

GS Lab's approach to quality is tuned to working predictably with fast moving start-ups. A start-up customer profile needs processes more agile than those normally governed by SEI CMM or ISO standards. As such, GS Lab's processes and best practices outlined place a significant emphasis on requirements control and quality of the basic output of engineers rather than checks and balances in procedures. It has strengthened its daily build process, iterative development process, and test automation process to the extent that the required quality of the development output is ensured as increasingly complex requirements are met in a controlled fashion. In addition to these processes, GS Lab lays significant emphasis on review of estimates, designs, and work products. This ensures that mistakes are caught early in the cycle.

GS Lab's Process Philosophy

GS Lab has well-defined processes and practices in place so that clients have control over their projects. The transparency maintained by the GS Lab Technical Managers ensures that the customers know status of individual projects with the required precision. GS Lab provides a basic structure for deliveries along the following four aspects of a development process:

- Requirements Control Process
- Build Process
- Development Process
- QA Process

GS Lab's documentation standards are flexible and are governed by the standards of each customer. A template of sample documents such as requirements document, design document, test plan, and test sheet can be supplied at an appropriate time.

Requirement Control Process

A critical part of working with fast moving remote development organizations is to convert telephonic/email conversations into precise development tasks for engineers. GS Lab ensures that this happens by taking the following steps:

- When a new requirement emerges, the Executive, Technical Manager, and Architect (who together constitute GS Lab's team Management and are all required to be hands-on) contribute to clarifying the feature and help the understanding for assigned engineer immediately after the requirement is first conveyed

- The engineer is required to confirm the understanding with a design document/note. The domain understanding and past experience enable follow-on local decision-making to minimize the communication delays related to approvals and clarifications
- The Technical Manager then follows-up this new requirement to closure through multiple builds and releases, each time making the feature and the related test automation and test plan/coverage more complete
- GS Lab has developed and deployed mature internal web-based tool called GS Collab, which is made available to our customers, for tracking all items of work, their estimate breakdowns, and completion status. This tool forms the foundation for ensuring that local plans are in sync with the customer's plans

Build Process

An automated build process is an integral part of the product development discipline at GS Lab. GS Lab ensures that this process is started at the earliest possible opportunity and follows the process below.

- GS Lab and customer maintain a master source repository with our client and a mirror of that at GS Lab
- GS Lab mirror gets periodically synchronized with the master so that all the data is locally available for development and QA team
- Checkouts are done locally, whereas check-ins are made into the master repository
- Not only are the source files but all documents, schedules, libraries, toolkits are also maintained in the repository

GS Lab follows the practice of making daily builds. The key points about the daily builds are:

- Automatic: The build process is automated and carried out on daily basis at a fixed time using a scheduler (or even more often).
- One-step: It takes just one step to make a shipping build from the source snapshot. There is a single script you can run that does a full checkout from scratch, rebuilds every line of code, makes the executables and libraries, in all their various versions, languages, and #ifdef combinations, creates the installation package, and creates the final media and CDROM layout.
- Sanity tested: The CDROM is put to a sanity test-bed which guarantees minimal functionality on success of tests. The report is automatically sent to the development and QA team.

The advantages of this process are as follows.

- When a bug is fixed, test engineers get a new version quickly and can retest to see if the bug was really fixed.
- Developers feel more secure that a change they made is not going to break any of the "multiple versions" of the system that get produced.
- Outside groups like marketing, beta customer sites, and so forth, which need to use the early product, can pick up a build that is known to be fairly stable and keep using it for a while.
- By maintaining an archive of all daily builds, if strange, new defective behavior is discovered suddenly, it is possible to generate a clear idea what is causing it by using a binary search on the historical archive to pinpoint when the bug first appeared in the code. Combined with good source control, GS Lab team is able to track down which check-in caused the problem.
- When the team is working in two time zones, adjusted schedule of the daily build ensures that the people in one time zone do not create difficulties for the people in the other time zone.

The above process ensures a check-in discipline, constant sync's, and early detection of problems. **A mature daily build process is a significant component of the GS Lab engineering process.**

Development Process

GS Lab's software development process has a problem-solving approach which typically follows the steps below:

- Determine the problem to be solved
- Understand the problem to be solved
- Develop a design and plan for solving the problem
- Execute the plan
- Assess the solution

GS Lab development process includes 8 stages which span these 5 steps. Figure below shows the process flow. A brief description of these steps is given below.

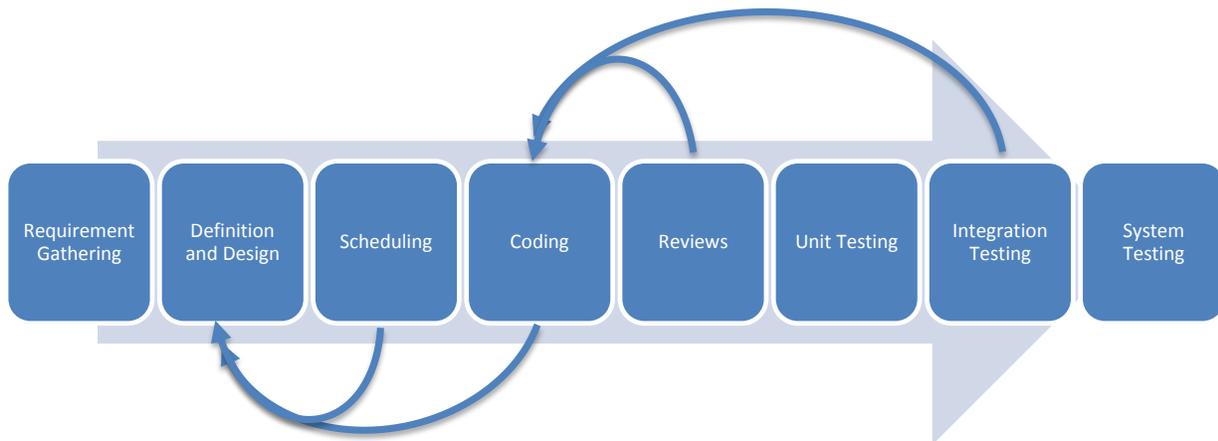
- **Requirement Scoping:** This captures the problem the development team solves. It defines the scope of the problem and standards to which the final implementation is held accountable.

