

AUTOMATYCZNE TESTY FUNKCJONALNE W APLIKACJI SAP

Rafał KLAUS, Tomasz MAĆKOWSKI

Streszczenie: Testowanie ma istotne znaczenie podczas projektów wdrożeniowych i programistycznych w środowisku aplikacji SAP. Ze względu na zwiększającą się konkurencję na rynku wdrożeniowym powstaje presja aby projekty przeprowadzane były w coraz krótszym czasie. Jednocześnie projekty te stają się coraz bardziej skomplikowane i obejmują swoim zakresem coraz szersze obszary. Bardzo duża liczba procesów biznesowych i ich wariantów czyni niemożliwym dokładne ręczne przetestowanie wymaganej funkcjonalności. Rozwiązaniem tego problemu jest automatyzacja testów.

Słowa kluczowe: mySAP, testy funkcjonalne, eCATT, QTP.

1. Wprowadzenie

Coraz więcej przedsiębiorstw decyduje się na wdrażanie kompleksowych systemów informatycznych wspierających zarządzanie. Motywują do tego takie czynniki jak szukanie możliwości usprawnienia działania operacyjnego firmy, usprawnienie obsługi klientów, zwiększenie nadzoru nad finansami, zmniejszenie zapasów magazynowych, sprostanie nowym wyzwaniom rynkowym. Systemy te stale poszerzają swoją funkcjonalność i stają się coraz bardziej skomplikowane. Jak pokazują dane raportów dotyczących sukcesów projektów informatycznych takie kompleksowe wdrożenia nie są prostym zadaniem. Tylko około 34% projektów kończy się pełnym sukcesem - wdrożeniem systemu zgodnie z założonym harmonogramem i budżetem. 15% projektów zostaje zarzucona przed ukończeniem, a 51% kończy się z opóźnieniami, przekroczonym budżetem lub z mniejszym niż pierwotnie zakładano zakresem wdrażanych funkcjonalności [9]. Jednym z czynników, który w znaczący sposób może przyczynić się do opóźnień projektu i dodatkowych kosztów jest niedostateczne przetestowanie wymaganych funkcjonalności i co za tym idzie błędne lub niezgodne z wymaganiami działanie końcowego systemu.

Najczęściej wybieraną na świecie, przez duże przedsiębiorstwa, platformą informatyczno-biznesową są produkty firmy SAP. SAP od ponad 30 lat specjalizuje się w rozwijaniu zintegrowanych systemów wspierających zarządzanie przedsiębiorstwami.

Produkty SAP znane są z bardzo dużych możliwości konfigurowania i rozszerzania swojej funkcjonalności. Te właściwości będące niewątpliwie zaletą mają jednak istotny wpływ na zwiększenie ryzyka związanego z niepowodzeniem projektu wdrożeniowego.

Rozwiązaniem problemów związanych z zapewnieniem jakości wdrażanych i rozwijanych aplikacji SAP może okazać się automatyczne testowanie.

W referacie porównano dostępne narzędzia służące do automatyzacji testowania w środowisku SAP NetWeaver, będącym platformą technologiczną dla wszystkich rozwiązań aplikacyjnych SAP. Analizie poddano narzędzia służące do automatyzacji testów funkcjonalnych pozwalające na weryfikację zgodności działania aplikacji z wymaganiami.

2. Metodyka wdrożeniowa

Wdrożenie systemu SAP jest skomplikowanym przedsięwzięciem informatycznym i organizacyjnym. Przedsięwzięcie to wymaga systematycznego i przemyślanego podejścia. Jedną z metod wdrażania jest metodyka ASAP (AcceleratedSAP). ASAP definiuje jakie czynności powinny zostać wykonane w ramach projektu, kto powinien je wykonywać i w jakim harmonogramie powinny się one odbywać. W ramach metodyki SAP dostarcza również zestaw narzędzi, listy kontrolne, szablony dokumentów i instrukcje wspierające cały proces wdrożeniowy.

