Excellence through Quality Assurance

- Karthik

"Prevention is better than cure"

This maxim perfectly explains the difference between **quality *assurance*** and **quality *control***. Quality Assurance is a set of processes that needs to be followed throughout the SDLC process, not just during testing. These processes help in preventing defects from occurring throughout the lifecycle, whereas quality control is simply a validation activity which helps find defects.

At CGN Global, we understand the importance of true quality assurance and we are committed to deliver quality products – *every time*.

This document explains the Software Quality Assurance and Testing processes followed at CGN Global that help in delivering high quality software products with a lower budget for Fortune 500 companies.

> CGN Global insists that Quality Assurance must saturate the entire project and the processes to assure quality must be observed throughout each phase.

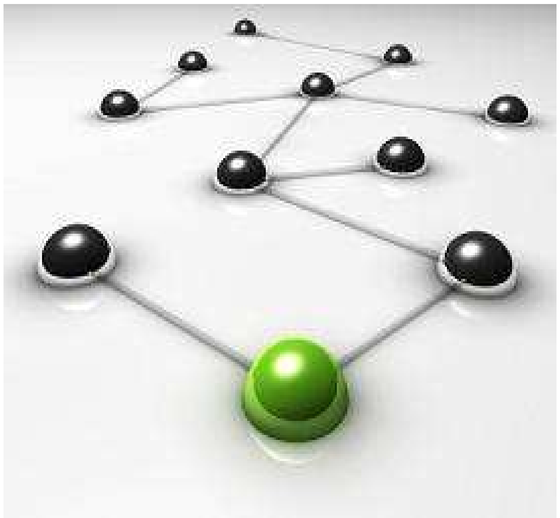
State of the art ALM

ALM (Application Lifecycle Management) is a suite of tools that help with entire software development life cycle starting from requirements management through delivery. At CGN Global, we understand the importance of picking the right strategy and tools that play a vital role in the success of any project.

In order to perform Quality Assurance and achieve quality delivery, CGN Global uses state of the art techniques and industry best practices, through an ALM suite of tools. There are various industry standard ALM tools available in the market. They are HP ALM, Visual Studio ALM, etc. CGN Global has extensive experience using these tools and has implemented them for projects at different client places to achieve effectiveness and efficiency, resulting in quality delivery. We also train the client's staff on these tools, so that it is sustainable for them in the long run.

Requirements Traceability is key

One of the main reasons many software projects don't yield the desired results, is the lack of *requirement traceability*. At CGN Global, we thoroughly understand the importance of *requirements traceability*.



There are different types of traceability that need to be maintained throughout the project to make sure that a quality product is delivered at the end of every project. They are:

- Business vs. System requirements traceability
- System requirements vs. Code traceability
- System requirements vs. Test Cases traceability

Requirements traceability is no longer just a QA or testing task. It needs to be maintained throughout the life cycle of the project, especially during development processes in order to prevent defects from occurring and identify any gaps that may exist.

Business vs. System Requirements Traceability

Requirements can be in many forms – Business requirements, System requirements, Functional requirements, etc. At CGN Global we understand the significance of different requirement types and trace it accordingly to make sure it meets the client organization's strategic goals.

As a first step, our industry experts along with business analysts meet with client SMEs to gather high level business requirements. These business requirements reflect the client organization's strategic goals and are in line with that. Once the business requirements gathering is over, the CGN Global BA team details out the system requirements and these requirements reflect the application or system needs. Once this exercise is complete, we trace all the system requirements back to the business requirements to make sure we are still in line with client organization's strategic goals which is the ultimate objective of the project. This traceability will also ensure that all the business requirements have been converted in the form of system requirements which will be the base for development and testing.

In order to achieve full traceability, system requirements are uploaded into ALM. These are simply the detailed system requirements that need to be fulfilled for the project and they are easily accessible by the entire team in the ALM tool. When any requirement change occurs, it is reflected here and available to all members.

> Requirements Traceability is more important than you think.

Development Traceability – System requirements vs. Code

Once the requirements are uploaded, development team creates tasks for each and every requirement in the tool. These tasks are the detailed activities that need to be performed by the development team in order to successfully develop/code a specific requirement by considering different permutations and combinations of scenarios. Each task then gets assigned to project team members for development. This activity makes sure that each and every requirement is traced back to code, preventing defects from occurring and ensuring a quality code delivery to QA team for testing.

