A TEAM. A SOLUTION. A PROCESS.

Performance Testing Made Simple

nü echo

» METHODOLOGY

1 Introduction

With the development and launch of the NuBot™ Automated Performance Testing Platform, Nu Echo has set itself apart within the Contact Center and telephony infrastructure market as the go to consulting firm for performance testing, quality control and assurance. NuBot helps take the risk out of full or partial, new or existing contact center deployments or telephony infrastructure testing by ensuring business requirements are met. Our clients bring us onboard to validate their end-to-end infrastructure and ensure their systems meet capacity requirements in order to deliver flawless and superior customer satisfaction. A well-built and tested contact center and telephony infrastructure ultimately results in satisfied clients and long-term sales potential.

… Nu Echo has set itself apart within the Contact Center and telephony infrastructure market as the go to consulting firm for performance testing, quality control and assurance.

2 About our Automated Performance Testing Practice

2.1 What is High Capacity Automated Performance Testing?

In the context of contact centers and telephony infrastructure, high capacity automated performance testing means the ability to place a high number of simultaneous calls to the tested infrastructure and validate that it performs as expected under the types of load that it is expected to handle. Each of these calls simulates a real person making a call into the system and performing a variety of tasks, as defined by the scenarios developed and programmed into the automated testing platform. Automated performance testing can also involve:

- Simulating contact center agents logging into the contact center platform and receiving the automatically generated calls, therefore making it possible to test the contact center system end-to-end;
- Testing outbound IVR applications by simulating users receiving the calls generated by the outbound application;
- Simultaneously simulating users making calls and users receiving calls, therefore making it possible to test telephony networks end-to-end.

2.2 When Should High Capacity Automated Performance Testing be Used?

Following industry best practices, this testing approach should be used prior to any new contact center or telephony infrastructure deployment to ensure its efficient and effective functioning. High capacity automated performance testing guarantees the integrity of your infrastructure even under extreme operating conditions. Other reasons for this type of testing may include the addition of new equipment, new software modules, new call routing strategies or the implementation of new processes in the contact center system.

> **High capacity automated performance testing guarantees the integrity of your contact center even under the most extreme market conditions**

2.3 What are the Benefits of Automated Testing?

Automated testing using Nu Echo's NuBot Platform provides the following advantages when compared to traditional manual testing:

- The number of simulated callers and agents can easily be increased as needed.
- A test using the same calling pattern can be repeated several times to reproduce a problem or to verify that a fix corrects a problem found in a previous test execution.
- All calls can be recorded along with a call event history for future listening or inspection. Various filtering criteria are available when listening to calls or analyzing their event history.
- Precise statistics from the caller's point of view can be systematically collected, including success/failure rate per scenario, application response time to caller actions, call setup time, and others.

2.4 What are the Benefits of a Full Service Project Testing Approach?

The full service testing approach means we focus on testing so you can focus on your core competencies/business. Nu Echo will manage test design & development, NuBot testing platform operation, results collection, results summary production, and final reporting. We provide project management services to coordinate these activities with our clients. There is no need for the client to invest time and efforts to learn how to design test projects, develop test scenarios, or operate the testing platform.

> **…we focus on testing so you can focus on your core competencies / business**

3 Performance Testing Methodology

3.1 Overview

Nu Echo uses the Testing Process Pipeline™ Methodology in its Performance Testing Practice. This methodology is based on an iterative approach, consisting of five stages: Design, Implement, Execute, Analyze, and Report. For each stage, some information or action (customer input) is expected from the customer and some deliverables are produced (stage output). Several iterations of the Execute and Analyze stages can occur in a project and the decision to perform an additional iteration is part of the Planning and Coordination. The process starts with a Discovery & Business Process Review with the client. Prior to designing or executing a test plan, Nu Echo's team of testing experts will get to know you, your business, and objectives with regard to your contact center. We want to understand what is important for you from your system in order to effectively propose ways to test and achieve optimal system performance. Our experienced team brings the industry's best performance testing practices to the table for you to consider and benefit from.

This methodology is based on an iterative approach, consisting of five stages: Design, Implement, Execute, Analyze, and Report.

Once discovery with the client is complete, we will produce a statement of work (SOW). The overall process is shown in Figure 1.

