Mindtree
Welcome to possible

Leveraging a LEAN model of catalog-based performance testing for quality, efficiency and cost effectiveness

Traditionally, organizations have leveraged project-based models for performance testing. Either a central performance testing team is engaged or performance testers are on-boarded as part of the project team to understand, estimate and operate on a project basis. In addition, mature project execution methodologies allow for a well-defined test strategy phase during the initial phases. While this model has served enterprises well, it does not allow for organizational learning to improve efficiency and quality, or significantly impact cost.

Some of the key issues are:
- Extended intake process that involves estimations for every application under test
- Extended project lifecycles to develop turnkey test strategies for every project
- Unpredictable cost and quality
- Overheads in alignment with SLAs, acceptance criteria and related documentation approvals

Enterprise learning is not leveraged as effort is duplicated with every projects

| Define | Setup & implement | Validate | Deploy |

- SLAs
- Acceptance criteria
- Process
- Review & approvals
- Tool set up

- Test data management
- Scripting
- Scheduling
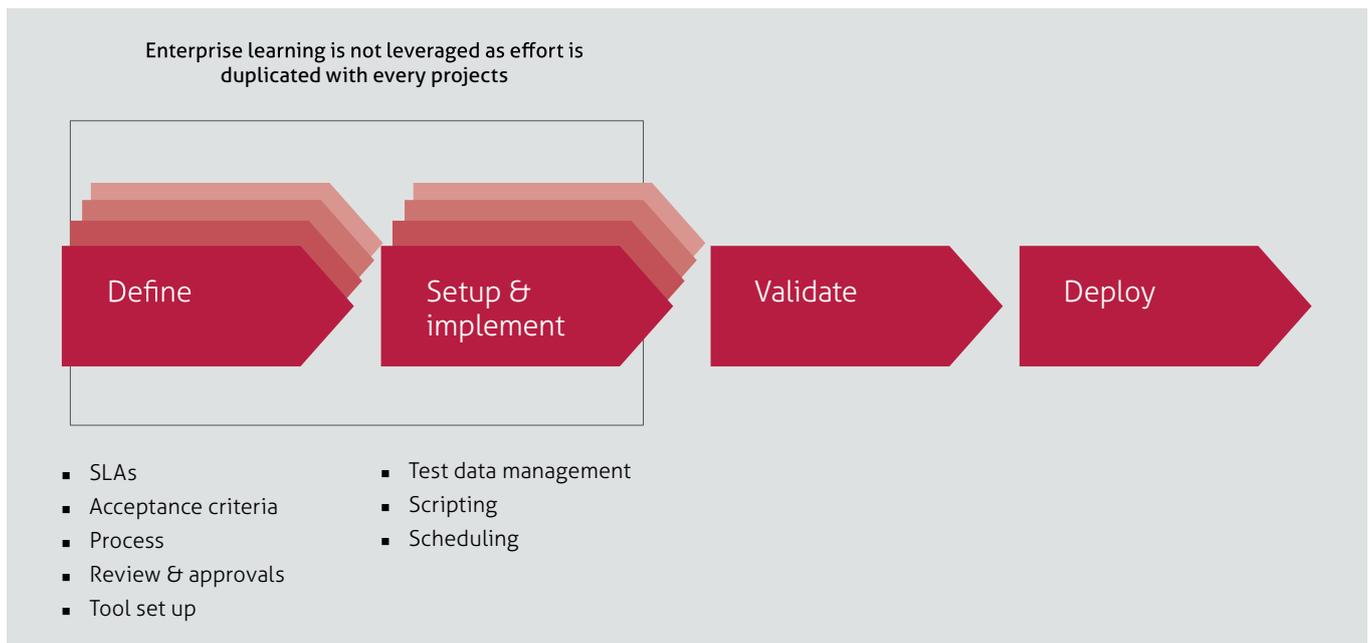
Catalog-Based Models

We are all familiar with ordering food in a restaurant by picking from a menu, thereby saving time and gaining certainty with respect to costs and the dish that will be served. Catalog-based models are similar to the restaurant experience, where enterprises pick the service needed, understand the standard cost and only review the end result. This eliminates touch points in the process that do not add value; and creates a LEAN organization.

