Poster PP08

Ensuring Confidence in Trial Simulations: An Automated Testing Pipeline

Anoop Rawat, Cytel Statistical Software & Services Private Limited, India, Delhi
Hema Patil, Cytel Statistical Software & Services Private Limited, India, Pune

ABSTRACT

As clinical trial designs become more sophisticated, statisticians rely increasingly on simulation tools to evaluate operating characteristics—yet manually testing these tools can be time-consuming and error-prone. East Horizon, Cytel's platform for trial design and simulation, offers a rich set of features for examining results, but its complexity demands systematic verification. To address this, we developed an in-house automation testing framework that defines test scenarios across diverse trial designs, compares outputs against acceptance criteria, and generates summary reports for continuous integration. In this poster, we apply the framework to a representative Phase 3 design, demonstrating how automated execution uncovers subtle discrepancies, expedites verification cycles, and sustains confidence in simulation outputs while reducing manual effort.

INTRODUCTION

Modern clinical trials increasingly rely on complex statistical designs to improve efficiency, ethical balance, and decision-making. These designs often involve adaptive features, interim analyses, and sophisticated decision rules that require rigorous computational evaluation. Ensuring the correctness and reliability of such evaluations is critical, as errors can directly impact trial conclusions and downstream regulatory or scientific decisions.

East Horizon is a clinical trial design platform that performs core statistical computations required to evaluate complex trial designs. These include the calculation of test statistics and decision rules, estimation of operating characteristics such as power and Type I error, and generation of interim and final analysis outcomes. To support deep design understanding, the platform provides rich analytical tools, including interactive tables, visualizations, filters, and scenario comparisons that allow users to explore results across multiple dimensions.

As the platform evolves, new trial designs and methodological enhancements are continuously added. While this innovation is essential, it introduces the risk of unintended impacts on existing designs and functionalities. Given the complexity of both the statistical algorithms and the supporting software infrastructure, traditional manual testing approaches are insufficient to ensure long-term correctness and stability.

This paper presents an automated testing framework designed to validate numerical correctness, user interface behavior, and end-to-end integration within East Horizon. The framework is built around the user journey, combines multiple technologies, and supports both feature-level validation and long-term regression testing, enabling scalable and reliable platform development.

MOTIVATION FOR AUTOMATED TESTING IN CLINICAL TRIAL DESIGN PLATFORMS

Clinical trial design platforms present unique testing challenges due to the high dimensionality of inputs, the complexity of statistical computations, and the need for precise and reproducible results. Each new design or enhancement must be validated across multiple fronts, including numerical accuracy, correct user interface behavior, and proper integration between frontend, backend, and computational engines.

Manual testing in this context is time-consuming and difficult to scale. Verifying numerical outputs across a wide range of scenarios, interacting with complex visualizations, and validating multiple user workflows requires substantial effort and domain expertise. Moreover, manual testing is inherently difficult to repeat consistently, increasing the risk of undetected errors.

Importantly, testing does not end when a feature is released. As new designs are introduced and existing components are modified, previously correct functionality may break due to unforeseen interactions or dependencies. Without systematic regression testing, such issues may remain unnoticed until they affect users.

Automated testing addresses these challenges by enabling repeatable, comprehensive, and efficient validation. It allows numerical results, visual outputs, and functional behaviors to be verified continuously as the platform evolves. In high-stakes domains such as clinical trial design, where correctness is paramount, automation is not merely a convenience but a necessity.

AUTOMATED TESTING FRAMEWORK AND METHODOLOGY

The automated testing strategy for East Horizon is structured as a layered defense, with each layer designed to detect different classes of errors as early as possible while supporting end-to-end validation of the user experience. The very first line of defense consists of unit tests written by developers during feature implementation. These tests target unit-level components and verify the correctness of small, isolated pieces of functionality. Once added, the full suite of unit tests is executed automatically every time a new feature or enhancement is introduced, ensuring that foundational logic remains stable as the codebase evolves.

Following unit-level validation, the core statistical engine is tested by statisticians to ensure that newly implemented algorithms behave as intended. This validation is performed in R using appropriate statistical methodologies tailored to the design under consideration. For example, trend testing is used to verify expected monotonic behavior of operating characteristics across design parameters, while parallel programming approaches are employed in extreme or high-risk scenarios to independently replicate simulation results. At the completion of this step, a diverse set of validated test scenarios is available for the new feature. These scenarios, together with their corresponding results, serve as authoritative benchmarks for all downstream automated testing.

