

NEWLANE

Optimizing the Software Development Lifecycle

Refining an SDLC Measurement Program

Success Story

Our client is a leading online brokerage.

Client Vision

Streamline the company's Software Quality Assurance Testing organization for improved efficiency and throughput.

Challenge

A comprehensive Measurement Program was needed to provide the data necessary to fix the problems in the QA process.

Solution

NewLane provided a short-term engagement that took the client through a six step process to define required metrics, gather the relevant identified data, analyze the results, and establish corrective actions.

Benefits

The results included an effective measurement program, a clear picture of the issues impacting the QA process, and effective actions to streamline and improve that process.

The benefits were significant:

- Employee morale and customer satisfaction improved dramatically.
- QA Dept. throughput doubled.
- The company saved \$1.5Million in development and delivery costs.

Measurement Program: Online Brokerage

When it comes to implementing a successful measurement program, most companies who have been chartered with this task tend to look for the silver bullet, the golden key, and the quick approach. In our experience, successful implementation takes an approach of incremental, planned steps, making sure that the measurements are incorporated as part of, not separate from, the process. One of the major reasons for failed attempts is to try the "measure everything" approach with the result being nothing more than enormous amounts of data. First, establish what is to be measured, why it's being measured, how to gather the data, and what actions will be taken with the information once it has been collected. This case study outlines an approach that is practical, affordable, and can deliver results in sixty to ninety days.

“NewLane provided us with a short-term engagement that will have long-term benefits.”

Business Problem

An online brokerage firm had the perception that their Software Quality Assurance (QA) – Testing organization was the roadblock to releasing products on time with a high degree of confidence in the product's quality. Some of the issues highlighted were the

What every Manager wants to know - What are the Top Five Measures I should be collecting?

It depends! The Top Five Measures will be identified based on an organization's objectives and pain points. If one of the major objectives for the year is to deliver projects on time / on budget, planned vs. actual cost and effort is needed to be measured. If one of the major objectives is to improve quality, how many defects are being found after release must also be measured. If where to begin is the question, we would suggest the following:

1. Tie the identification of what measures are going to be collected and analyzed to the goals and objectives of the organization. Make what has been collected and used meaningful.

2. Make sure the measures collected relate to different categories. For example, do not only collect cost measures or only quality measures. A balance/mix of measures is needed to tell a story.

3. Identify the data that has already been captured. If Management is all of a sudden interested in measures, the typical reaction is "they're watching, what are they looking for now". Assigning gathering data elements to one individual will only result in skewed data. Implementing a measurement system should be as seamless as possible. Use current tools before identifying new tools, whether it's automated or manual, to collect the data.

4. If data that is not already being captured is needed, make it part of the process, not something separate and apart from a daily routine.

5. Do not collect data and then not use it. After identifying the measures and collecting the data, there is still a large amount of work ahead in making improvements to the process.

high amount of testers, the testing effort took too long and yielded an exceedingly high cost. Testing activities were never completed as planned and the customer found numerous defects. NewLane was engaged to assess the QA organization, determine where the problems were, and make recommendations for improvement.

New products were not often released, but upgrades and enhancements to existing products occurred frequently – sometimes daily. The initial data gathering revealed procedures in place that would be the basis for a new measurement program.

They were:

- A Software Development Life Cycle (SDLC) model was defined
- Business Requirements Reviews were conducted
- A Change Control Board had been established
- Program Management Office (PMO) existed; Project Managers were assigned to each release
- Daily Status Meetings were part of the daily routines

NewLane Approach

Since the SDLC was defined, the **first phase** was to gather some basic information about the current SDLC. The following represents some typical questions asked during that exercise.

- How stable are the requirements? When are requirements frozen? Are they traced through the release?
- How many defects are found in QA?
- Where are defects introduced? Requirements, Design, Code?
- What would be saved if defects were stopped from migrating to QA? To Production?
- How much time is allocated for testing and how much time does it actually take?
- Where are the customers finding the defects so that testing efforts can be re-focused?

The **second phase** was to identify measures and data sources that were already in place. The source of the data, identification of the elements, who "owns" the data, how will it be captured and reported (when, by who, for how long) also needed to be identified. The categories of measures investigated were:

- Customer (defects found by customer)
- Effort (planned vs. actual)
- Quality (defects found, by type, by phase)
- Cost (of rework, per defect)
- Requirements Stability (planned vs. actual)
- Employee (#'s and Skills)

The **third phase** was to begin gathering the data as defined in phase-two. Based upon data from various systems and groups, an Industry

Trend matrix was used to help calculate the Cost per Defect found in

QA. After a number of weeks it was discovered that:

- Fifty-Three percent of product issues that were called into the Help Desk were related to Options, Quotes, Trading, and Balances
- QA had underestimated their level of effort by fifteen percent and the number of test cases run by twenty percent
- Thirty percent of the defects found by QA were due to requirements issues
- Twenty new requirements were added during the first week of testing
- The cost per defect found in QA was \$582

The **fourth phase** consisted of understanding the cause and effect relationship between the measures. A linkage diagram was created so that management would understand the impact of the suggested improvements. For example, if the organization were to make a change in stabilizing the requirements, would it have an impact in cost, quality and/or effort?