Pick & customize if needed

Catalog
- SLAs process
- Acceptance criteria
- Reviews & approvals
- Tool setup
- Test data management
- Scripting & scheduling

Refine, update & add

| Define | Setup & implement | Validate | Deploy |
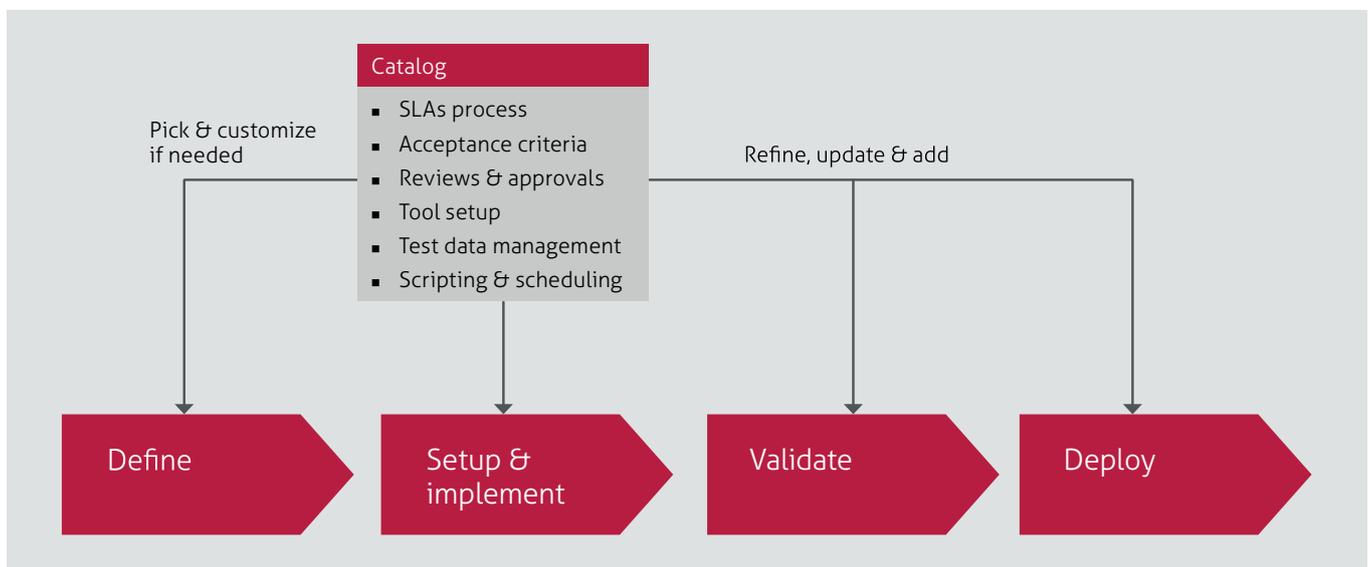
Catalog-Based Performance Test Factory

The essential components of a catalog-based performance test factory are depicted below:

Application portfolio

Predefined catalog

- Application type
- Performance test type
- Number of scenarios
- Number of vUsers
- Number of cycles

Category 1 apps

Category 2 apps

Category 3 apps

Service manangement

- Performance test requests
- Performance test guidance
- Continuous reading
- Real-time interaction
- Performance test reports
- Performance test checklists
- Proven quality
- Quality metrics
- Digital delivery
- Innovation
- Performance intake forms
- Performance test calendar
- Labs

Service manangement

Performance & tools

Performance test activities

Metrics & dashboards

A Service Catalog

A service catalog is a menu of performance tests that a customer can order from; and the dimensions of chargeback.
A catalog consists of the following components:

Foundational	Core parameters	Environment
■ Application types ■ Application lifecycles ■ Portfolio dependencies ■ Process maturity	■ Type of test ■ Number of cycles ■ User load ■ Number of scenarios	■ Enviroment setup ■ Issue resolution ■ Application support ■ Data management ■ Reporting

Chargeback specifics

Governance, risk management & issue resolution

Catalog updates

Continous improvement

Service Management

Once a catalog has been defined, an important aspect of the factory is service management. For instance, there must be a clear definition of the format in which service requests are accepted, acknowledged and serviced.

Key components of service management for successful delivery are:

Intake Mechanism

A template in which performance requirements are provided is essential and must be based on the service catalog. It is important to define a standard repository and format so that submission and monitoring is transparent. Building a process-based intake mechanism also allows for generation of important metrics for continuous improvement. This could be through an intranet site or a document repository.