Once benchmark scenarios are established, API-level testing is performed to ensure that user inputs are correctly interpreted and passed through the system. Using the validated scenarios, automated tests confirm that the backend APIs correctly translate inputs into calls to the core engine and return outputs that match benchmark results.

Testing of the results exploration interface presents a significant challenge due to the interactive nature of tables, charts, and filters. To address this, East Horizon employs, among others, an in-house, Python-based automated testing framework designed specifically for UI-driven validation. In this framework, test scenarios are defined once based on the testing requirements, including design parameters, expected numerical outputs, and expected visual patterns. When a test is executed, the framework programmatically creates a design, runs simulations, navigates to the results section, performs a sequence of user-like actions, and compares the resulting tables and charts against established benchmarks.

For example, in a Phase 3 design with interim and final analyses, the framework opens a window that logs in a test user, creates an empty project, enters prespecified inputs through the UI, and simulates the project. It then checks whether the various results tables—such as look-wise efficacy and futility counts and the boundaries—are as expected.

In several instances, this automated execution has uncovered subtle discrepancies that would have been difficult to detect through manual testing alone, such as a missing important result from the results section. By detecting these issues early, the framework significantly expedites verification cycles and reduces reliance on repeated manual checks. More importantly, it sustains long-term confidence in simulation outputs by ensuring that numerical correctness and UI behavior remain aligned as new features and designs are introduced.

Overall, this integrated testing approach—spanning unit tests, expert statistical validation, API testing, and UI-driven automation—enables comprehensive verification of East Horizon while substantially reducing manual effort. The Python-based framework plays a central role in bridging the gap between backend correctness and user-facing reliability, making it possible to test complex clinical trial designs in a scalable and reproducible manner.

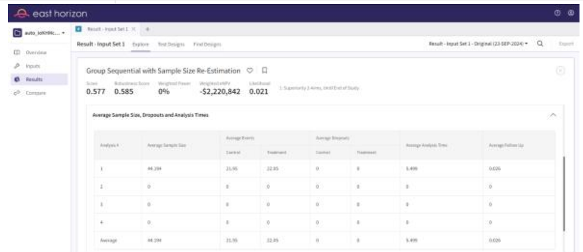
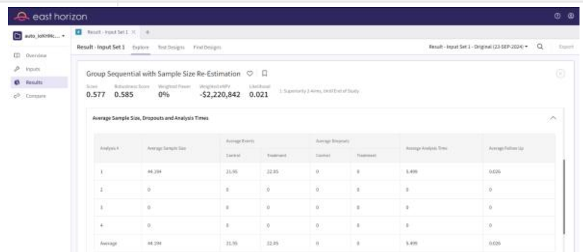
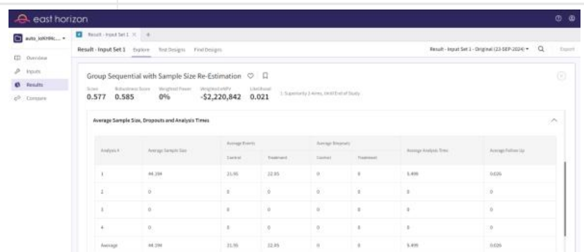
PASS: Section opened successfully	
PASS: Both expected and actual excels have same rows and columns	
PASS: All the values of both expected and actual excels match	

Table: Automated test report indicating pass/fail status of the UI actions

IMPROVEMENTS

While the current automated testing framework provides broad coverage across numerical, functional, and integration testing, several opportunities exist for further enhancement. One promising direction is the incorporation of AI-enabled risk-based testing, where test cases are prioritized based on the likelihood and potential impact of failure. Such an approach can focus testing resources on the most critical and error-prone areas of the platform.

Another future direction is automated or semi-automated test planning. By leveraging metadata about designs and historical failures, the system could assist in generating relevant test scenarios with minimal manual intervention. This would further reduce the effort required to maintain comprehensive test coverage as the platform grows.

AUTHOR CONTACT

Anoop Rawat – Anoop.Rawat@cytel.com