



UAT for Performance Transformations Is Notoriously Difficult. It Doesn't Need To Be.

User Acceptance Testing (UAT) can throw projects into a frustrating and mind-numbing stall, which is catastrophic to any transformation. The terror of botching a go-live event with insufficient data commonly leads users into a frenzy of checking, double-checking, approving, re-checking, and reapproving. As the process repeats across test cases, progress toward user confidence seems negligible. Performance UAT presents a unique environment where simple data validation between legacy and target state platforms will not suffice. Testing to yield no data differences is unrealistic and not the desired outcome. Instead, they explain the differences in the art of Performance UAT.

What Makes Performance UAT Tricky?

- Changing methodologies
- Different market data assumptions
- Service level changes
- Discrepancies can be valid
- Large data volumes

LARGE DATA VOLUMES COMPOUND THE DIFFICULTY

The sheer volume of data for testing compounds the difficulty of testing. Who can execute such a large-scale effort? With what tools? Everything from data extraction, transformation, and testing must be a conscious and coordinated effort. Additional consideration of time and computing power becomes part of everyday execution. While sampling is a common workaround, predefining all test cases and running into no new test case information through testing is unlikely. Samples that do not represent the population will disguise problems negatively impacting the Performance Conversion and future data-centric projects.

PLAN FOR UAT BEFORE UAT

Ideally, the planning for UAT starts early (as far back as defining requirements). Successful Performance UAT groundwork includes clearly articulating test cases and expressing what will elicit user approval. Data assessments help evaluate test complexity. Effective resourcing of the testing effort in people and tool selection fortifies future success. Psychologist Silvan Tomkins wisely said it's the "...tendency of jobs to be adapted to tools, rather than adapting tools to jobs. If one has a hammer, one tends to look for nails..." A proper approach to resourcing requires a deeper toolbox than just a metaphorical hammer (Excel).

At Meradia, our approach to Investment Performance UAT starts with a strong plan. Whether users intend to replicate legacy calculations or change methodologies in favor of progress toward industry standards impacts UAT execution. Predefined success criteria act as a North Star to follow in UAT's dark and scary journey. Next, our Performance Data Readiness Assessment identifies missing data elements, exposes inadequate data integration, and determines the compatibility of data structures and contents.

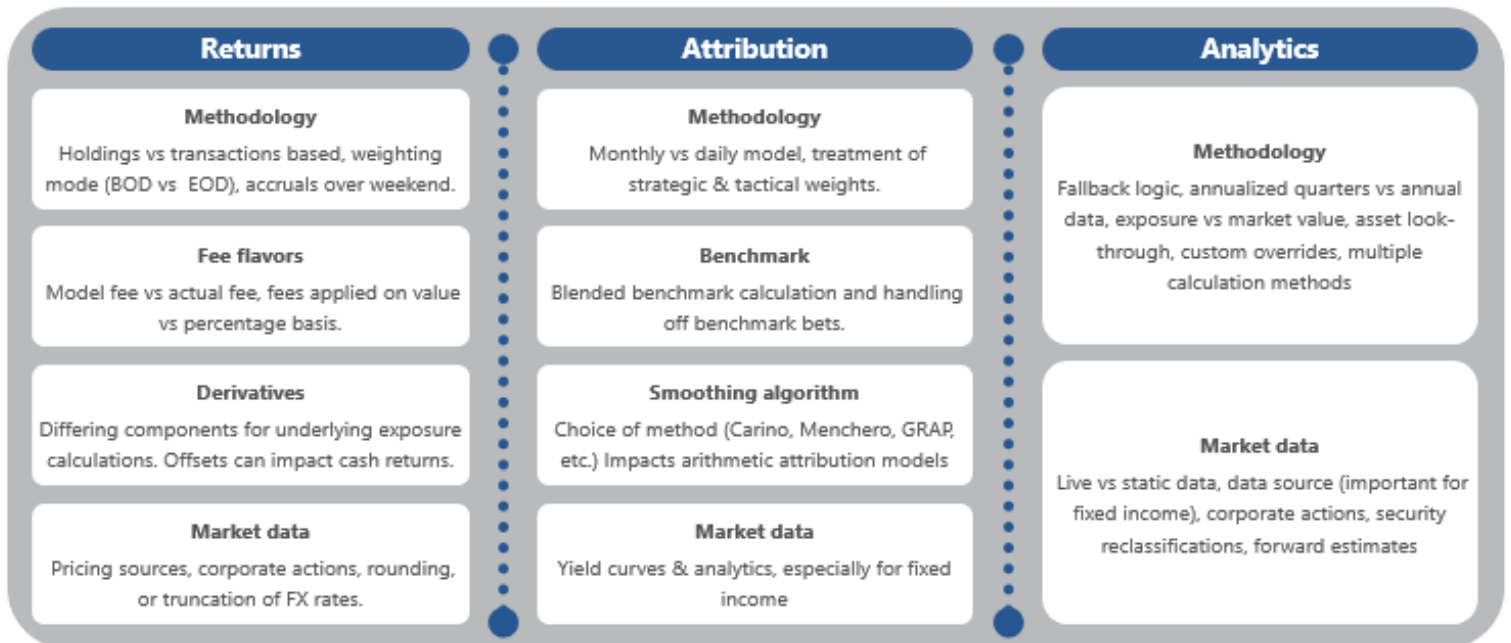
UAT for Performance Transformations Is Notoriously Difficult. It Doesn't Need To Be.