Demand Management

Encouraging business users to prioritize requests helps resolve conflicts arising out of resource constraints, rollout dependencies and QA dynamics.

Acknowledgement & Verification

Every request must be verified for completeness and acknowledged as per pre-defined SLAs.

Schedule & Estimate Validation

All requests require review to ensure they fall within predefined parameters. Schedule & estimate validation is also an important point to flag exceptions which go towards update and refinement of the service catalog.

Communication

As the request is estimated, validated and approved, alignment of cross-functional teams such as infrastructure, security and application support is important to optimize performance test calendars and identify risks.

Reporting

This step ensures detailed and high-level reports as required of both the test results and other process-related metrics required for governance. Generally transactional reports are automatically generated. The data is then supplemented with process metrics for reporting to various levels of management.

Knowledge Management

As lessons are learnt in execution, catalog updates, communication and reporting, these must be disseminated to all stakeholders to improve the effectiveness of the model as a whole. A centralized performance wiki for best practices and guidelines is an excellent vehicle to promote both structured as well as unstructured collaboration.

Service Delivery

Once the request is acknowledged and scheduled, the performance factory scripts, executes and delivers the performance reports back to requestor. The performance test life cycle is managed by the factory based on agreed upon SLAs. Both repeat executions and new applications go through the lifecycle depicted below.

Test planning & designing	Test scripting	Test execution	Test execution
■ Workload model ■ Performance test scenarios ■ Tool & connectivity variation	■ Creation of test scripts ■ Unit testing of scripts ■ Test Data Generation ■ Dry run on script	■ Load test ■ Stress test ■ Volume test ■ Scalability test ■ Endurance test ■ WAN / LAN test	■ Metric collection as designed ■ Test reports ■ Identified bottlenecks ■ Recommended tunings ■ Test sign-offs

Governance

Another key success factor for a Catalog-Based Performance Test Factory is a robust governance framework.
The factory enforces governance at three levels to achieve the following:

■ Improved efficiency around performance testing
■ Reduced time to market
■ Improved quality across application portfolios

Level 1
Factory SLAs

Level 2
Execution metrics

Level 3
Quality metrics
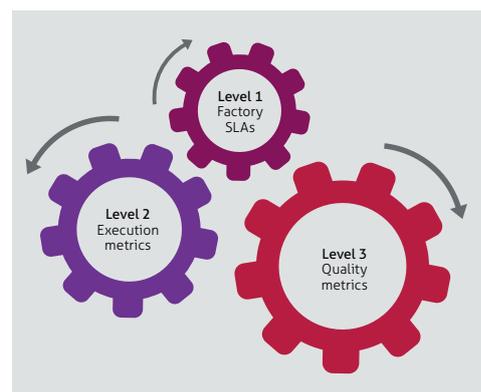
Level 1 – Factory SLAs

Each step in the process workflow is measured against a defined SLA that can eventually be linked to the payment model. Right from acknowledgement of the request to execution, the factory delivers against the agreed SLA. The SLAs can be refined as the engagement matures. Some examples of typical SLAs are:

- Acknowledgement Time – One business day
- Scheduled Execution – Seven business days from acknowledgement

Level 2 – Execution Metrics

At the Service Delivery level, the following metrics are helpful in driving excellence:

- Turnaround time
- Catalog-based execution metrics
- Deviation from execution SLAs

Level 3 – Quality Metrics

One metric that many enterprises measure is quality across various development suppliers in a multi-vendor setup. Performance test results across suppliers are collected and used to drive corrective action to improve quality.

A performance test factory is well suited to collecting and reporting quality metrics based on tests that are executed on a periodic basis. Key information captured for each supplier includes:

- Total requests
- Total passed requests
- Total failed requests
- Average number of cycles needed to meet performance criteria