Testing Traceability – System requirements vs. Test cases

The testing team uses a test management tool (Test Manager - part of ALM suite) for test cases creation. They initially upload the detailed system requirements into the tool via a query process and then create test cases for each requirement. This allows the testing team clear traceability of the number of test cases written for each requirement and helps to ensure complete test coverage. Requirements Traceability Matrix is also created to ensure test cases creation for all different permutations and combination of requirements. And then requirements are checked off as completed during creation and execution and get tied to direct functionalities.

The Process in a nutshell

As the development sprints or modules are completed, the development team updates the ALM tool, identifying and associating specific change sets developed to specific requirements. This process enables the project management and the development team to know which change sets need to be included to create the build that is deployed to the QA environment for testing and which requirements are being addressed.

With a well-defined build delivered to QA, test case execution and defect tracking are easy to monitor. Status is available 24/7 with a single query run in ALM tool.

Software testing in detail

Anyone experienced with software development understands that "perfect" software does not exist, nor is it possible to test every possible condition; however, there are key components to a successful software testing phase:

- Thorough understanding of business requirements
- Careful test planning and accurate assessment of risk
- Complete execution of the test cases
- Disciplined management of defects
- Use of automation to drive efficiency
- Development of sound, specific exit criteria to ensure completion within the timeline.

System test preparation

Once the requirements understanding sessions are successfully complete, the QA Lead creates a robust system test plan and strategy in place which guides the QA team in system test cases preparation. QA team produces system test cases for all different permutation and combination of requirements. Once complete, a review process takes place by both peer and lead. This assures that the test cases are of good quality and have complete test coverage.

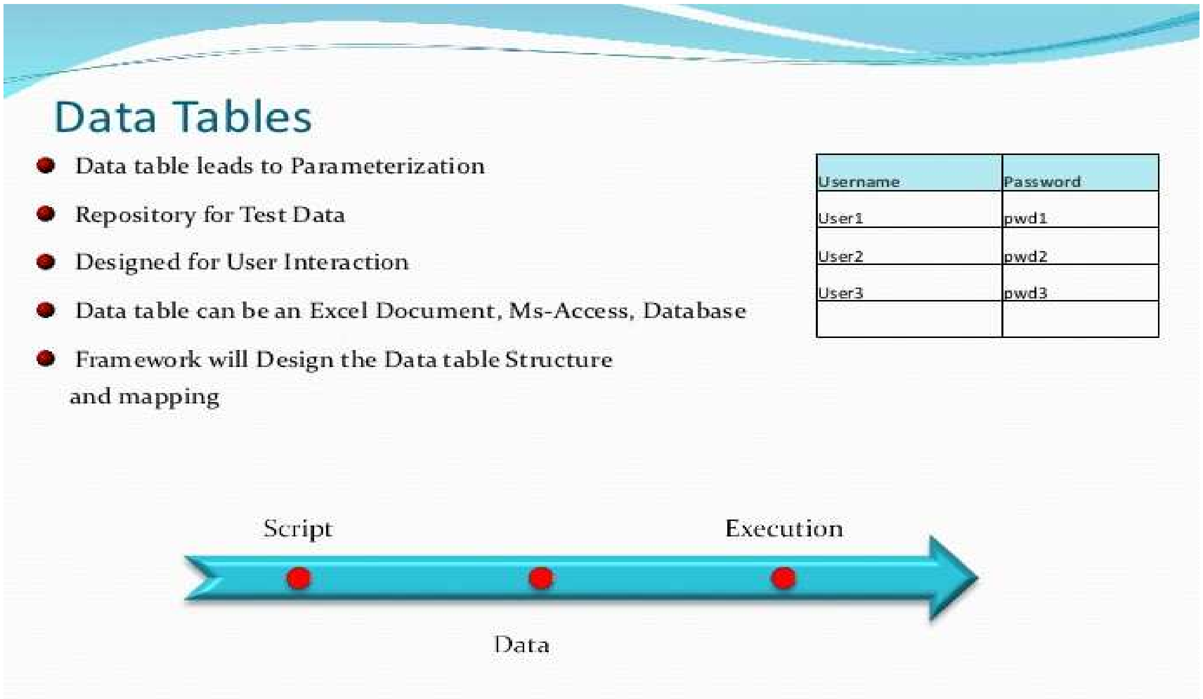
Requirements traceability is built into the ALM tool and test coverage is easily confirmed by a simple query to assure each detailed requirement is fully tested. Any testing gaps are immediately apparent and can be quickly addressed.

