Mindtree
A Larsen & Toubro Group Company

A POINT OF VIEW

By Test Advisory Services Team

Continuous Testing for SAP Systems – An approach

Navneeth Kowdle
Director – Test Advisory Services

Introduction – The need for continuous testing for SAP

Many of the world's largest implementers of SAP are continuously looking to improve the effectiveness of an already mature QA organization. As teams and releases become more Agile, the QA function is also expected to adapt to the needs of the program and provide a quick yet reliable turnaround on QA requirements. Therefore the QA strategy should intrinsically include a continuous testing approach that effectively combines processes, tools and resources.

Amongst the various key activities pertaining to QA, efforts towards Regression testing is compromised due to various factors such as paucity of time, non-comprehensive regression suite, inadequate automation etc. Regression testing efficiency can be improved to a great degree using effective impact analysis. In addition Enterprises can significantly reduce test cycle time by using customized continuous testing solutions to achieve effective regression testing with minimal additional cost. This PoV explores one of the approaches that can be taken (with a combination of suggested tools) to enable an effective Continuous Testing strategy for SAP regression testing

Suggested Approach

Enabling automated regression testing with a Test-Ops/Continuous Testing approach can be achieved by implementing a series of steps/phases. Each step should be considered as a pre-cursor to the next step and needless to mention, every phase will have a set of challenges that need to be tackled. This document does not get into the intricacies as such, but provides the reader with a high level insight on the recommended approach

Step 1 - A comprehensive regression Suite and Scenario Mapping

The first step, although seemingly obvious, is to ensure that a core regression suite is built, if not already in place. The regression suite should cover a comprehensive list of scenarios that are core to the functionality of the SAP modules implemented.

Step 2 - Automation of the regression suite

The second activity will be to increase the automation coverage of the regression suite to the maximum extent. A wide range of tools are available in the market to enable Test automation. Some are script based (for ex SAP TAO/CBTA) and some are script-less (for Ex WorkSoft). Based on the need and suitability, the right tool has to be adapted to enable as much automation as possible. For the purpose of this paper, SAP TAO/CBTA is assumed to be the tool in use for automation.

Step 3 - Impact based regression testing

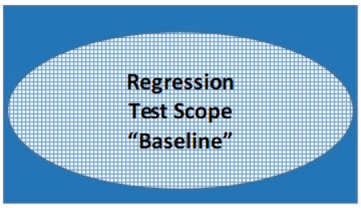


Fig A – Complete Regression Suite

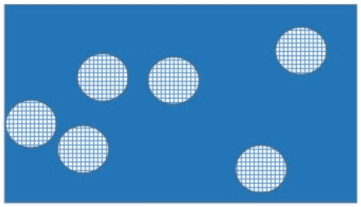


Fig B – Impacted Areas

Typically, when a regression suite is in place, an old-world approach would be to automate the suite to the maximum extent possible and run the entire regression suite (Manual+Automated) in each test cycle (Shown as Fig A above). Although this is time tested, it is also cumbersome, time-consuming and if the automation coverage is less, it is more effort intensive as more resources are needed to complete manual testing. In the world of agile, where time is at a premium, this approach is outdated. A better approach would be to understand the exact changes brought into the system and consequently understanding the impact to existing functionality due to the changes. Once the impact is understood, only the relevant test scenarios can be tested, thus reducing time and effort enormously. This impact based approach is depicted in Fig-B above where the small circles are meant to show impacted areas only instead of the entire baseline

To be able to move to an impact driven approach, all the test scenarios from the regression suite have to be mapped to the requirements/business processes. The business processes should be available in a SAP business blueprint.This activity is important to be able to perform the correct impact analysis when changes are introduced in any of the associated functionality of business processes

Step 4 – Automated impact analysis (BPCA considered in this paper)

Business Process Change Analyzer (BPCA) is a home grown tool provided by SAP that exposes modules impacted due to modifications made through new features or change requests. It helps compare the changes brought in through the transports with the current system and identifies all the impacted areas (Objects). With the right configuration and a mapping between the test management solution and the system, BPCA guides test scoping and reduces time/effort of testing by enabling businesses in identifying scenarios impacted by change. It supports a wide range of areas of SAP modules including ECC, CRM, APO, SCM etc. BPCA also has the advantage of being able to map impact changes to specific test scenarios. This is necessary as sometimes it may be required to identify process and data variants to be tested due to changes.

BPCA depends on 2 key inputs for its analysis and both should be mandatorily in place as pre-requisites -

1) A comprehensive Business Blueprint (BBP) (same as described in previous step)
2) TBOMS (Technical Bill of Materials)-a list of objects pertaining to a particular TCode/Business process as recognized by SAP.

The quality of analysis, and in fact the ability of BPCA to analyze is completely dependent on how comprehensive the Business Blueprint is and how detailed the TBOMS are.