In the **fifth phase**, analysis of the data collected was conducted to determine what was the root cause of the results, whether favorable or unfavorable. At this point, depending on the level and amount of data available, trending analysis can be done which will highlight special cause of problems or systemic cause of problems. It was determined that:

- Only ten percent of the Test Cases ran covered the Options, Quotes, Trading, and Balances modules. In addition, these modules were not fully covered in the Regression Tests. The modules with the highest defect rate were directly correlated to the lowest percentage of test case execution.
- Requirements were never frozen and continued to change even after QA had started the testing. The results were that QA never fully tested one release thoroughly. To further complicate the situation, Requirements Reviews were conducted without QA involvement. The QA Testers could not effectively estimate effort or test cases because they never knew what they would be testing!
- Eight new builds were promoted to QA in two weeks. QA Testers were working an average of ten hours overtime a week to account for additional test cases. The root cause was the inadequate information and communication with QA. If the Requirement's defects found in QA were stopped from migrating, a cost avoidance of \$48k to \$60k would have been realized for each two week QA cycle. This would have alleviated overtime and increased the number of test cases executed in the same two week duration.

6. In the same light, don't use data as a hammer. It's a tool, but a tool for improvement, not a tool for punishment.

7. Start out with a vertical view of measures, but plan on expanding it horizontally. The QA Testing department may want to start collecting defect data for the tests they are running and responsible for. The QA Manager can then make improvements to the QA process based on analysis of the results. However, the larger impact will occur when you start measuring the defect migration rate - sampling those measures that cause the defects to migrate into QA in the first place. How many code reviews were done and what types of errors/defects were found? What QA receives has a clear impact on what they will find and what they will deliver.

8. Ensure that the employees providing the data know what it is going to be used for. This breaks down the barrier and begins to build a collaborative relationship.

9. Set achievable improvement goals. Use benchmark data to establish best in class, but if the expectation is to increase productivity by thirty-percent, the data collected will represent the case. Data is easily manipulated and if enough investigation is done, the result will yield how quickly bad data becomes favorable.

10. By completing the above tasks, what hurts has been identified, determined how to get better, and determined how strong you want to be. What is required next is to understand the regiment / investment that will be necessary to get and remain healthy. Use these scorecards as the basis for quarterly and yearly health checks to understand just how healthy the organization has gotten and how much more to improve.

NewLane Solutions

Software Quality Assurance

- Standardized and repeatable processes for the development of software products.

Requirements Management

- Processes and guidelines for developing software requirements and handling related artifacts.

Measurement & Metrics

- Track and manage project progress, allocate resources for optimal ROI and demonstrate project value.

Software Configuration/ Change Management

- A control system for software development projects.

Vendor Management

- Processes and procedures for qualifying, selecting, managing, and evaluating outsource vendors.

Project Management

- Processes, procedures and techniques for the organization and orchestration of all types of software development projects.

Quality Assurance (Testing)

- A total testing solution - QA Framework - that includes process, automation tools, and skills assessments.

Continuous Process Improvement

- Instills stability and predictability into software development processes.

Organizational Process/ Change Management

- Influence organizational behavior to improve the acceptance and practice of process and procedural changes.

NewLane

NewLane Partners
The Heritage Building
150 Coolidge Road
Berlin, MA 01503-1326
978-567-2100

www.newlane.com
info@newlane.com

Solution

The analysis drove the recommendations for change – the **sixth phase** – Taking Action! The recommendations were:

- Increase the number of test cases for the defective modules in the regression test suite
- Re-evaluate the Change Control and Defect Reporting processes
- Requirements Reviews (established as part of the exit criteria for the Requirements Phase of the SDLC) included QA personnel.
- Skills analysis was completed to assure that the QA group possessed the appropriate skills for the job profiles.

Benefits

By collecting these available measures, we began to quantify the existence of other items that eventually lead to a breakdown in communication and processes between QA and Business Analysts. Once we began to analyze the data, we began to tell the story. That story provided the customer with the following measurable benefits:

- Employee moral was increased as a result of an overtime savings and elimination of ready-aim-shoot procedures.
- Customer satisfaction increased along with confidence that each new application functioned as specified.
- Cost savings of \$1.5Million in development and delivery based on the new procedures and communication guidelines implemented.
- Lower turnover as a result of hiring the skills required for each position and team.
- Throughput for Quality Assurance has doubled with the decrease of changes introduced at the time of testing.



NewLane Packaged IP

NewLane Partners is an innovative consulting firm that delivers process improvement to large IT organizations - optimizing the components of the software lifecycle like a software factory - to measurably reduce waste and increase consistency, predictability and quality. NewLane's expert consulting team leverages proven best practices to reduce costs, instill change, streamline processes, and produce quantifiable returns. As a result, NewLane's clients realize improved output, greater business value and measurable savings in technology, resources and capital expenditures.