ASAP definiuje model procesu w podziale na następujące fazy:

- Przygotowanie projektu. W fazie przygotowania projektu następuje zdefiniowanie celów i harmonogramu projektu. Wyznaczone zostają również zespoły projektowe odpowiedzialne za poszczególne obszary funkcjonalne.
- Koncepcja biznesowa. W fazie koncepcji biznesowej definiowane są szczegółowe wymagania jakie docelowa aplikacja ma spełnić. Identyfikowane są również procesy, które nie są pokryte standardową funkcjonalnością w celu przygotowania specyfikacji dla przygotowania rozszerzeń.
- Realizacja. W fazie realizacji następuje konfiguracja i dostosowanie systemu zgodnie z koncepcją biznesową. Na tym etapie prowadzone są również prace programistyczne mające na celu zapewnienie funkcjonalności nieobjętej standardowym zakresem aplikacji.
- Przygotowanie do startu. W fazie przygotowania do startu przeprowadzane są szkolenia użytkowników końcowych i ładowane są dane konieczne do rozpoczęcia pracy produktywnej na systemie.
- Start produktywne i wsparcie. W fazie startu produktywnego następuje początek faktycznej pracy z systemem. Użytkownicy korzystają jeszcze ze wsparcia zespołu projektowego w celu rozwiązywania bieżących problemów. W tej fazie następuje również ocena całego przedsięwzięcia [2].

Aby zapewnić spójność danych i nieprzerwaną pracę zalecane jest podejście polegające na wprowadzeniu dwóch lub więcej systemów w środowisku projektowym. Najbardziej pożądanym jest pejzaż trój-systemowy.

Poszczególne systemy w tym pejzażu spełniają następujące role:

- System konfiguracyjno-rozwojowy (DEV)
System konfiguracyjno-testowy służy do przygotowywania konfiguracji systemu i jednocześnie jest środowiskiem rozwojowym dla wszelkich artefaktów programistycznych. Jest to jedyny system w całym pejzażu na którym dozwolone są zmiany w konfiguracji i w repozytorium ABAP.
- System zapewniania jakości (QAS).
System zapewniania jakości służy do testów funkcjonalnych i integracyjnych. Powinien zawierać dane zbliżone do rzeczywistych najczęściej uzyskuje się to poprzez załadowanie systemu danymi z poprzednich systemów lub kopią danych z systemu produktywnego. System QAS pozwala sprawdzić czy wszystkie wymagane obiekty zostały zawarte w transportach i przetestować ich działanie w warunkach maksymalnie zbliżonych do tych w których będą działały w rzeczywistości.
- System produktywne (PRD).
System produktywne jest właściwym środowiskiem w którym funkcjonuje firma i w którym przetwarzane są dane.

Wyróżnienie w pejzażu systemowym trzech systemów związane jest z funkcjonalnością systemu transportowego. System transportowy udostępnia mechanizmy zapisywania wszelkich zmian konfiguracyjnych i zmian w ABAP Workbench dokonywanych na systemach DEV. Zmiany te zapisywane są w tzw. zleceniach transportowych. Zlecenia te przenoszone są ze wszystkimi przynależnymi obiektami do kolejnych systemów (QA i PROD) co pozwala zachować spójność i utrzymywać pojedynczy system na którym wprowadzane są zmiany.

3. Testy funkcjonalne

Testy funkcjonalne, wykonywane w końcowych etapach fazy realizacji i w fazie przygotowania do startu, mają potwierdzić zgodność działania systemu z wymaganiami i założeniami przyjętymi w koncepcji biznesowej. Skrypty testowe automatyzujące testowanie funkcjonalne często służą później jako podstawa do testowania regresyjnego w przypadku wprowadzania zmian konfiguracyjnych, rozwijania nowych funkcjonalności uaktualnień systemu, czy wgrzywaniu poprawek. Testy te mają za zadanie sprawdzić czy powyższe działania nie mają negatywnego wpływu na działalność operacyjną w systemie produktywnym.

Do automatyzacji testów funkcjonalnych możemy wykorzystać jedno z dwóch narzędzi. Pierwsze z nich to Extended Computer Aided Test Tool (eCatt)- narzędzie zintegrowane z ABAP Development Workbench, dostępne od wersji 6.20 serwera aplikacyjnego WAS. Drugim z badanych narzędzi jest produkt QuickTest Professional, z rozszerzeniem dla testowania aplikacji SAP, firmy Mercury. QuickTest Professional jest certyfikowany przez SAP do współpracy z platformą NetWeaver i jest najczęściej wybieranym zewnętrznym narzędziem do testowania.