Turnaround Time (days) - Cumulative

- 18%
- 16%
- 15%
- 13%
- 11%
- 09%
- 07%
- 05%
- 04%
- 02%

category based execution details
Month: May 2013

- Total requests
- Total executed
- Passed
- Failed

Vendor based execution details
Month: June 2013

- Total requests
- Total passed
- Total failed
- Avg no of cycles to pass
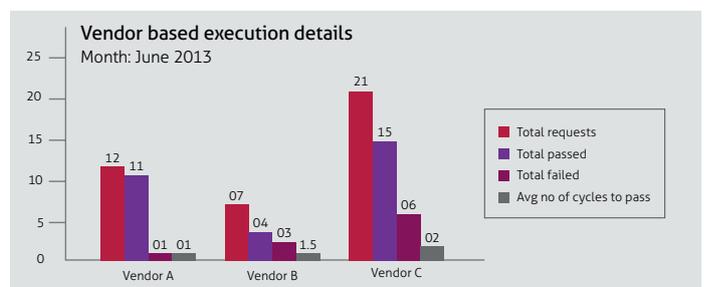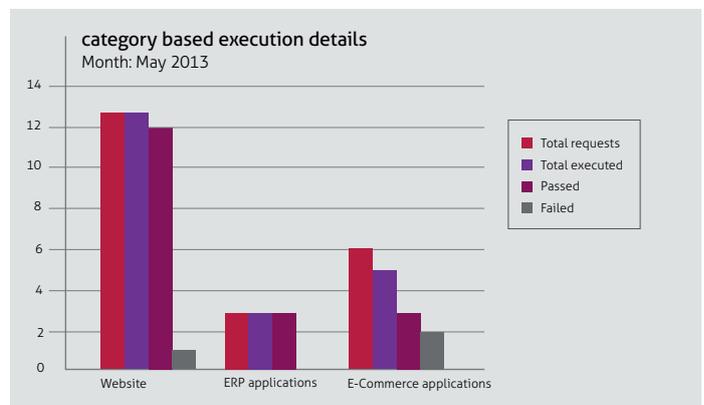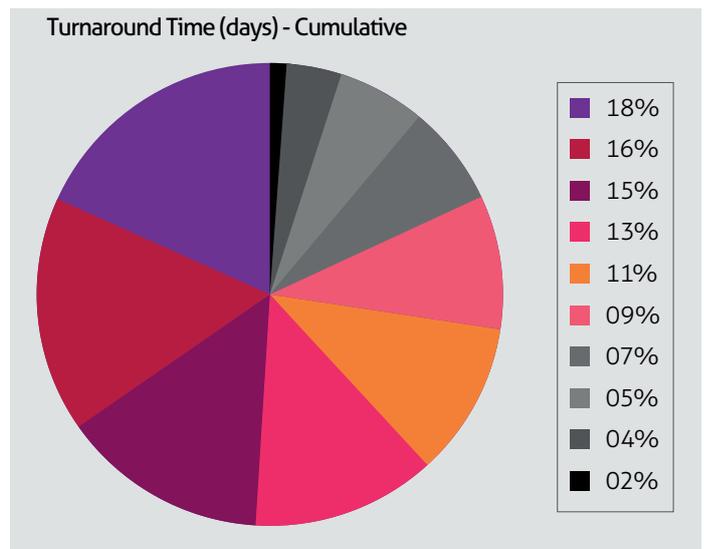
Conclusion

You should move to a Catalog-Based Performance Test Factory if:

- You plan to launch multiple applications to external customers and worry about performance
- You are concerned about the increasing costs of performance tests and need a more predictable pricing model
- You want to move away from a resource-based pricing model and face a challenge in making sure resources are available to script and run a performance test
- You need a common view of performance across all your applications or portfolios
- You want to significantly reduce the amount of non-

productive time your SMEs spend on ensuring that performance testers understand their needs

Catalog-based models provide enterprises with an alternative to conventional models while delivering cost, efficiency and quality. With the move towards LEAN testing, catalog-based engagement models also eliminate waste and ensure seamless scalability; with predictable costs. While organizational needs and application portfolios differ, catalog-based models can play a valuable role for selected portfolios; or alongside conventional models. The future is bright for the Catalog-Based Performance Testing Factory Model, and for early adopters of this innovative strategy.

About the author:

Nischal is Director of Testing for the Travel and Transportation vertical at Mindtree. He has worked extensively on transformational test strategies for customers in North America. He specializes in organizational maturity improvements through shared service units and customized engagement models.

About Mindtree

Mindtree is a global information technology solutions company with revenues of over USD 400 million. Our team of 11,500+ experts engineer meaningful technology solutions to help businesses and societies flourish. We enable our customers to achieve competitive advantage through flexible and global delivery models, agile methodologies and expert frameworks.