

ROI of an Application Testing – Repeatable Automated Test Suite

EGS helps design and deploy the test architecture and automation strategy for building, executing and maintaining a comprehensive automated regression test suite for SAP. The test architecture, the automation strategy and the SAP automated regression test suite will provide an efficient, continuous and ever-evolving SAP functionality verification process that will meet the following primary testing objectives:

- Speed up testing to accelerate releases
- Allow testing to happen more frequently
- Reduce testing costs and time
- Improve test coverage
- Ensure testing consistency
- Improve the reliability of testing and user confidence
- Easy maintenance during SAP application Upgrades

Challenges of Testing QA:

Application complexity

- Large scope: Many modules, many users, multi-org, large databases
- Customization: Configured to specific company, integration with many surround systems

Technology stack complexity

- Multi-tiered, hybrid (proprietary) architecture, Citrix for remote access, large databases
- Security layers (firewalls, proxies, SSL, digital certificates)

Application quality

- Patches/upgrades can number into the hundreds per quarter!

Vendor dependency

- You don't own the code, so your developers can't make fixes
- Vendor's Help Desk challenges ("you're not on the latest release!")

Test Environment

- Multiple environments, lots of hardware, storage

- May need specialized Citrix scripts to test 'remote access'
- Copy of production database often very large (100GB? 1 TB?); takes time to create, refresh
- Need at least two environments for upgrade testing –one w/ version x.y, another w/ x.z

Test Planning

- Ensuring good test coverage requires experience w/ the app
- High reliance on business users for workflow definition, execution review

Test Development

- Automated testing requires hundreds of scripts
- Upgrade testing requires two script versions when functionality changes btw releases
- Application-under-test is hard to freeze; production patches are being regularly applied

Test Execution

- Upgrades require execution in two environments and comparing results

Test Reusability

- High script maintenance required to match changed functionality
- Input data and expected results files need to be documented and re-populated

Two Distinct Functional Test Objectives : Different ROI

One-time deployment test: less investment, one-time return

- Tactical, short-term objective; "get the testing done" by release go-live date
- Requires little planning & design
- Basic script development
- Record many granular business processes
- Parameterize input data only
- Test execution sequencing and 'data passed forward' is done manually
- Script maintenance for next release: High; 40-50% rework
- Execution time for next release: Low-Med; 50-60% savings

Repeatable, maintainable test suite: higher investment, ROI grows by iterations

- Strategic, long-term objective; focus on repeatability, maintainability
- Requires planning and design of end-to-end processes, data dependencies, future likely application changes
- Comprehensive script development:

- Record 'macro' business processes
- Parameterize via mix of static data and dynamically-fetched data from database
- Parameterize expected results
- Automate passing-forward of data from one script to the next
- Script maintenance for next release: Low; 15-20% rework
- Execution time for next release: Very Low; 70-80% savings

Automated Testing Framework:

Foundation for Repeatability and ROI

Computing ROI

Quantify initial cost to automate

- Cost of software licenses, cost of script development

Quantify manual testing cost per iteration

- Burdened average hourly rates of manual testers * no. hours
- Add other costs, such as travel from remote user-testers, opportunity costs

Measure cost of automated testing per iteration

- No. hours to execute automated scripts * average hourly rate of testers

Measure the time spent maintaining scripts for the next iteration

- 10-15% is a typical average over the lifetime of the scripts for a well-designed framework

Multiply the difference btw Man\$ & Auto\$ and sum it over the number of test iterations i

- This yields the execution cost difference for i iterations

Example 1: Process Manufacturing

Fortune 100 Company

- Diversified, multinational supplier of engineered materials to customers spanning industries, from paper and energy to plastics and construction
- International operations with high cost of production stoppage
- Nervous user community, upgrade-averse due to previous experiences
- Bringing on int'l business units driving upgrade (new functionality)
- U.S. Operations (13 Plants, approx. 1000 Users) using Oracle 24/7

Project

- Upgrade 115.7 to 11.5.9 Financials & Process Manufacturing
- Business Driver: Required 11.5.7 manufacturing functionality
- Success Factor: Zero interruption of current operations

Results

- Surfaced a configuration error that would have been a show-stopper, and corrected in time
- Upgraded over Thanksgiving weekend and went live on schedule
- Developed 193 test scripts
- Logged 89 defects; fixed all (37) priority 1 & 2
- Zero defects after go-live
- Positive ROI during maintenance upgrade 3 months later

Example 1: Hi-Tech Manufacturing

- Manufactures cell/radiophones for world-wide wireless providers
- 3 US plants, 2 abroad
- Committed to automated testing, but not implemented on OA
- 1008 manual test cases developed by another division was starting point

Project

- Objective: Validate that the business properly performs all financial & manufacturing functions supported by OA11i & its interfaces
- New implementation of 11.5.9 (migrating from 10.7)
- 14 modules, Financials & manufacturing
- 18 system interfaces

Results

- Completed testing and ship-accepted on time
- 5 test cycles, 2212 total hours (manual + automated)
- Cut test execution time in cycles 2-5 by average of 60%
- 241 automated scripts replaced 350 macro test cases*
- Automated testing paid for itself in cycle 5
- Zero defects after go-live

(*e.g., create requisition, two approvals, create PO, one approval, and receive goods)



Headquarters

Excel Global Solutions Inc.
2727 N. Grandview Blvd.,
Suite 117, Waukesha WI 53188 USA
Ph: 262-347-4911



www.excelglobalsolution.com



info@excelglobalsolution.com