Narzędzia te zostały przeanalizowane ze względu na możliwości i ograniczenia jakie prezentują w czterech obszarach: przygotowania systemu do pracy, tworzenia skryptów testowych, zarządzania danymi testowymi i zarządzanie procesem testowania.

3.1. Extended Computer Aided Test Tool - eCATT

Extended Computer Aided Test Tool (eCATT) jest złożonym narzędziem będącym składnikiem oprogramowania podstawowego dostarczanego przez SAP razem z serwerem aplikacji SAP WAS. Narzędzie to pozwala zarówno na wykonywanie prostych testów ograniczonych do pojedynczych komponentów czy programów jak i skomplikowanych scenariuszy testowych przebiegających przez wiele rozproszonych systemów.

Przygotowanie systemu

eCATT został zaprojektowany jako narzędzie zarządzane centralnie, operujące na jednym serwerze. W związku z tym należy desygnować specjalny serwer aplikacji służący jako platforma dla eCATT. Nie jest wymagane aby był to serwer spełniający tylko tę rolę. Najlepiej jako centralny system testowy wybrać jeden z systemów rozwojowych (DEV). Serwer eCATT komunikuje się z pozostałymi serwerami za pomocą wywołań RFC konieczne jest zapewnienie łączności sieciowej pomiędzy serwerami i konfiguracja połączeń zdalnych z systemu eCATT do pozostałych. Wykonywanie automatycznych testów na zdalnych systemach wymaga zawsze manualnego zalogowania się poprzez podanie użytkownika i hasła. W niektórych scenariuszach może być to uciążliwe i może obniżyć efektywność procesu testowania. Aby tego uniknąć parametry autoryzacyjne połączenia można zapisać na stałe w jego konfiguracji rozwiązuje to problem ale

jednocześnie stanowi zagrożenie dla bezpieczeństwa i poufności danych. Najlepszym rozwiązaniem jest zestawienie zaufanego połączenia RFC pomiędzy systemami. Pozwala to na uniknięcie konieczności manualnego logowania przy czym dane autoryzacyjne nie są nigdzie zapisywane.

Biorąc pod uwagę zarówno poziom bezpieczeństwa i możliwość pełnego przetestowania aplikacji najlepiej przeprowadzić konfigurację zezwalającą na pełne uprawnienia dla zaufanych połączeń eCATT na wszystkich mandatach systemów rozwojowych DEV i zapewniania jakości QAS oraz całkowite zabronienie eCATT na serwerach produkcyjnych PROD. Ze względu na istotność tych ustawień decyzje te powinien podejmować Lider Testów w porozumieniu z Kierownikiem Projektu oraz zespołem odpowiedzialnym za bezpieczeństwo. Ostatnią rzeczą konieczną do wykonania przed przystąpieniem do opracowywania skryptów testowych jest zdefiniowanie kontenera danych systemowych. Kontener ten przechowuje dane o systemach biorących udział w procesie testowania łącząc ich logiczne nazwy z definicjami fizycznych połączeń RFC. Kontener danych systemowych wykorzystywany jest w następnie w skryptach testowych do wskazywania systemów docelowych dla poszczególnych kroków skryptów. Zastosowanie kontenera pozwala odseparować informację o powiązaniach i połączeniach od faktycznych testów i pozwala na zmianę parametrów bez konieczności ingerencji w strukturę skryptów testowych.

Tworzenie skryptów testowych

Zadaniem skryptów testowych jest weryfikacja poprawności badanych programów i ich zgodności z założeniami z fazy koncepcji biznesowej. Przystępując do ich definiowania należy wiedzieć jaki program lub proces mamy przetestować i jaki jest spodziewany rezultat. Te informacje powinny być jasno zdefiniowane w opisie przypadku testowego. Skrypty testowe składają się z atrybutów opisowych i administracyjnych oraz sekwencji poleceń. Komunikacja i wymiana danych z pilotem testu i innymi skryptami może odbywać się jedynie poprzez parametry importowe i eksportowe (interfejs skryptu). Testując aplikację w środowisku SAP przy użyciu narzędzia eCATT należy określić jej rodzaj. Jest to związane z architekturą i możliwościami samego narzędzia. eCATT przeprowadza testy z użyciem określonych pilotów testów. Piloty te działają w różnych technologiach przez co przystosowane są do testowania różnych rodzajów artefaktów:

- Pilot testu TCD - Służy do testowania podstawowych programów dialogowych nie zawierających kontrolki ActiveX.
- Pilot testu SAPGUI - Służy do testowania programów dialogowych zawierających kontrolki ActiveX, takiej jak np. zaawansowana kontrolka tabelowa ALV.
- Pilot testu FUN - Służy do testowania modułów funkcyjnych.
- Pilot testu WEBSERVICE - Służy do testowania usług sieciowych.
- Pilot testu WEBDYNPRO - Służy do testowania aplikacji internetowych w technologii Webdynpro.

