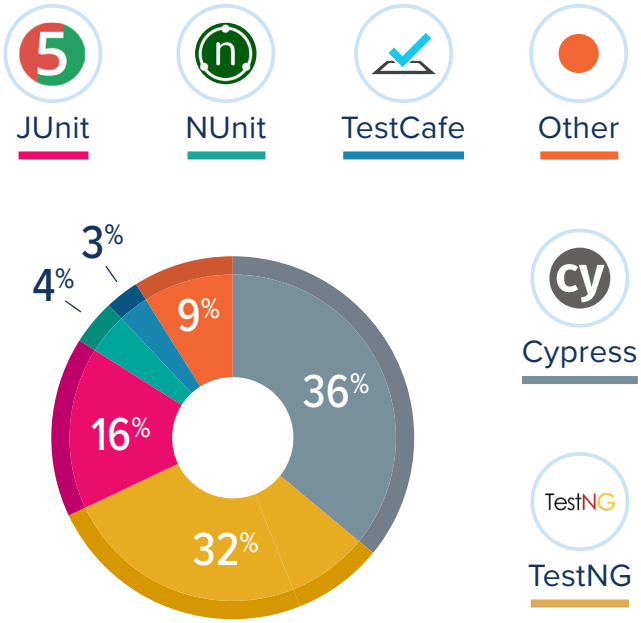
% OF SUBMISSIONS BY TEST RUNNER



% OF SUBMISSIONS BY LANGUAGE



% OF SUBMISSIONS BY TEST FRAMEWORK



Our 100 Ultrafast Cross Browser Testing Hackathon Winners

Grand Prize Winners

Tarun Narula
Naukri.com

Oluseun Orebajo
FamsItSolutions

Arjan Blok
Mendix

Himanshu Soni
Amadeus Software

Henry Andres Correa
Team International

Michael Haselhurst
Sage UK

Predrag Pavlovic
Redbox

Stephen Kilbourn
Dialexa

Ricardo Mediavilla Maldonado
Karsun Solutions

Maros Kutschy
Ness

Kerry McKeever
Paylocity

Thomas Knee
TestifyQA

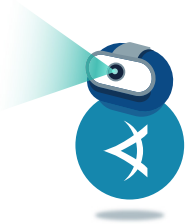
Are You Ready To Try It?

In Just A Few Hours, You Can Deploy Modern Cross Browser Testing

A Virtual, Private Hackathon

Want to test your skills across your global team?

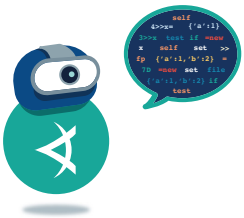
We will package up our Hackathon app, provide instructions, guide your team through the contest, judge results, and celebrate the winners!



Your Private Visual AI Upskill Webinar

Want to learn how to use Visual AI from Angie Jones or Raja Rao?

For qualifying teams, we will secure a private webinar reviewing all the use cases, showing you how it works, and sharing more details about the results of this study.



We'll Run Your First 10 Cross Browser Tests For You

Ready to start now? Let's get going!

For qualifying teams, we will take a portion of your existing code based test suite and add Visual AI tests in just a few hours.

TEST AUTOMATION U

Upskill for Free at Test Automation University

Modern Functional Test Automation Through Visual AI

Selenium Webdriver with Java

Introduction to Cypress

Methodology

Build The Application - First, we built a fully functioning modern, responsive, eCommerce application representing real world cross environment testing use cases. Participating engineers were presented with a bug-free v1 application, then challenged to manage cross environment quality for a new v2 candidate of the application focusing on three use cases. These use cases included testing for 1) cross-device elements 2) dynamic shopping experience and 3) product details presentation. Each of these challenges was presented across 7 browsers and 3 viewport sizes for a total of 21 cross environment combinations.

Execute the Ultrafast Cross Browser Testing Hackathon - In order to recruit testers and incent them to take 15 to 20 hours of their time to learn about modern cross browser testing, we gamified the research and created a hackathon. This hackathon delivered 100 prizes worth US $50,000 including the two $5,000 Grand Prizes and ten $1,000 Platinum Prizes for the testers who did the best job against a set of pre-established criteria. Engineeers had from June 1st until July 7th, 2020 to submit their work. Winners were announced in late July 2020.