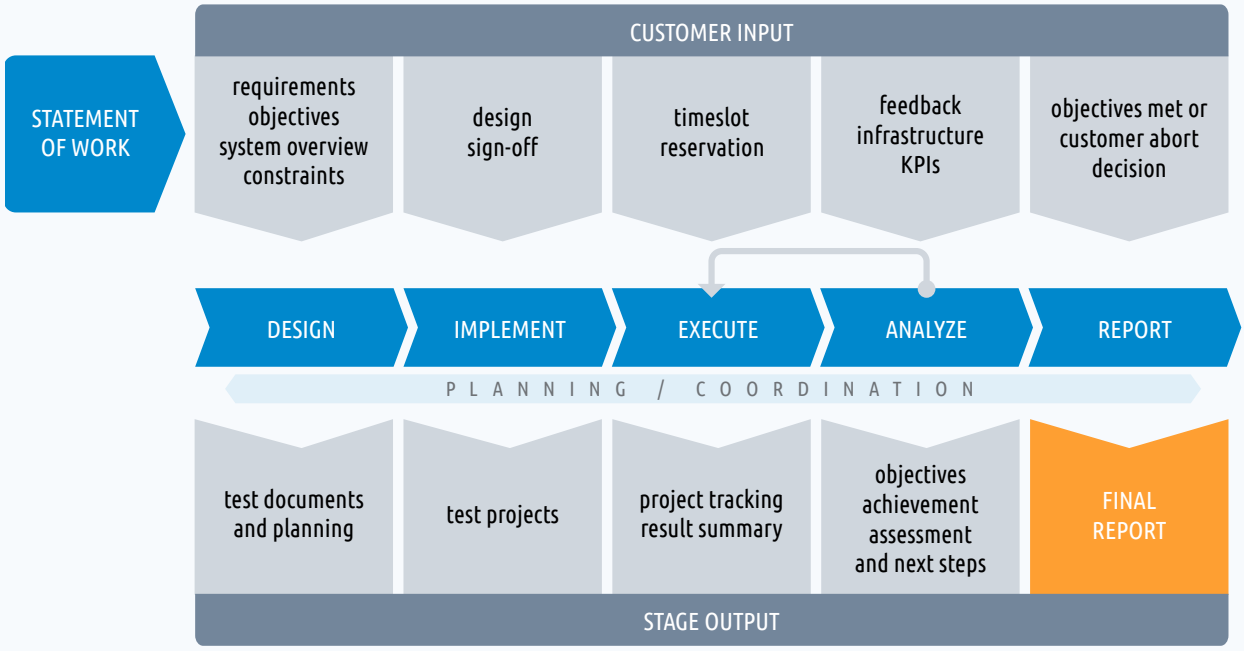

Figure 1 - Testing Pipeline Overview

The SOW contains budgetary information, preliminary estimates of resource requirements, along with a high-level overview of the project's objectives. It is worth mentioning that the SOW is a budgetary evaluation.

Depending on the number of defects found throughout the project, additional "Execute and Analyze" iterations may be required, therefore possibly affecting overall project costs and schedule. At project completion, the client will receive a final report.

3.2 Pipeline Stages

3.2.1 Design

The Design stage brings together elements that are likely to influence and shape subsequent stages of the pipeline.

Client input at this stage includes test requirements and objectives, client architecture, key performance indicators, as well as any relevant client specific constraints. Some of the previously listed information was required in order to produce the SOW, and will therefore be readily available. In order to collect the missing information, Nu Echo provides a questionnaire to be completed by the client with Nu Echo's support and guidance.

The deliverables, or outputs, for this stage are the project plan (detailing the main project milestones agreed upon by both parties) and the performance testing specifications document, which includes:

- the overall testing strategy
- high-level use cases
- detailed scenarios
- the testing architecture and tested components
- global test parameters
- a technical specification for each test and the associated objective to be met
- an initial grouping of test executions into scheduled working sessions or iterations.

3.2.2 Implement

The Implement stage is triggered by the design sign-off, which signifies that the deliverables produced in the previous stage have been accepted by the client. During this stage, NuBot scenarios and test descriptors are scripted and configured into the NuBot Platform, resulting in a NuBot test project. At this stage, all scenarios are tested at low call density, which implies that the customer infrastructure must be able to handle test calls during test project development and testing.

3.2.3 Execute

The Execute stage starts with a test work session, which requires that a testing time slot be reserved by the client. During the session, several test executions are conducted as per the performance testing specifications document. During test executions, both client's and Nu Echo's operations teams collect relevant statistics.

The NuBot platform automatically collects detailed information and statistics from the generated calls, as well as call recordings. These statistics provide a detailed portrait of the system's performance, as would be perceived by callers. Depending on the test specifications, the Nu Echo test expert may also perform manual test calls while automated tests are in progress. This can be useful in order to capture qualitative data in addition to the quantitative metrics automatically produced by the platform. In this case, the expert logs all relevant observations in a computerized logbook for further analysis.