W obrębie jednego skryptu testowego można korzystać z różnych pilotów, przez co możliwe jest testowanie procesów, których poszczególne kroki wykonywane są za pomocą programów tworzonych w różnych technologiach. Najczęściej używanymi pilotami testów są TCD i SAPGUI. Pozwalają na testowanie wszelkich programów zawierających interfejs użytkownika. Stosując obydwie z nich mamy możliwość nagrywania przebiegu czynności użytkownika w celu późniejszego wykorzystania w skrypcie testowym. Implikuje to podejście do tworzenia skryptów składające się z trzech głównych kroków:

- Nagranie skryptu - Nagranie przebiegu programu przez wykonywanie czynności zgodnie z przypadkiem użycia.

- Dostosowanie - Ręczne poprawienie kodu skryptu, wstawienie punktów kontrolnych i poleceń zapisujących do logu.
- Przystosowanie do ponownego użycia - Zdefiniowanie interfejsu wejściowego i wyjściowego, parametryzacja poleceń wewnątrz skryptu.

Nagrywając skrypty TCD i SAPGUI należy powtórzyć nagranie co najmniej 3 razy dla każdej transakcji w procesie. Do tego należy przygotować zestaw danych gwarantujący bezbłędne przejścia procesu. Powtórzenia te najlepiej wykonywać w trybie synchronicznym, dialogowym. Umożliwia to śledzenie przebiegu skryptu na ekranie i wychycenie ewentualnych nieprawidłowości w jego przebiegu. Za drugim wykonaniem skryptu mogą pojawić się komunikaty błędów, dodatkowe okna itp. zmieniające sekwencję przetwarzania co w efekcie może prowadzić w późniejszym etapie do błędnych wyników. Skrypt można uznać za gotowy do dalszego przetwarzania po kilkakrotnym bezbłędnym przejściu. W niektórych transakcjach mogą pojawiać się domyślne wartości. Ich źródłem może być konfiguracja, domyślne warianty ekranu, domyślne wartości zachowane w profilu użytkownika. Ważne jest aby wartości te były takie same na wszystkich systemach. W tym celu należy sprawdzić spójność konfiguracji i użycie tych samych domyślnych wariantów ekranu. Aby uniknąć błędów wynikających z rozbieżności wynikających z wartości domyślnych z profilu użytkownika należy utworzyć na wszystkich systemach konto użytkownika przeznaczone tylko do wykonywania testów automatycznych i ustawić na wszystkich systemach te same wartości w profilu.

Sprawdzanie poprawności wyników

Zadaniem skryptu testowego jest weryfikacja poprawności działania badanej aplikacji. Żeby tę poprawność zweryfikować należy do skryptów testowych dodać odpowiednie punkty kontrolne sprawdzające wymagane warunki. W zależności od procesu podlegającemu testowaniu warunki te mogą się zmieniać. Można wyróżnić jednak typowe sytuacje podlegające weryfikacji podczas testowania funkcjonalnego aplikacji SAP: weryfikacja komunikatów, wartości zmiennych i zapisów do baz danych.

3.2. Quick Test Professional

Quick Test Professional (QTP) firmy Mercury Interactive Corporation jest oprogramowaniem służącym do tworzenia automatycznych testów funkcjonalnych i regresyjnych. Oprogramowanie to przystosowane jest do testowania szerokiej gamy aplikacji wykonywanych w takich technologiach jak Java Enterprise Edition, Microsoft .Net, Adobe Flex, oraz aplikacji klasy ERP/CRM. QTP jako jedna z nielicznych aplikacji do testowania została zatwierdzona przez SAP jako 'Certyfikowane rozwiązanie dla SAP Netweaver'. Oznacza to, że została ona przebadana pod kątem zgodności z interfejsami integracyjnymi platformy NetWeaver.

