

## Using predictive mechanism to improve effectiveness of the 'Quality Assurance' team

The 'Quality Assurance' team usually looks for defects in the product/service before it is delivered to the client, but there is hardly any mechanism to check the effectiveness of QA in a proactive manner. Adding another level of QA i.e. QA of QA un-necessarily adds to the cost without matching benefits. This white-paper portrays a mechanism to measure the effectiveness of the 'Quality Assurance' team before the deliverable lands on client's desk without the extra cost of second level of QA tests.

Mangalam measures the productivity & performance of each of its project through relevant quantitative metrics. For example, we measure the productivity of the E-Discovery team by the amount of data that gets processed per person per day, and this is tracked on a continuous basis. This helps us benchmark our performance, and look for avenues to improve it further. We have similar productivity metrics for our document coding team of the 'Litigation Support Group'. This team is measured by the number of documents coded per person per hour.

As part of the operational process, after the coding team completes its task, the Quality Assurance (QA) team reviews the work product, identifies defects and gets it rectified before final delivery to the client.

The number of defects found by the QA team tells us about the performance of the coding team. We capture the data on the type of defects found by the QA team, the number of defects per document and use this data in training and re-training of the coding team. We use the defect data to measure the quality of output by the coding team, and run six sigma/kaizen initiatives to reduce the number of defects of each type on a continuous basis.

QA team measures the quality of work delivered by the coding team. However, how do we measure the effectiveness of the QA team? How do we know that the QA team has found out all the possible defects that were present in the output delivered for quality check? One way to do it would be the feedback from customers. If the customer reports defects or errors in the delivered output, that would go as a defect against the QA team. But this approach has two problems:

- 1) Sometimes, the client would not refer the wrongly coded document for a long time, and in such case the defect is buried into the deliverable without us even knowing about it.
- 2) A bigger problem with this approach is that the information about the defective delivery comes after the output is delivered to the client.

We were looking at pro-active mechanism to detect defects after QA review, but before the delivery to the client.

## Using the historical defect data

To address the above challenges, we came up with a validation mechanism for the QA team. We used a statistical approach so that the QA team can validate whether they have found adequate number of defects in the work under review. If historical data tells that the coding team makes certain number of defects in a particular field being coded, the QA team should have found proportionate number of defects of that particular type in the work product under review.

Let us look at an example to understand it better. For example, historical data tells us that the coding team makes 3 errors in 'document date' for every 100 documents coded. So, if the current work product under review is of 1000 documents, then the QA team should expect around 30 defects of the 'document date' type. Similarly – if the historical data tells us that there are 2 'document type' errors for every 100 documents coded, then in the assignment of 1000 pages under review, the QA team should expect around 20 'document type' errors.

Based on this historical data, the QA team creates an estimate of the number of errors in each type of defect for a particular project under QA. After creating this estimate of expected errors of each type, the QA team would review the deliverable. If the number of defects found by the QA team in this deliverable varies by +/- 5% - the QA review is considered fine. However, if the variation is larger than +/- 5%, the QA team carries out a more rigorous review process to ensure that there are no lurking defects left in the work product & the review is adequate.

We have found this method to be very effective way of improving the quality of reviews by the QA team. There are additional variations to this predictive mechanism to make it more effective without additional efforts. We found that in more than 70% of the re-reviews by QA team based on a higher variance with the estimate, the QA team did find out more defects in the work product. This would not have been possible in the absence of the trigger mechanism for a re-review by QA team.

If you think, your QA team can benefit from similar predictive mechanisms, our team of experts will be happy to help you implement such a system in your project. Send us a note at [sales@mangalaminfotech.net](mailto:sales@mangalaminfotech.net) and somebody from our team will contact you.

## About Mangalam Information Technologies Pvt. Ltd.

Mangalam is a leading offshore 'Litigation Support' Services provider, based in Ahmedabad-India. Our services cover a wide range of litigation support services including Offshore EDD Data Processing, Bibliographic Coding and Records Retrieval related services.

Our clients include some of the largest national litigation support services provider in the US, Canada, Australia and UK since more than 7 years. We are an ISO 27001:2013 certified company

For more details – Contact: [sales@mangalaminfotech.com](mailto:sales@mangalaminfotech.com)