The client team is responsible for gathering Key Performance Indicators (KPI) regarding the tested infrastructure, including network, CPU, and memory usage along with response time and error logs for various architectural components. The client is also responsible for generating statistics over time in order to map system behavior with potential defects.

Test execution is performed in a collaborative mode where all involved teams meet at distance via Conference Bridge and/or a chat room for live coordination. Following each test work session, a results summary is produced by the Nu Echo team and the project manager sends a project status update in terms of NuBot platform usage and human resource costs.

3.2.4 Analyze

The Analyze stage occurs after each Execute stage. Two different activities occur at this stage following a test work session.

Activity 1 the Nu Echo project manager will contact the client if the project is expected to exceed budget. Budgets are most often exceeded when unanticipated issues arise during test executions or if issues are more difficult to resolve than initially anticipated. This often implies the addition of extra iterations (i.e., additional occurrences of Execute and Analyze stages). Depending on the client's decision, the project can exceptionally move to the Report phase even if the test objectives have not all been met.

Activity 2 the Nu Echo team and the client meet together to share the information gathered during the various test executions. During this meeting, results and analysis are discussed and three outcomes to the discussion are then possible:

- All test objectives (as per the test specifications) are considered met and the project moves to the Report phase.
- It is decided to return to the Execute stage as initially planned in the performance testing specifications.
- Considering the results from the previous test executions, corrective actions are to be taken by the client before another Execute stage is started and/or the tests to be executed in the next Execute stage, specified in the performance testing specifications, need to be revised.

3.2.5 Report

At the Report stage, a final report is produced, including an executive summary, a review of the project's objectives and how they were met, known remaining issues, technical highlights, and an aggregation of summary results produced after each Execute and Analyze iteration.

4 Common Testing Patterns

Several testing patterns can be used during performance testing depending on the client infrastructure characteristics and client testing objectives. The following subsections describe a few testing patterns that can be used in a test. Understanding these patterns is useful when reading the performance testing specifications document produced for each project.

4.1 Single Scenario / Single Channel Testing

This is a simple pattern where a single test call on a single telephony channel is generated. This is usually done when developing and debugging the NuBot scenarios, or at the beginning of a working session to verify that, a single call can go through prior to launching a higher density test generating a higher number of calls.

4.2 Baseline Testing

In most projects, there is a need to produce baseline results to which other tests will be compared. As a rule, we normally produce these baseline results by testing the system at 20% of its target capacity.

4.3 Progressive Ramp Testing

One common and important objective to be achieved in high capacity performance testing is to validate that the client infrastructure can sustain a targeted number of simultaneous inbound calls. In general, best practices require testing to start with a low call density and to plan several steps of increased density rather than immediately testing at the full target capacity. We recommend planning a sequence of test executions where the call density is gradually increased from one test execution to the next. The reason for this approach is that if an issue is found at low capacity testing, there is the potential to resolve the issue without the risk of: 1) wasting NuBot platform testing minutes and 2) the issue will more easily be found in low load testing. A high number of calls presents the potential for a high number of errors, making it more difficult to track and diagnose causes for failure.

4.4 Maximum Capacity Measurement Testing

This pattern is used to determine the maximum capacity in terms of call density that an infrastructure can handle. In this case, the progressive ramp testing method is used, but this time the call density is gradually increased up to a point where an unacceptable amount of failures is monitored.

4.5 Stress Testing

Using this pattern, specific boundary conditions related to the client infrastructure are tested while the infrastructure is under heavy load. Examples of boundary conditions may include: a very long call length, a high CPS (calls per second) rate, a high number of agent calls being recorded, calling only speech recognition applications, etc.

4.6 Marathon Testing

Occasionally, it is necessary to assess whether an infrastructure will sustain traffic during a long period of time without failures. The marathon (or soak) testing pattern normally lasts several hours or a few days with a low to medium call density.

4.7 Service Continuity Testing

An infrastructure often needs to be certified for survivability, which means that it is able to continue providing service even after certain components have failed. During continuity testing a medium to high call density test is achieved over a period of time during which various component failures are simulated. Depending on the requirements, it may or may not be acceptable that some calls related to the component failure be dropped. In most cases, calls initiated after the component failure should proceed without error. This test allows the client to validate that the service continuity requirements pertaining to his infrastructure are met.

4.8 Multiple Scenarios Testing

When an infrastructure is tested to validate its maximum call handling capacity, it might be required to generate a call pattern that corresponds to a typical distribution of calls amongst a set of use cases; each use case being associated with a service that the infrastructure is offering. In this test pattern, several call scenarios are involved, each bearing its own weight in the total number of calls generated by the NuBot automated testing platform. It is wise to initially use a single scenario for capacity testing because it usually makes errors easier to diagnose. Once each single scenario works at medium to high density, a larger test with multiple scenarios can be considered.