A combination of manual and automated test case types will be created for a variety of uses during this phase:

- Smoke test cases
- Functional test cases
- Integration test cases
- Usability test cases
- Database and Data Migration test cases
- Security test cases
- Accessibility test cases
- Browser and OS compatibility test cases
- Regression test cases
- End to End test cases

System test execution

Three rounds of system testing are usually executed. Once the build is delivered to QA, smoke testing takes place to make sure that the build delivered to QA for system testing is stable and of good quality, for further testing. The smoke test serves as build acceptance criteria. Functional and integration testing take place after the successful completion of smoke testing. Test execution consists of completing all the test case types mentioned in the preparation phase. Defects will be logged and managed using robust defect management processes. Defects Tracking Tool plays a vital role during system test execution and act as a central repository of all the defects. Defects raised will be assigned to respective developers for fixes and then get verified and closed by the QA team. Defects life cycle is used to make sure the team follows the correct statuses, type, etc. and this will also speed up the defects fixing process and get us early fixes.



Data Tables

- Data table leads to Parameterization
- Repository for Test Data
- Designed for User Interaction
- Data table can be an Excel Document, Ms-Access, Database
- Framework will Design the Data table Structure and mapping

Username	Password
User1	pwd1
User2	pwd2
User3	pwd3

Script
Execution
Data

We also make very good use of automation testing during the regression testing phase using *Test Automation Tool – Data Driven Framework*. At CGN Global, we understand that test automation can play a vital role in delivering high quality software applications, especially when they are data extensive. We understand the importance of effectiveness and efficiency hence test automation is used wherever and whenever required. We have great experience in industry standard test automation tools like HP QTP, MS Visual Studio Coded UI, etc.

Test Automation: *Data Driven Testing Framework*

Data driven testing framework is often used to test applications with data extensive workflows and functionalities. We create automated test scripts, parameterize it to read data from an Excel spreadsheet and pass a variety of data sets as inputs to the application under test. We add check points (assertions) to the script that capture the output (actual value) during the test run and then validate against the expected values entered in the data table. In this way, one automated test scenario can be used to test any number of scenarios with different sets of data and a great deal of manual testing effort and human errors are avoided.

Use of Test Automation to drive effectiveness and efficiency

Test data table creation is a one-time activity and this data table will be used to run any number of tests during the regression testing cycle without manual effort. Each time the application undergoes defect fixes and enhancements, the automated regression suite is run to confirm all existing functionalities remain unaffected by the code change.

Leveraging an automated test suite for data migration testing, provides definite advantages: we can include different sets of migrated data in the data table and test without investing additional hours.

Performance Testing

After the successful functional testing, we run the system through different sets of load to make sure that the newly developed system is capable of handling various loads when it goes live. We test the newly developed application's performance to make sure that it is in line with system's performance requirements.

We use performance testing tools to report the performance of the system. Industry standard performance testing tools like *Load*

Runner or *Visual Studio* web performance & load test tools are used at CGN Global for this purpose.

As a first step, we create a web test script that touches the major workflows of the system. Then we add any number of virtual users based on performance requirements which will act as load (Load Testing). When we run the load test script, the tool gives you key indicator reports like throughput, response time, CPU usage, errors, etc. using which we pick the areas where performance needs to be increased. Optimization fixes are then made to address any performance issues. We also increase the load to maximum to see at what point the system crashes (Stress testing). We can now document the data point as a system performance threshold, which should be in line with system's performance needs when it goes live.

After the successful completion of system testing, build is delivered to UAT (User Acceptance Test).

User Acceptance Testing

During this phase, much importance is given to business requirements rather than system requirements. We along with the client or end users make sure that the newly developed application meets business requirements and the project deliverables are in line with client organization's strategic goals.

There are typically two rounds of UAT and it begins directly after system testing completion. CGN Global is committed to deliver the highest quality site possible to UAT in order to assure there are no major defects found by the client team during UAT. Whatever defects are raised in UAT Round 1 will be fixed, verified and regression tested by the CGN Global QA team. The UAT Round 2 build, will be delivered to the client team for testing. The client team is usually expected to close all the defects found in UAT Round 1, during Round 2, and no new defects should be found during their regression testing.

Go-Live and Post-Production Maintenance

After the successful completion of UAT, a Go/No-Go meeting is conducted involving all the stakeholders; a decision to push the application to production will be made.

During the post-production phase, the automated regression suite, created by the CGN Global QA team, will be of great significance and reveal its full cost effectiveness. The automated regression test suite may be run as many times as necessary, throughout the lifetime of the application, whenever there are defect fixes applied or enhancements made, in order to assure existing functionalities remain intact. Executing the automated regression test suite has proven to be very effective and efficient for the client team.

Conformance to requirements is done and done right the very first time. CGN Global helps in delivering high quality software products within a much smaller budget.