SAP provides a means to automatically generate a Business Blueprint and also generate TBOMs. TBOMs generated this way are called static TBOMs or Semi-Dynamic TBOMS. However the maximum capability of test case optimization or test coverage gap identification can be realized only if some additional effort is taken to create the TBOMs manually (Dynamic TBOMs) and create the BBP as per the structure in the test management tool (typically HP-ALM, IBM Rationale).

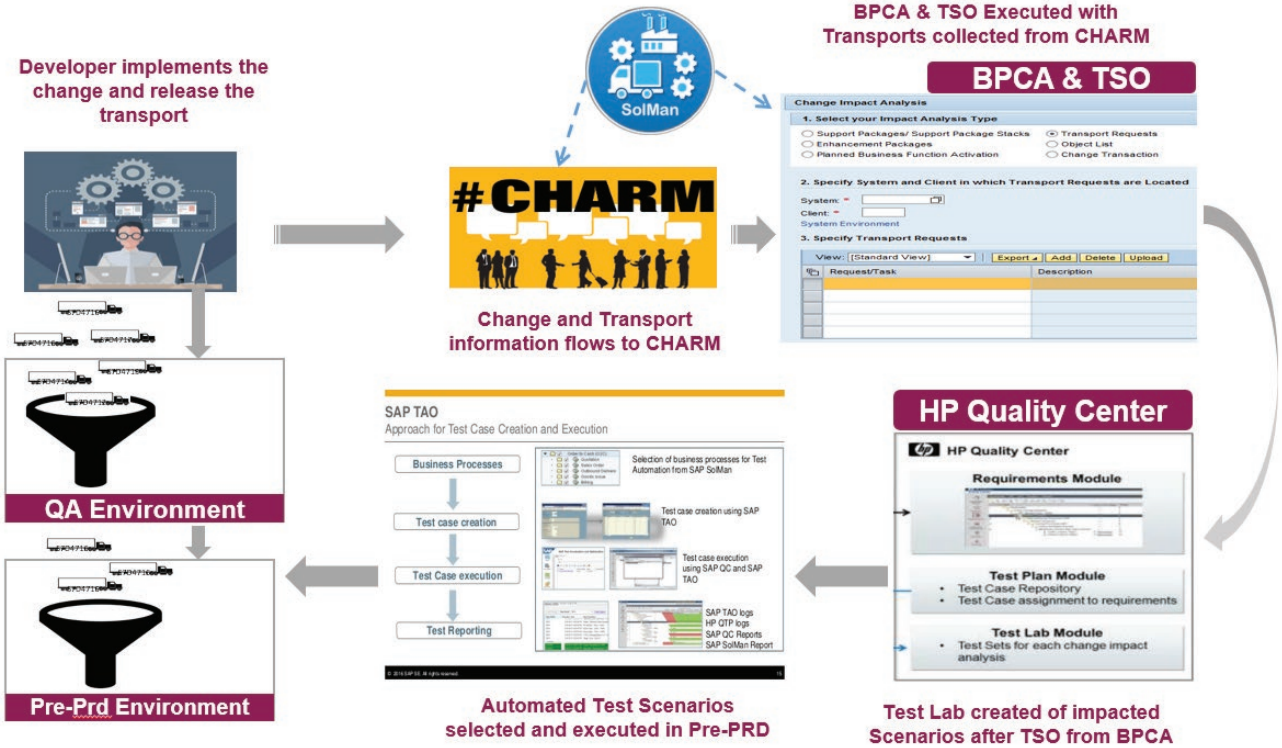
Step 5 – implementation of Change Management using SAP ChaRM

SAP ChaRM is a tool delivered with SAP Solution Manager (Ver 7.2) that manages activities performed during a change from design to testing to final promotion to production system. It tracks change requests, transport requests in the change management system for the entire business solution. Using ChaRM can help in identifying the changed transports and consequently, impacted areas automatically, thus playing a critical role in the implementation of continuous testing for SAP