- **Definition and Design:** This stage technically defines the problem, designs a solutions for it, and breaks the solution development into subtasks each achievable between a short period of time. These tasks include the effort required to write automated tests for the features. (The above two steps are carried out with frequent interaction with the client)
- **Scheduling:** Each of the sub-tasks is estimated and by analyzing the dependencies among the subtasks and development resources a schedule for implementation and test is created. The schedule is conveyed to the client and is adjusted in accord to the feedback.
- **Coding:** The implementation is carried out in this phase.
- **Review:** After code-complete stage, code review is carried out and the code is checked in by incorporating the review comments. This stage also involves memory-leak and performance testing with available tools.
- **Unit Testing:** The developer writes unit-tests for his code and the code is tested against them.
- **Integration Testing:** QA team writes integration test cases and the feature is tested against those. We place a strong emphasis on automation of these tests.
- **System Testing:** Regression test cases for the feature are added into the framework and it is ensured that the feature when assembled with the system meets all the requirements.

For all the features in development, the matrix of feature against the phase with respect to the schedule timeline is maintained and exchanged with the client frequently. The GS Collab tool is used to track all the stages.

Figure 1 highlights the iterative nature of these phases. Typically, in early stages of a feature, it gets revisited with some changes after the release cycle is complete. GS Lab manages this situation by dedicated feature ownership within the team and maintaining the duplicate knowledge of the feature development issues with the reviewer. For all the features in development, the matrix of feature against implementation stage is maintained and exchanged with the client frequently.

GS Lab specializes in working with fast-moving start-up companies who have a tremendous time-to-market pressure. **An incremental development process with constant feedback loops is absolutely essential.** Our development process ensures that basic aspects of quality are retained while making rapid progress possible.



GS Lab's software development process

Quality Assurance Process

GS Lab projects emphasize maximum possible automation during QA activities. Each project produces a test framework that covers

- Automated build sanity testing
- Automated regression testing comprising of
 - Unit tests
 - System tests
 - User Interface tests
 - Load tests
- Tools to
 - Create test data
 - Traverse the state machine of the system and bring it to any desired state
 - Capture, store and replay system trace

The QA team develops the test plan, initial tests and test framework enhancements in parallel with the implementation. Regression is carried over all the builds for every version of product. Completely automated regression ensures that no repetitive action is involved apart from GUI tests which have to be done manually during the early stages of any product.

Emphasis on **automation**, **daily regressions**, and **continuously improving code coverage** are a critical component of GS Lab's QA process. The defect management process is run so that there is a constantly growing regression test suite produced from a growing number of test cases.

GS Lab has standardized a tool called Test Link for managing the QA activities and test cases for any deliverable. There are clear guidelines defined for developer to QA hand-offs.

QA Project Management

GS Lab's QA project management is anchored around the Technical Manager, the QA Lead (preferably a test engineer with QA process background), and a Test Architect (preferably an engineer with a development background).

The Technical Manager schedules the test runs and QA cycles, the QA lead supervises the team of test engineers, and the test architect ensures that the tests are run efficiently and the coverage is broad as well as deep. The test architect emphasizes automation and replication of the realistic test environments.

In many early stage product releases, there is a period of manual testing before automation is realized. The QA team is managed such that boredom does not set in due to repetitive nature of the job. The QA lead and Test Architect work together to ensure that every test engineer gets new assignments and is trained to go beyond simply finding and logging the defects.

Defect Management Process

Every project maintains its own defect database. Life cycle of a defect in GS Lab is as shown in figure below. Once the source repository is frozen for new check-ins, the check-ins are blocked by associating them with bug-ids and maintaining approved bug-id-list. This ensures that no developer can check-in his bug fix without approval.

GS Lab's mature defect management process ensures that there are no regressions during the critical phases of a release cycle, communication between teams is efficient, and the release happens with a known quality.

Communication Quality

In outsourced product development, there are a number of occasions when personnel in the two teams depend upon each other to achieve progress. The most important ingredient required for the success of such distributed teams is communication. GS Lab spares no effort in ensuring highest quality communications. Some salient points about how GS Lab maintains communication quality are as noted below.

- Two scheduled calls weekly with entire team with status exchange
- Separate scheduled QA/support calls once a week
- Separate weekly calls between subgroups during design phase for architecture and design reviews
- Daily calls during release time
- Dedicated point persons with check-in responsibility within India team
- Dedicated point persons with communication responsibility within India team
- Daily calls between peers
- Daily exchange of build and regression results
- Multiple broadcast channels for ensuring everyone is in sync