Recruit Representative Testers - Testers were recruited using a variety of tactics including social media, paid advertising, and direct marketing. Any tester, anywhere in the world could qualify. Our goal was to obtain at least 100 submissions, but in the end we received 203 submissions from among 2,224 participants. We ended up with a highly representative data set geographically, technographically, and demographically.

Represent All Major Test Runners, Frameworks, and Binding Languages - Testers qualify by successfully completing the challenge using their preferred tradition cloud testing solution leveraging any test runner, framework, and binding language they want. They then repeated the effort using Applitools Ultrafast Test Cloud using one of our 50+ SDKs aligned to their preferred environment.

Judging Submissions - Highly experienced quality engineers judged every submission. 100 points were possible for each test framework for a total of 200 points in all. Points were awarded based on

- **Test Creation and Execution Time** - How code and time-efficient is test creation? How quickly do tests run?
- **Test Coverage and Analysis** - How easily and quickly are results analyzed? Did the tests catch all the bugs?
- **Team Analysis and Collaboration** - How easily and quickly are bugs logged and resolved across the team?
- **Test Stability and Auto-Maintenance** - How easy and quick is it to maintain tests? Do all tests execute successfully?

Aggregate and Analyze the Data - Data from each individual submission was logged, blinded to protect the identity of any individual, and then aggregated prior to analysis. Standard research quality control procedures were used to ensure that any result included was valid and met all the criteria for inclusion. Feedback was also obtained from all 203 submitters and permission to use their name and a quote was obtained prior to publication.

Release the Findings - Findings were analyzed and released on August 12th, 2020. Additional details by use case will be released throughout 2020. Academics or analysts who want to access more details should contact us at ImpactofVisualAI@Applitools.com

Measure Definitions

Time To Complete Full Test Cycle - The amount of time in hours and minutes required to complete a full test cycle (e.g. install, write, run, analyze, report, and maintain) comparing v1 of the hackathon app to the new v2 candidate of the hackathon app cross all 21 combinations for all 3 test cases.

Test Code and Screenshot Efficiency - Number of lines of test code and screenshots used to complete a full test cycle expressed as an absolute number.

Engineer Satisfaction (also called Net Promoter Score) - Net Promoter Score is a metric that poses a single question to users asking their "likelihood to recommend a product or service" to gauge their overall satisfaction. It uses a scale of 0 (not at all likely to recommend) to 10 (extremely likely to recommend). See the complete explanation on page 6.

Time to Install - The amount of time in hours and minutes required to install the cross browser testing framework.

TIme to Write - The amount of time in hours and minutes required to write the cross browser tests required to compare v1 of the hackathon app to the new v2 candidate of the hackathon app across all 21 combinations for all 3 test cases.

Time to Run - The amount of time in hours and minutes required to run the cross browser tests required to compare v1 of the hackathon app to the new v2 candidate of the hackathon app across all 21 combinations for all 3 test cases.

Time to Analyze - The amount of time in hours and minutes required to analyze all results of the cross browser tests required to compare v1 of the hackathon app to the new v2 candidate of the hackathon app.

Time to Report - The amount of time in hours and minutes required to report all bugs identified in the cross browser tests comparing v1 of the hackathon app to the new v2 candidate of the hackathon app across all 21 combinations for 3 test cases.

Time to Maintain - The amount of time in hours and minutes required to fix all test code that threw errors or false positive results for the cross browser tests comparing v1 of the hackathon app to the new v2 candidate of the hackathon app across all 21 combinations for 3 test cases.

Number of Locators - Number of DOM locators used to provide test coverage for the cross browser tests comparing v1 of the hackathon app to the new v2 candidate of the hackathon app across all 21 combinations for 3 test cases.

Time Learn Modern Cross Browser Testing - The amount of time in hours and minutes required to learn modern cross browser testing using Applitools Ultrafast Grid powered by Visual AI.

Data Dimensions

RESEARCH SEGMENTS

Hackathon Applicants - Any quality engineer that signed up for the Visual AI Rockstar Hackathon. There were a total of 2,224 applicants.

