

Test Process Transformation Protects Product Development Investment



STAG transforms the test process to enable effective product assessment and certification of product fitness for beta release, which helps protect the investment in product development for a leading Fleet Management solution provider.



Domain/Category -
Transportation / Fleet Management



Microsoft technologies / JAVA

CUSTOMER AND PRODUCT BACKGROUND

The customer is a leading provider of hybrid terrestrial and satellite technologies for telecommunications that has its headquarters in Singapore, two offices in the US and other parts of the world, and a development center in Bangalore.

The product in question is a mobile asset management solution, the flagship product of the company, which helps track, manage access, and monitor all trucks or other mobile assets. It integrates GPS technology and wireless communications, thereby providing customers with location-relevant and time-sensitive information about their mobile resources. Developed using Microsoft technologies, the product has a substantial installed customer base.

PROBLEM STATEMENT

The customer had invested significantly in radically remodeling the current product and outsourced product development, to be done on the Java platform, to a development partner. The customer had identified certain beta clients for end-user testing. The feedback from the release to the beta clients was important to the customer in terms of future business opportunities. Therefore, the customer was looking for an independent quality assessment of the product's performance and fitness certification prior to its official launch.

SOLUTION

The product was in the last stage of the development cycle when the STAG team started the engagement. The team noted that there were no historical test artifacts available to judge the current maturity of the product, despite the fact that some level of testing was going on. It was able to identify the following gaps:

- There was a lack of a clear test strategy to target the most critical issues.
- The existing test cases lacked sufficient test coverage in terms of the length and breadth of testing.
- There was no clear-cut strategy in place for regression testing.

The STAG team then formulated a test strategy using HBT techniques like Error-Fault-Failure (EFF) and Critical to Quality (CTQ). This helped clearly define the issues to be targeted for uncovering during the test phase and the types of tests to be used for this purpose. The team designed test cases applying HBT techniques like Behavioral Stimuli (BeST), boundary value analysis, equivalence class, and domain specific special value.

The team enhanced the breadth of testing by adding test cases to validate the key non-functional aspects of each module. By applying the Interaction Matrix (IM) HBT technique, the team was able to clearly identify the scope of regression for every change made to the code by way of enhancements or bug fixes.

At the end of every cycle of testing, the STAG team shared a report that clearly rated, on a scale of 1 to 5, each feature with respect to the ratio of test cases executed to those failed, the severity of the issues detected, and the effort required for the workaround, with the customer. The team also conducted a performance evaluation of the system using a mix of commercial and open source tools.

OUTCOME AND VALUE ADDITIONS

The STAG team transformed the work process completely by first breaking the work down to the smallest possible levels. Each task in the process was assigned a schedule and alerts were raised at the slightest deviation from the schedule to ensure the project remained on track.

The performance and fitness report showed high variance in the defect closing vs. test cycle, indicating a highly unstable code structure. The performance tests also found very critical design related issues due to weak architecture.

The frequency with which the closed defects were being reopened during the entire validation was alarming. As a result of fixes, new defects were being injected. Further, earlier passed test cases were also failing in the subsequent tests. The Quality index was found to be non-linear.

The report enabled the customer to make an informed decision to defer the product release for beta client testing and focus on sorting out the product fitness issues first.

-  Percentage of designed test cases accepted by customer - 97
-  # Levels of functional testing: 2
-  # Rounds of module validation: 4
-  # Rounds of role-based testing to validate end-to-end scenarios: 2
-  # Rounds of performance testing: 1