Step 6 - Automated E2E Regression Testing

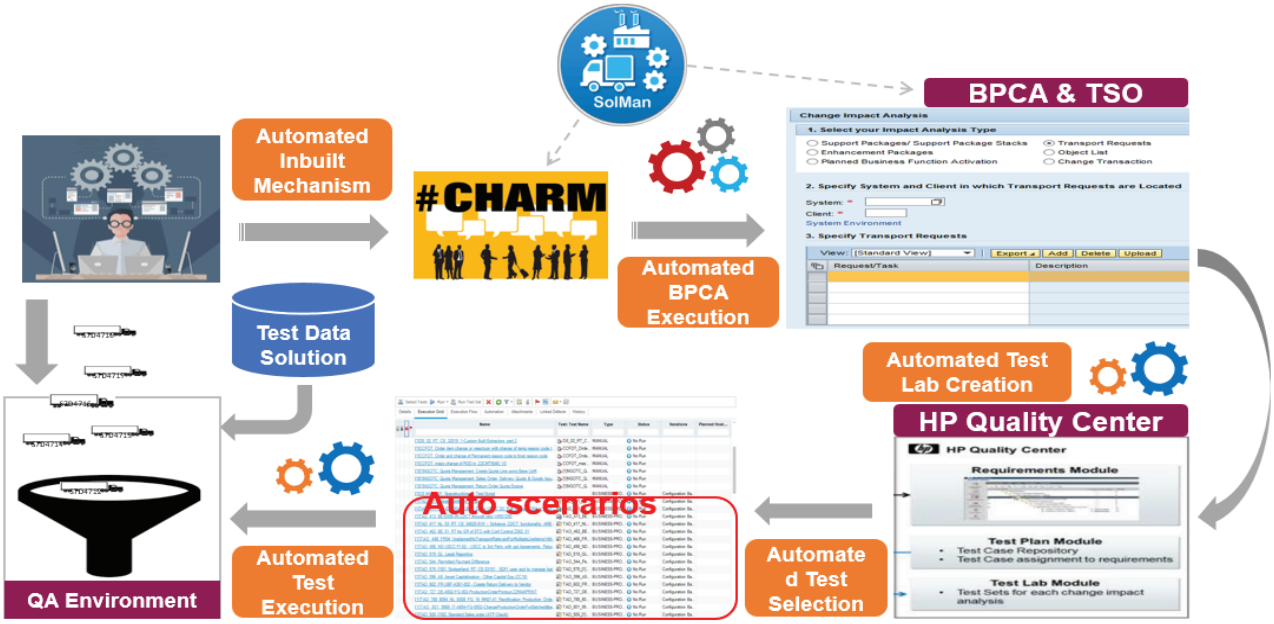
Continuous impact based automated testing for SAP can be implemented in 2 distinct phases. Phase 1 will focus on automated impact based testing. The focus of this phase would be to ensure that the pre-requisites mentioned above is implemented and work seamlessly and flawlessly. ChaRM will be the primary tool for information on any changed/newly introduced transport. The tester will physically look into the system to identify the transports. Once the transports are identified, the Transport IDs are fed into BPCA so that Impact analysis can be done. The result of the impact analysis will be the existing set of TBOMs (technical bill of materials) that have been impacted by changes. With the help of the mapping between the TBOMs and Test Scenarios (as done in Step 4), the set of test scenarios that need to be executed to test the impacted areas can be easily identified. In the example below, HP-ALM is shown as the Test management tool, but other Test management tools can also be used to create a Test Lab that will contain all the relevant test scenarios to be executed.

Regression testing – Impact based approach



To ensure that the system works fine, a 2 step approach can be considered to test the impact based testing approach. In the first step, once the test lab is created, a dedicated QA resource can be deployed to execute the combination of automated and manual scenarios. This will give the necessary oversight and confidence initially that the correct test scenarios have been identified by the process deployed. In case some scenarios are found to be missing or incorrect, some rework will be done to correctly map the scenarios, check if all TBOMs have been created, create missing scenarios and map them to requirements etc. It is recommended that 4-5 cycles are executed using a manual approach to make sure that all the requisite TBOMs are created and there is correct mapping between requirements and test scenarios.

Continuous Automated Regression Testing



The second and final step (Phase 2) that should be deployed to enable a fully automated E2E regression testing with a Continuous Testing approach would be to fully automate the process mentioned. A scheduled run (on a daily or more-frequent basis) can automatically check in CHaRM on changes to the system, automatically deploy a BPCA run, create a test lab and execute the automated scenarios with zero manual intervention. The same has been pictorially shown in the figure above. If this system is correctly put in place, a Continuous Testing solution that automatically completes impact based regression testing for all changes in SAP can be successfully realized

Conclusion

SAP implementations invariably tend to be complex with a significant effort required for improvements and maintenance. In a fast moving environment where change and adaptability to the market is measured in days if not hours, it is imperative that QA also play a role in helping business bring in the change and align to a continuous delivery model that has become more of a norm than an exception. The benefits of being able to recognize change in functionality and automatically validating that nothing is adversely impacted due to the change are significant. This ability will not only help in capturing any potential problems due to functional changes early in the lifecycle, but also ensure minimum manual effort is needed (thus reducing cost) to do the same.

The author has provided one of the approaches (with some assumptions on the tools used) to showcase the possibility of Continuous Testing for SAP systems. Other approaches are also possible based on the needs, tools and requirements of the consumers and appropriate changes can be adapted to suit the needs. But in summary, program directors of SAP implementations should earnestly consider adapting continuous testing to bring down the effort and cost of testing.

About the Author:

Navneeth Kowdle is a Consultant with Test Advisory Services (TAS) in Mindtree. He is an experienced IT Professional with 18+ years of extensive global experience in the areas of QA Consulting, QA Program Management, Account Management and Client Relationship Management