All 203 Submitters - Any quality engineer that successfully completed the hackathon project. This is the full sample used as a foundation for the study and amounted to 3,112 hours of quality engineering data.

Top 100 Winners - The top 100 quality engineers who secured the highest point total for their ability to provide test coverage on all use cases and successfully catch potential bugs using both code-based and Visual AI approaches.

Grand Prize Winners - The top 12 quality engineers who scored the highest on the hackathon.

TECHNICAL DIMENSIONS:

Traditional - Cross browser testing done using old-school cloud testing solutions (e.g. SauceLabs, BrowserStack) or homegrown solutions built on top of device farms like AWS.

Modern - Cross browser testing done using Applitools Ultrafast Grid powered by Visual AI.

Glossary of Key Terms

Cross Browser Testing (also Cross Environment Testing) - A way to test if an application is functioning and visually appearing as intended across different browsers and browser versions, operating systems and versions, devices, and viewport sizes.

Traditional Cross Browser Testing - This is an approach to cross browser testing developed over 10 years ago that typically uses virtual machines to call real devices in an external cloud environment to render a candidate application for testing purposes. This approach relies heavily on locators and test code subject to technical debt that limits its utility and scale for modern apps.

Modern Cross Browser Testing - This is an approach to cross browser testing developed as a result of 1) standardization of browser behavior through the W3C 2) the emergence of microservices technology and device emulators and 3) the emergence of Applitools Visual AI. This approach relies on screenshots and very few locators which eliminates almost all technical debt and scales quickly and cheaply for modern apps.

CI/CD - The continuous integration and deployment of feature code used by modern engineering teams.

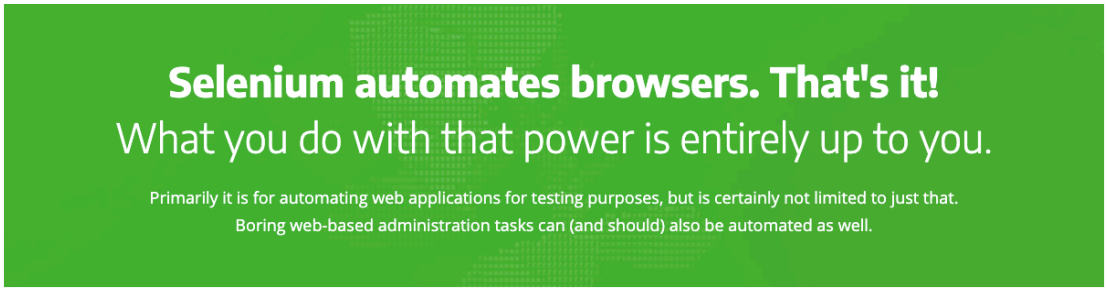
Digital Transformation - The 21st century transition of brand and companies from the physical world to the digital world across all aspects of the business from marketing to sales to delivery to support.

Shift Left - Describes a quality management approach whereby feature developers assume responsibility for the quality of the features they develop often leveraging unit testing and other automated testing techniques applied prior to code check in.

Visual Testing - Testing a web or native mobile application by looking at the fully rendered pages and screens as they appear before customers. This was historically done manually or by using error prone pixel matching and DOM based tools, but more recently Applitools Visual AI has modernized and automated visual testing with 99.9999% accuracy and made it vital to Agile and CI/CD DevOps processes.

Glossary of Key Terms

Browser Automation Tools - Browser automation tools connect to browsers and automate navigation, button clicks, data entry, and the return of this data from a DOM element within the browser. This leads to their main use of automating different types of common user paths and tasks through the browser. One major application of these browser automation tools is testing, but others include web scraping, taking screenshots, etc. Note that, despite the popular misconception, these tools are not purpose built test automation tools, but rather browser automation tools re-purposed for testing as one of their main use cases. The largest browser automation tool is Selenium which emerged in the late 90's as browser came on the scene.



Cypress (https://cypress.io) - A JavaScript based Test automation tool. You can find more info here

Webdriver.IO (https://webdriver.io/) - A JavaScript based Test automation tool. You can find more info here

Selenium (https://www.selenium.dev/) - A popular browser automation tool that's used heavily for functional testing. You can find more info here

Test Automation University (https://testAutomationu.com) - Free test automation courses with videos, transcripts, quizzes, credits, ranks badges, and certificates!