PLAN FOR UAT BEFORE UAT (continued)

The assessment aids in evaluating test complexity. With goals established and data assessed, practical resourcing tools and personnel round out the planning stage. The people and tools involved in Performance UAT will make or break the testing effort.

COMPONENTS OF TEST COMPLEXITY: DOMAINS, FORMATS, AND VOLUME

Test complexity is the product of data domain, format, and volume. The data domain, or data content, introduces unique complexities for consideration. Testing performance returns vastly differ from testing analytics. Further, testing attribution results is even more complex as both the underlying returns and the attribution results should be evaluated when explaining differences. Various domains demand in-depth knowledge, which many analysts designated for Performance UAT need to improve. Novices extend timelines as they must ongoingly consult SMEs or, more frequently, research as they become "surface-level SMEs" to have any hope of completing testing.



Valid Discrepancies Across Performance UAT Data Domains
Source: Meradia

Next, data format, specifically differences in data format between legacy and new systems, adds a dimension of complexity to Performance UAT. Any differences in format between sources require keys or restructuring to link the opposing data into a testable configuration.

Consider these examples:

Example 1

Legacy System

Evaluates securities based on multiple identifiers to determine unique securities.

New System

Evaluates securities based on a single identifier based on a waterfall of many identifiers.

Conundrum

Securities will not match with a "Join" or "Look-up" (depending on SQL or Excel use). To avoid securities falling out of testing, the waterfall logic must be replicated using the legacy system's identifiers as a step of testing.

UAT for Performance Transformations Is Notoriously Difficult. It Doesn't Need To Be.

COMPONENTS OF TEST COMPLEXITY: DOMAINS, FORMATS, AND VOLUME (continued)

Example 2

Legacy System

Portfolio level analytics are stored under common columns with complimentary dimensional columns specifying the variety of the analytic. The same portfolio can have multiple rows.

Portfolio Identifier	Calculation Method	Price to Earnings
Port01	TTM	12.3
Port01	Forward 12m	14.8

New System

Portfolio level analytics are stored in different columns for each calculation methodology of a given analytic. Each Portfolio has a single row.

Portfolio Identifier	P/E TTM	12m-Forward P/E
Port01	12.6	15.2

Conundrum

Comparing the correct analytics of a portfolio becomes cumbersome at scale as no "join" or "look-up" will span the logical gap. To solve, the legacy data must be unpivoted on the common analytic column (Price to Earnings), concatenated with the calculation method column and then mapped from the concatenated name to the new system's analytic naming convention all to be re-pivoted to match the new system's format. Pivots are amongst the most "expensive" computations and commonly cause common analyst tools like Excel to crash.

