SAP Implementation: DO’s and DONT’s – CitraTest & CitraTest APM

As an IT pro supporting SAP, or tackling an SAP implementation, simplicity is certainly not the first attribute that comes to mind. In fact, as an enterprise suite of ERP, CRM, PLM, SCM and SRM applications, SAP links and integrates many business processes across your organization. While 253,500 customers in 188 countries successfully rely on SAP to accelerate their business operations, inadequate planning, lack of integrated testing during each implementation phase, improper SAP monitoring, or even poorly managed user training can all result in costly pitfalls and mistakes for you and your organization, just see some examples below.

Some time ago, Wall Street Journal reported that Avon Products Inc. was pulling the plug on a $125 million SAP overhaul that had been in the works for four years after a test of the system drove away many of the salespeople who fuel the door-to-door cosmetics company's revenue. It is not clear whether the failure was due to the software or the implementation, or both, but the consequences were damaging to the company’s business.

Recently a $50 million SAP implementation by San Diego — the biggest IT project ever undertaken by the city, designed to provide a core IT architecture for 15 years — ended up tripling employees’ workloads for certain types of tasks! Pair this up with other highly visible SAP implementation failures at Marin County, or at Hershey’s where order fulfillment time doubled to 12 days and you will get the point. Comprehensive and realistic testing during each implementation phase --by simulating and validating user interactions with end-to-end SAP workflows -- is critical for success. Similarly, proactively monitoring availability and response time of production systems, as perceived by your end-users, are critical to meet SLAs and find problems before users are impacted. Here are some tips to help you get started.

PRE-IMPLEMENTATION GUIDELINES: PLANNING, TESTING & AUTOMATION

• **DO #1. Document existing business processes**: Since SAP touches so many business functions, take the time to understand and document how each business user (project managers, HR, sales, financial teams, etc.) completes existing processes and common workflows.