Visual AI Rockstar Hackathon - A global online hackathon that Applitools ran in November 2019.

Test Creation - Authoring or writing automation tests most often using a programming language (such as Java), an assertion library (JUnit.assert), a test framework(Junit) and a browser automation tool (Selenium).

Test Maintenance - Updating previously created tests that are failed in the new run or fixing tests that are throwing false positives because of the changes in an application that's being tested.

Glossary of Key Terms

Test Locators - These are DOM element locators that are used by the browser automation tool to locate the element within the DOM and then interact with that DOM element. These interactions, for example, could be clicking or returning values displayed in a DOM element.

Test Labels - These are actual texts that are displayed on the app. For example: Error messages, Field labels etc. In order to verify that text is actually displayed, test automation engineers need to copy that label and hard code the value as an expected result. Then, they compare this hard-coded text with the actual value in future test runs to verify quality of the new candidate.

Test Code Stability - Stability is a description of how often your tests fail or throw false-positives due to changes in Test Locators and Test Labels. If a test uses a lot of locators and labels, then it's considered less stable because these locators and labels can change in insignificant ways at any point leading to false-positives.

Test Coverage - A description of the amount test coverage a quality team is providing an application expressed as a percentage. 100% coverage is the desired goal. While that 100% goal not achievable or even necessary in practical terms, current coverage levels are generally considered inadequate, to slow to be achieved, or both, for modern apps.

About Applitools

Applitools delivers a Next Generation Test Automation Platform through Visual AI and Ultrafast Grid. We enable engineering teams to release high quality web and mobile apps at incredible speed and at a reduced cost.

Applitools Visual AI modernizes important test automation use cases -- Functional Testing, Visual Testing, Web and Mobile UI/UX Testing, Cross Browser Testing, Responsive Web Design Testing, Cross Device Testing, PDF Testing, Accessibility Testing and Compliance Testing -- to transform the way organizations deliver innovation at the speed of CI/CD at a significantly lower Total Cost of Ownership (TCO).

Hundreds of companies from verticals such as Tech, Banking, Insurance, Retail, Pharma, and Publishing -- including 50 of the Fortune 100 -- use Applitools to deliver the best possible digital experiences to millions of customers on any device and browser, and across every screen size and operating system.

Applitools is headquartered in San Mateo, California, with an R&D center in Tel Aviv, Israel. For more information, please visit applitools.com.

Contact:

Jeremy Douglas
303-589-1941
jdouglas@catapultpr-ir.com

SOURCE Applitools

Related Links: http://www.applitools.com

LEARN MORE AT

applitools.com

About the Authors

Raja Rao

Sr. Director, Head of Global Growth at Applitools
linkedin.com/in/rajaraodv/

Raja has had a unique and diverse career from dev to evangelist to growth marketer. After receiving his Masters in CS from UIUC in 2002, he worked as a QA automation engineer, team lead, then architect for the next 7 years. Raja won awards for moving Yahoo! from QTP and homegrown testing to Selenium RC and TestNG back in 2008. TestNG was truly NextGen at the time. After a transition to Front End development, Raja was instrumental in evangelizing Salesforce and was part of the wildly successful Trailhead learning platform, and co-created Test Automation University after joining Applitools.

Like Yahoo! in 2008 and Salesforce in 2013, Raja joined Applitools recognizing the huge opportunity to contribute back to the QA community by helping them learn modern testing through Visual AI. His course on Test Automation University entitled “Modern Functional Testing Through Visual AI” led to this report and is a must for every quality engineer.

James Lamberti

Chief Marketing Officer at Applitools
linkedin.com/in/jameslamberti/

James is a global go-to-market executive who specializes in helping B2B companies position complex technology solutions for a faster and easier sale. James has experience both in large global companies (Experian, Clorox) as well as growth start-ups scaling to $100m+ ARR (Applitools, comScore, InMobi, AdTruth, and now Applitools). James’ expertise lies in demand generation, product marketing, and account-based marketing in combination with enterprise sales experience and go-to-market strategy.

James is obsessed with creating a mutual exchange of value between prospects and customers and the companies he represents -- a principle too often absent in modern sales and marketing organizations.