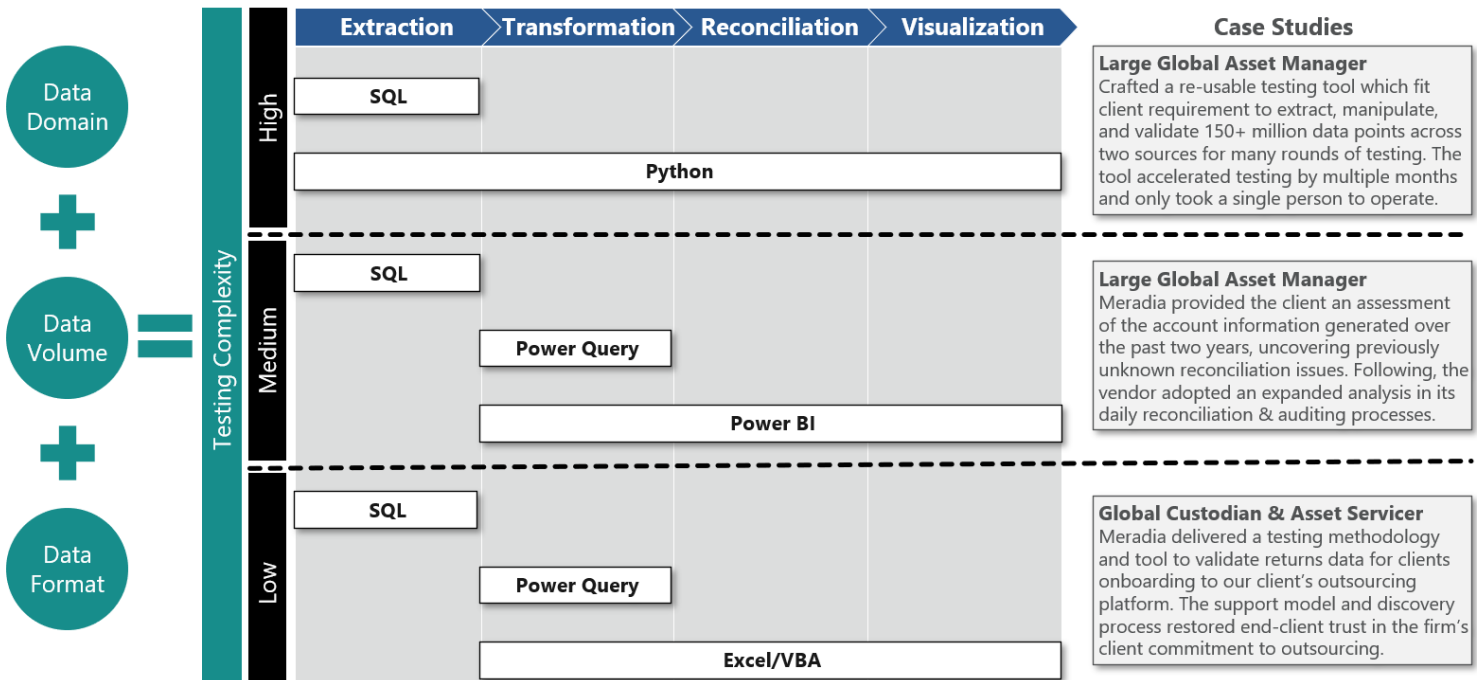
Source: Meradia

Data volume is the 3rd dimension of test complexity. Large data volumes in Performance UAT require systematic testing solutions with replicable and robust functionality. With scalable and efficient approaches to data challenges that arise during UAT, testers will save time executing and likely re-executing test cases. Every minor hiccup compounds into a time-consuming hurdle with a large data volume unless testers apply scalable solutions from the start.

CHOOSE TOOLS TAILORED TO TEST COMPLEXITY

As stated, test complexity is the product of data domain, format, and volume, and letting test complexity dictate the tool used for Performance UAT is best practice. No single tool optimally addresses all Performance UAT scenarios, so a diverse toolbox lays the foundation for successful UAT. Throughout the testing effort, tools must extract data, transform data, reconcile data, and visualize test outcomes. Meradia's demonstrated toolbox of Power BI, Python, Excel, and SQL allows us to meet different test cases and complexities by matching the best tool to the job. The below graphic demonstrates how Meradia approached various Performance UAT scenarios with other tools.