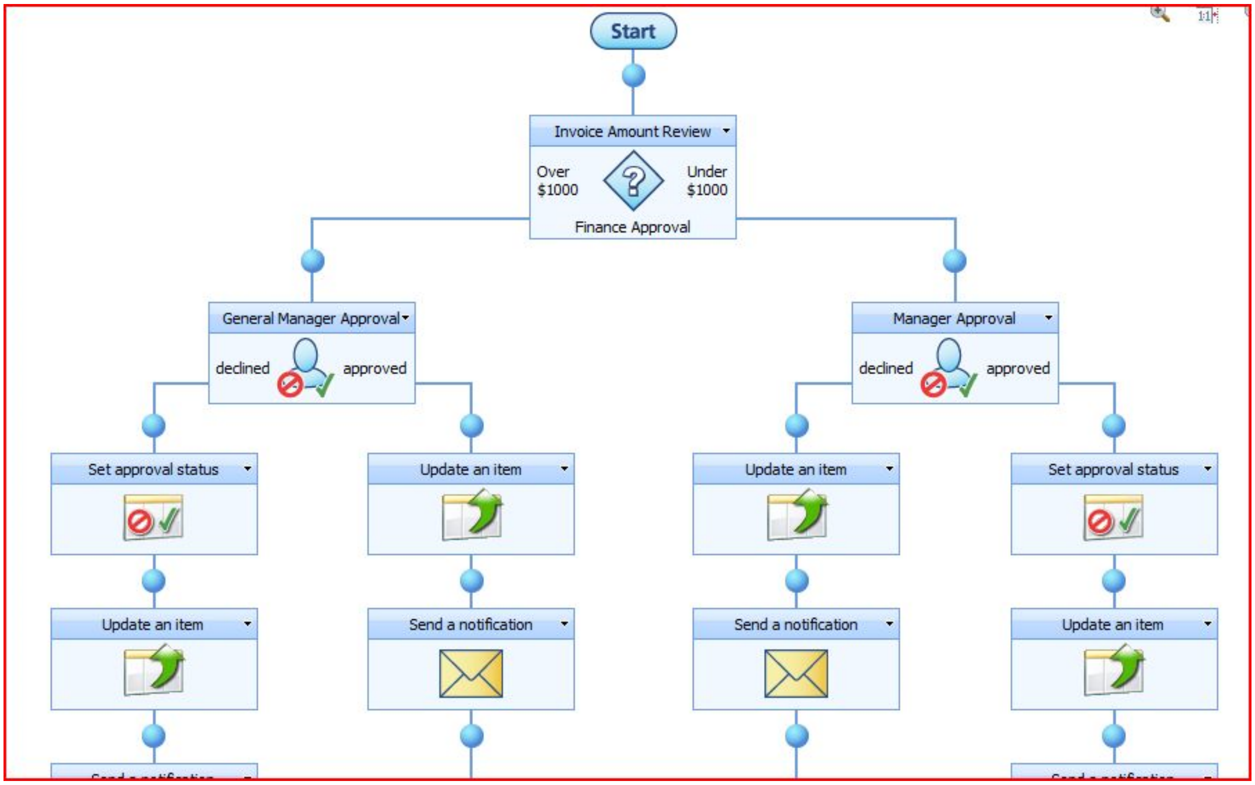

Figure 1- Document your existing processes

• **DO #2. Redesign business processes**. Many of the SAP implementation pitfalls that were mentioned above were primarily due to simply replicating existing processes instead of redesigning them with SAP for more efficiency. Therefore, don't skip this crucial, yet time-consuming step, and map and optimize all user requirements, for each group of users, into new workflow models. This information will also help you guide your testing and monitoring efforts later on.

• **DO #3: Reduce the number of customized functions**. Try to use as many objects and functions as you can from the off-the-shelf version of SAP instead of writing your own custom code. You might run into quality issues with custom objects later on, plus they might not work as expected if/when you roll-out a new version of SAP.

• **DO #4: Test and validate end-to-end workflows from an end-user perspective**. Begin manual testing as early as possible, so users can provide feedback on SAP business processes and validate reporting structures. Later on, automate your testing processes to save time, increase efficiency and thoroughly test and re-test each workflow as needed to ensure quality and top-notch SAP performance. Look for a testing tool that can help you simulate and replicate all real user interactions with SAP, generate test scripts for critical workflows (e.g. enter an order, generate a shipping receipt, create an invoice, and so on), and then automate the entire testing process.

• **DO #5: Build comprehensive testing scenarios**. Build comprehensive testing scenarios with custom grouping based on physical locations (Headquarters, NYC, remote offices), SAP functionality or user groups/roles. Everything should be measured in absolute values, percentages or statistical deviations. For example, SLAs for SAP load time, login time, how long it takes after a "button" is clicked or until the next window or item appears, and so on. That way, you can understand how SAP performs per location, per step or per group, both historically and in real-time, identify underperforming components or problematic trends, and ensure that SLAs are being met.

Once SAP is implemented, don't forget to build a comprehensive monitoring strategy. Availability and response times must be monitored round the clock and measured against desired service level goals.

PROACTIVE PRODUCTION MONITORING TO MEET SERVICE LEVEL GOALS

Did you know that the #1 customer complaint faced by organizations that use SAP is "SAP is too slow"? Therefore, instead of infrastructure-centric SAP monitoring, look for synthetic monitoring capabilities to automate the driving and monitoring of your SAP applications — just like a real user would do — visually examining the desktop and responding to changes, driving the keyboard and mouse and continuously taking availability and response time measurements along the way, to get alerted *before* users are impacted. Ideally, you should reuse your test automation scripts for production monitoring, and monitor all critical functions and workflows that you tested on a 24x7x365 basis to detect early performance degradation signs. Here are some additional tips:

- **DO #6. Build a comprehensive Monitoring Plan**. Just like you documented your business processes and workflow for each set of business users, take the time to plan and document your SAP monitoring strategy, including key areas like: Monitoring scripts (so you can proactively measure complete end-to-end user interactions with SAP), Alerting Policies (who should be notified when specific functions slow down), Notification Groups, Reporting Distribution, Monitoring Locations, Service Level Agreements and so on.

- **DO #7. Focus on Essentials workflows first**. Tackle the basics, including key financials, HR/payroll, supply chain processes and workflows, and add more monitoring scripts later. In addition, consider using a dedicated "SAP user" monitoring account to keep your SAP data secure.

- **DO #8. Look for non-invasive agentless SAP monitoring** that will not change your application code. That way you can ensure no impact on your SAP production system and complete monitoring accuracy.

- **DO #9. Be aggressive in your SAP monitoring thresholds (availability and response time).** Since proactive SAP monitoring helps you identify problematic trends early on and before your end-users are impacted, you should baseline and setup both warning and critical thresholds for all monitored SAP transactions.

- **DO #10. Don't monitor SAP in isolation**. Integrate SAP monitoring, alerting and performance dashboard views within your existing IT management tools and frameworks tools to simplify troubleshooting, increase IT efficiency and streamline your response procedures.

Are you ready to avoid SAP problems and ensure success? If so, think about your testing and proactive monitoring strategy. Check out CitraTest (SAP test automation) and CitraTest APM (proactive SAP response time SLA monitoring from a user perspective).