

Bering Labs Engineering Process

Companies tend to run their businesses at a faster pace these days and we need processes to be more agile than those normally governed by SEI CMM or ISO standards. Bering Labs follows a suite of time tested and proven methodologies and Agile, Continuous Delivery and DevOps practices. Our software engineering approach is to optimize software processes by removing bottlenecks, avoid wasteful activities and completely automate repetitive jobs and tasks.

We put significant emphasis on review of requirements, designs, code and final product. This ensures that mistakes are caught early in the cycle.

Beringi

Bering Labs has well-defined processes and practices in place so that clients have control

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

Bering 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 development organizations is to convert requirements, telephonic/email conversations into precise development tasks for engineers. We ensure that this happens by taking the following steps:

- When a new requirement emerges, the Client Partner, Business Analyst, and Architect (who together constitute Bering 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

Bering Labs Engineering Process

- 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 Business Analyst and Offshore Delivery 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
- Bering Labs uses web-based dashboard, 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

Automated Build Process

An automated build process is an integral part of the product development discipline at Bering Labs. We build the software regularly, updating tiny changes as they happen so everything is up to date when a big change comes along. We ensure that this process is started at the earliest possible opportunity and follows the process steps below.

- Bering Labs and customer maintain a master source repository with the version control system's client. (mirror of that at Bering Labs)
- Bering Labs 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.

Bering Labs 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.

Bering Labs Engineering Process

The advantages of this process are as follows.

- Produce a full, clean build each time when our team publishes new features
- 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.
- Take less time when publishing new features and Provide a history of which change actually broke the software when it fails
- 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, Bering Labs 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 Bering Labs engineering process.**

Automated Continuous Integration(CI)

With this practice, At Bering Labs, members of a team integrate their work frequently, even multiple times per day. This means there's a system that automatically integrates a new feature or bug fix into the product. Before it reaches the general public, each integration is verified by an automated build (including testing) to detect errors as quickly as possible.

This approach incorporates the automated build process mentioned above, but is not limited to it. CI is extended to include running automated tests, all the way through to generating a build, installing the product, and running full tests on it.

Practicing CI allows us to significantly reduce integration problems and have following benefits

- Develop cohesive software more rapidly
- Highlight as early as possible when code changes break the system
- Ensure that all aspects of your development process are lined up to permit the daily creation of a working version of your product.
- Detect who caused the break in the system (this psychological effect is very effective at encouraging our developers to do testing before pushing code).

Bering Labs Engineering Process

Development Process

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

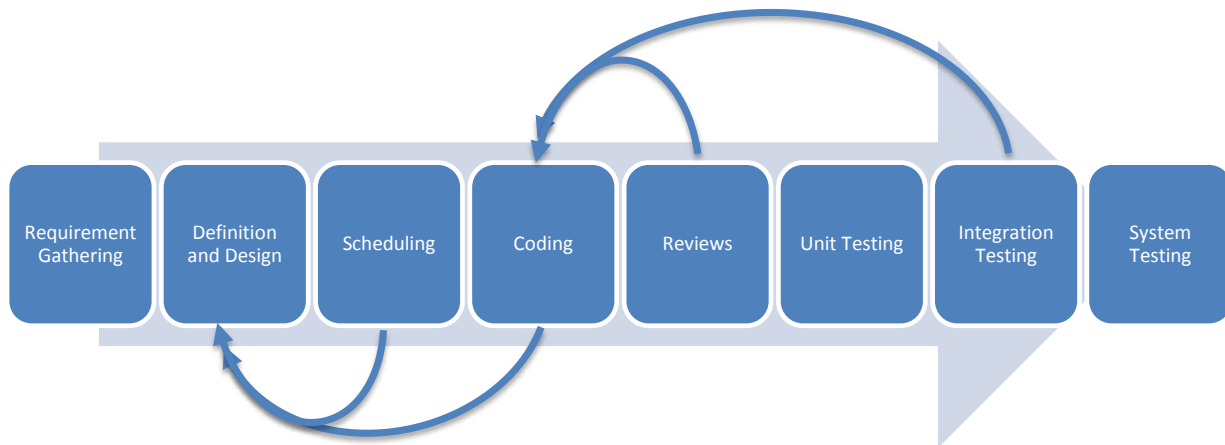
- Determine the problem to be solved
- Understand the problem to be solved
- Develop a design and plan for solving the problem
- Execute the plan
- Assess the solution

Bering Labs 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.

Figure 1 highlights the iterative nature of these phases . At Bering Labs, Agile development takes place in short iterations, in which releases occur in short periods of time. If our customers have a build, say, every 10 days, we close the communication gap earlier on. The client gets a release and a demo to play around with and can provide feedback on course corrections that need to be made early on in the cycle. Clients don't wait weeks and months to find out that a software delivery was off course all along. The client looks at the software earlier on and provides feedback on whether we are on the right track or not

Bering Labs Engineering Process



Bering Lab's software development process

Quality Assurance Process

Bering Labs 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 Bering 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.

Bering Labs Engineering Process

Bering Labs QA practice has experience in almost all types of testing – manual, automated, back-end, REST, API, white-box, black-box, functional, regression, performance and load testing. We have experience with commercial and open source automation tools such as HP Quality Center, HP QTP, Silk Test, IBM Rational Robot, Loadrunner, Watir, CubicTest, Maveryx, Apache JMeter and many more.

QA Project Management

Bering 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 Bering 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.

Bering 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. We spare no effort in ensuring highest quality communications. Some salient points about how Bering Lab maintains communication quality are as noted below.

Bering Labs Engineering Process

- Onsite client partner and business analysts co-ordinate project development activities with client and offshore teams.
- 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