4.9 Multi-Site Testing

Infrastructures that are highly redundant and robust to failures are normally distributed between two or more sites. Specific telephony configurations are often required for these tests as phone numbers for reaching a service at a specific site will often need to be provisioned only for testing purposes. These tests must be carefully planned when systems are already in production: when all sites are exercised at once, regular end- customer calls might not get through during the testing period.

> **…moderate to severe problems are always found in performance testing projects even when clients are confident in their infrastructure's performance.**

5 Common Causes for Diagnosed Problems

Various problems can be discovered during performance testing, including poor voice quality (low MOS), unexpected busy signals, unanswered and/or dropped calls, dead air, delays in audio prompt playback, and unrecognized DTMF input. Based on Nu Echo's experience, moderate to severe problems are always found in performance testing projects even when clients are confident in their infrastructure's performance. When problems are diagnosed, their causes are often unexpected. Although the nature and source of problems are different from one client infrastructure to another, there exist commonly diagnosed causes. The following subsections categorize these causes into three main classes.

5.1 Customer Local Carrier Issues

These causes are related to the client's local carrier. They are sometimes hard to fix because the client must have enough information on hand to convince the carrier that the source of the problem is on the carrier side. Typical examples include:

- Telephony related parameters not properly configured (maximum number of calls allowed on a DID, toll-free number configuration, DTMF encoding method, etc.)
- Bad network connections (unstable T1 connection, high latency SIP connection with lag/jitter, etc.)
- Capacity problems (not enough resources or automatic limitation for high CPS rate, etc.)
- Security issues (DoS safeguard).

5.2 Infrastructure Hardware and Software Issues

When clients have full control on their infrastructure, these infrastructure related causes can often be fixed quickly except when the cause is related to a malfunction of a third party equipment or software. Common causes include:

- Misconfigured hardware (switch, router, load balancer, SBC, etc.)
- Misconfigured software (bad redundancy settings, bad OS configuration, incorrect software parameters, or other)
- Misconfigured technical services (DNS, Domain Controller, etc.)
- Insufficient number of software licenses to support the tested capacity (number of contact center agents, number of concurrent speech recognition sessions, etc.)
- Faulty equipment (T1 card, media gateway, etc.)
- Over-rated hardware or software specifications (equipment or software does not sustain planned capacity) in physical or virtual environments
- Unreliable network connections or lower network bandwidth than specified in the architectural documentation (local network bandwidth, cross-site network latencies, media packets jitter, etc.)
- Miscellaneous inefficient services (load balancing, back-end services, speech recognition, etc.)

5.3 Architectural Issues

These issues may introduce important delays in the commissioning of an infrastructure depending on the severity of the architectural problem. In some cases, architectural design problems may require major changes to the infrastructure in order to correct them. Examples include:

Required computing resources significantly higher than expected (not enough memory, saturated CPUs, etc.)

- Inadequate QoS (Quality of Service) design
- Security constraints not correctly taken into account.
- Any other architectural design problem.

6 Terminology

Stage: One of the five elements of the performance testing methodology pipeline.

Iteration: An occurrence of the "Execute" and "Analyze" stages.

Scenario: A detailed test description, authored in the NuBot Platform that describes the expected interaction of a simulated caller with a called service.

Test descriptor: A NuBot project test descriptor which refers to one or several scenarios and which defines several test parameters such as the call rate, called numbers, number of simultaneous calls, scenario weights, etc.

Test: A call pattern to be generated during a predefined amount of time, which is associated with a business objective in the test specifications. In the NuBot platform, the call pattern is defined using a test descriptor.

Test execution: The uniquely identified execution of a test as defined in the "Tests Specifications" document. In the NuBot platform, a test execution corresponds to the launch of a test descriptor, which triggers the call generation process and the start of results collection

Test working session: A working session, often scheduled during the night, to perform one or several test executions. In turnkey projects, a member of the Nu Echo team operates and monitors the NuBot platform during the working session.

Phase: A complete project in and of itself going through all the five stages. Because of the size, complexity, or duration of the testing project, it may be required to brake it into multiple phases.

Performance Testing Methodology

Version 1.3 2015-10-05

Nu Echo Inc.

1435, Saint-Alexandre, bureau 200, Montréal (Qc), Canada, H3A 2G4
Tél: (514) 861-3246 | Fax: (514) 861-1676

www.nuecho.com | blog.nuecho.com