Test Automation for Improving Testing Efficiency
Success Story

Customer & Product

The customer is a global organization in designing, manufacturing, selling and servicing gas monitoring instruments, systems and related products. They are a company of over 800 team members spread across different parts of the world. They offer gas detection as a service and their products excel in monitoring hazardous gases in the most demanding work environments.

Their web application acts as the core product for the complete gas detection solution. The web application is used to configure, monitor, record, analyze, alert and report gas detection devices used in fields. The web application has multiple interfaces i.e. for administrator, support team, and client interfaces.

The industry requirements continue to evolve over the years, resulting in the need to frequently update, retest and release the application. With over 200+ pages, 2300+ test cases and 300+ new/updated requirements per year the testing efforts for the application were increasing dramatically. Our customer started using test automation for improving testing efficiency. The test automation project was handed over to Powersoft19 SQA team.

Challenges

We have faced different challenges in the past four years of automating new scripts, utilizing them in testing and maintaining them.

Following are the major highlights:

›› The framework used by client provided .xml based reporting with limited formatting and structure capabilities for readability.
›› QTP/UFT Framework (original) was provided encapsulation even a user who does not understand the automation tool can execute and analyze the report. After QTP/UFT training client requested to remove that encapsulation therefor requiring scripting and reporting structural updates in framework.
›› The application was not developed while keeping automation in mind; therefore, frequent internal updates were observed that affected scripts but did not affect application functionality.
›› Throughout the release cycles i.e. 2-3 releases per year with 100+ new or updated requirements in each release made scripts maintenance and utilization task difficult.
›› Application's core server interacted with hardware for data gathering which could not be automated using QTP/UFT.
›› QTP/UFT was crashed or stuck randomly while executing more than 1200 scripts continuously.

Solutions

With the help of combined efforts, the Powersoft19 SQA team was able to resolve all the above mentioned problems effectively and timely.

Following is the summary of solutions that were implemented to resolve each problem respectively:

» The original frame provided following features:

  » Encapsulation over QTP so that team members without QTP knowledge can select data, select test case, start execution and analyze reports.
  » Used xml for generating complete execution report. In the xml report all the steps and verification for a script were being reported in a continuous manner.

The requirement was to maintain the encapsulation; however, we suggested improving the reporting structure so that we could group steps and verifications of scripts to map to original test case. The purpose was to have a better test coverage and make report understandable for only the persons who had understanding of the test case. For this, we suggested and modified the reporting structure and its library functions. We created the final summary in Excel. In Excel report we presented summary mapping to test case steps in one column and added detailed steps and verification summary for each step in next column. This also provided a better opportunity of report filtering and analysis.

» The encapsulation of QTP/UFT provided in framework made it easier for the team members who did not understand QTP/UFT. However, this resulted in reduced utilization of built-in features of QTP/UFT and made the updating process difficult. The customer also obtained training of QTP/UFT and therefore became more interested in using QTP/UFT features and QTP/UFT reporting rather than relying on Excel or any external reporting feature. With very close coordination and useful feedback from our client, we were able to completely revamp the framework as per need of the client. However the revamped framework still kept the aspect of script mapping to test case steps and produced summary overview of these steps for better understanding of report by layman.

» The application was not developed with test automation in mind, and therefore we observed frequent and random updates in object properties that were used by QTP/UFT for object identifications. Therefore, we communicated and consulted with clients' design/development teams about the requirements of automation and finalized the object properties that could be used for object identification on customer's web application. We also conducted this meeting with development team frequently to revisit the properties being used and communicated the importance of maintaining the properties as much as possible.

» To keep up with pace of the application updates and to have maximum utilizable scripts at the time of regression, we divided the scripts updates into three cycles:

  » Cycle-1 only focused on fixing the scripts that failed for different reasons but not due to application update. This helped us achieve a better coverage during functional testing of functionalities that were not updated.
  » Cycle-2 focused on newly introduced failure and updated test case, as by that time the new functionalities had matured within the application.
  » Cycle-3 started after the version had been released. The focus of this cycle was to maximize the automated scripts for the next release.

» The external devices communicated with application using web service. We first identified test case that required end to end testing and should not be simulated. For those that could be simulated, we developed functional library and then used it to communicate and simulate hardware based devices. After uploading the simulated data, we then verified the application behavior. We could not use the QTP/UFT web service component as it was not available in the license; therefore, we had to develop our own functional library.

» One of the major problems was that the execution of 1300+ scripts could be interrupted if QTP/UFT got stuck or crashed. For the overnight script executions where we were expecting the execution would complete by next day when we arrive in office, the execution would have been interrupted by the QTP/UFT crash. To handle this we developed a VB script based launcher file. This file runs independent of QTP/UFT and uses automation object model of QTP/UFT to launch the automated scripts and then monitors the QTP/UFT progress.

If it observes that there is no progress but scripts remain to be executed then it resets the environment and re-launches the QTP/UFT.

Achievements

Following are the highlights of things that we were able to achieve during our test automation development life cycle:

» Optimized the framework with following features:

  » Simple to execute
  » Properly utilizing QTP/UFT features
  » Easier to maintain
  » Easier to analyze by using summary report mapped to test case and QTP/UFT detailed report combined together
  » Meets client's expectations

» Increased the application automate-ability with the help of development team and by having a continuous communication channel.

» Revamped the GUI of one version of the application, yet the functionality and the objects were reasonably consistent. This enabled us to have 80% of the scripts ready before the regression run.
» Devised a continuous and smooth script updates process while obtaining maximum utilization of scripts during functional and regression testing.
» Increased the automation coverage by having hardware simulated where required.
» Improved scripts execution because with launcher script we were able to have uninterrupted test runs and the executions were completed in time.

Contact Us

Explore ways to use our expertise in growing your business while establishing a valuable partnership with us.

Contact our consultants at:

Phone: +1.412.533.1700
E-mail: sqa@powersoft19.com
Website: www.powersoft19.com/sqa