Source: Meradia



UAT for Performance Transformations Is Notoriously Difficult. It Doesn't Need To Be.

TEST EXECUTION: EXTRACTION, TRANSFORMATION, RECONCILIATION, AND VISUALIZATION

Smooth execution of Performance UAT depends on the plan put forth beforehand. With a good plan in place, testing execution is more likely to succeed. Generally, execution includes data extraction, transformation, reconciliation, and visualization. Applying these four steps unravels test complexity into digestible results for users offering sign-off. Ideally, critical considerations are uncovered in testing, but stuff happens even when a plan is in place. So, how can one eliminate surprises?

Extraction

In data extraction, query your data in a way that limits the amount of transformation required to compare the two data sets. Effort upfront will enable nimbleness in the later stages of testing. This becomes increasingly important with large data volumes. When extracting data, make sure appropriate fields are included in the dataset. This may sound intuitive, but mistakes happen where least expected. Often, unmastered data will have many field variations intending to represent a similar thing with minute differences. Even mastered data is subject to this problem. Data extraction is the easiest step in Performance UAT.

Transformation

Trim the data as much as possible through data transformation before any "expensive calculations." Additionally, perform simple tests on your data throughout and enrich it before performing any reconciliation. For example, ensure all the fields in a date column are dates, check that all columns possess only a single data type, ensure no null values in key fields like identifiers, and evaluate how the two systems treat zeros and null values. Do they return the same value from both systems? Does the new system generate a 0 and the legacy a dash? Check whether the results are uniformly denominated. All these simple tests will save time when interpreting test results.

Reconciliation

Reconciliation is not what it sounds like. Simply comparing whether results between two systems equal each other will not suffice. Some data domains benefit from testing the underlying inputs of the results. Additionally, some analytics results require different tests, like differences in analytics between the same securities in a data set. These additional tests enhance testers' ability to conclude the dataset, ultimately accelerating the testing effort by identifying problematic items and red herrings. Once again, the extra upfront effort is better than re-work.

Visualization

Lastly, the visualizations created should illuminate problems for users and provide the necessary information for them to sign off on UAT. People respond to pictures and graphics far better than raw data. Visualizations incorporating critical test case dimensions prove the performance results or analytics generated will meet requirements. It also identifies essential problems efficiently and directly. Performance UAT requires less research into issues when the investigation is baked into the data visualization.

REPORTING TEST PROGRESS: COMMUNICATION AND DOCUMENTATION

Reporting of test execution results boils down to communication and documentation. Following the last aspect of execution, summarize your findings effectively, whether through visualization or a narrative. Little time should be spent on what went well; most of the time should be spent on what broke and why. That said, remember to document testing successes and gather approvals. Demonstrate progress through completion and approvals but drive progress in issue identification and solutions. Meet your audience at their level. Use appropriate data domains and granularity for different audiences. Report throughout the execution process, not at the end. Running these in parallel will save many headaches.

WRAPPING IT UP

Without a seasoned guide, the intricacies of UATs create ripe conditions for transformational catastrophe. A proper plan is the foundation of success. Execution with the right tools and staff is critical. The cherry on top is creating transparency around the testing effort through reporting. A problem in any aspect will compound into issues throughout. Incorporating best practices across these steps rejuvenates failing Performance Data Conversions and fosters stability and confidence for future efforts.

UAT for Performance Transformations Is Notoriously Difficult. It Doesn't Need To Be.

HOW MERADIA CAN HELP

User Acceptance Testing (UAT) is a necessary step in any Performance Data Conversion. Meradia's tailored approach to Investment Performance UAT leverages modern technologies to efficiently ensure confidence in your performance data. Don't let UAT destroy your Performance Data Conversion. Trust Meradia's experience and expertise to help your project succeed.



Clay Corcimiglia

Clay Corcimiglia supports Meradia's clients through his proficiency in data mapping, data confirmation analyses, and data visualization to streamline operations between the front and middle office. While working for a global investment manager, Clay worked through Python and SQL to perform data confirmation analysis. Clay successfully evaluated the firm's performance data and provided guidance on best practices for their future systems. Clay enabled the client to achieve their end goal of a new, complete global data integration.