QTP jest narzędziem ogólnego przeznaczenia i może być stosowany do testowania zarówno aplikacji internetowych dostępnych przez przeglądarkę jak i okienkowych aplikacji na platformie Windows. Do testowania aplikacji SAP został przygotowany specjalny pakiet rozszerzający możliwości QTP o nazwie QuickTest Professional Add-in for SAP Solutions. To rozszerzenie zawiera zestaw obiektów testowych specjalnie dostosowanych do współpracy z aplikacjami SAP. QTP w interakcji z tymi aplikacjami wykorzystuje SAP GUI Scripting API, czyli ten sam interfejs programistyczny jaki wykorzystywany jest przez pilota testu SAPGUI w narzędziu eCATT. Ogranicza to zatem zakres testów do programów i transakcji dialogowych testowanych metodą nagrywania i odtwarzania.

4. Porównanie i ocena

Aby zapewnić możliwość kompleksowego, automatycznego testowania funkcjonalnego w środowisku aplikacji SAP wybór odpowiednich narzędzi ma kluczowe znaczenie.

Tab. 1 Porównanie głównych cech Mercury QuickTest Pro i eCATT

Cecha	Mercury QuickTest Pro + SAP Add-in	Extended Computer Aided Test Tool
Koszty licencji	Dodatkowo płatne	Wliczone w licencję aplikacji SAP
Testowanie innych aplikacji (okienkowe, terminalowe, inne pakiety ERP)	Pełne wsparcie dla aplikacji okienkowych Windows. Współpraca z aplikacjami Siebel, Oracle®, PeopleSoft	Brak możliwości testowania aplikacji poza SAP
Testowanie aplikacji internetowych dostępnych przez przeglądarkę	W pełni wspierane	Wsparcie tylko dla technologii SAP Webdynpro
Wsparcie procesu testowania (planowanie, zarządzanie, analiza)	Wymaga zakupu osobnego pakietu Test Director	Dostępne razem z pakietem
Sposób tworzenia skryptów testowych	Nagrywanie przebiegu działania aplikacji + dostosowanie	Nagrywanie przebiegu działania aplikacji + dostosowanie. Możliwość ręcznego definiowania skryptu.
Język skryptowy	SQA Basic	ABAP

Główną zaletą eCATT jest to, że jest on dostarczany jako składnik serwera aplikacji WAS, a co za tym idzie nie wymaga dodatkowych nakładów związanych z zakupem licencji i kosztów wsparcia producenta produktu. Jego funkcjonalność ograniczona jest jednak w znacznym stopniu do środowiska SAP. Podstawowym rodzajem testów są testy programów dialogowych z użyciem pilotów testów TCD lub SAPGUI. Możliwe jest jednak wykorzystanie pozostałych pilotów do testowania np. usług sieciowych Web Services, co w niektórych przypadkach umożliwia przetestowanie procesu z udziałem systemów działających w innych technologiach.

W tym kontekście większe możliwości prezentuje QTP. Ponieważ jest to narzędzie ogólnego przeznaczenia, jego funkcjonalność związana z testowaniem aplikacji SAP jest tylko jedną z wielu możliwości. Dzięki temu można automatyzować procesy testowania rozciągające się na kilka systemów, z których nie koniecznie wszystkie stanowią systemy SAP. Przewagą QTP widać również w kwestii automatycznego testowania aplikacji internetowych dostępnych przez przeglądarkę. Umożliwia on testowanie dowolnych aplikacji w odróżnieniu od eCATT, który wspiera jedynie aplikacje tworzone i uruchamiane na serwerze aplikacji SAP w technologii Webdynpro. Minusem QTP jest natomiast jego brak wsparcia dla pracy grupowej i wsparcia dla kompleksowego zarządzania procesem testowania. Funkcje związane z planowaniem testowania, przydziałem testów do wykonania przez odpowiednie osoby, monitorowaniem statusu testów w całym projekcie są niedostępne. Możliwa jest jedynie integracja z pakietem Test Director zapewniającym te funkcjonalności, ale wymagającym zakupu osobnej licencji. Dla środowiska eCATT natomiast wsparcie procesu testowania dostępne jest bez dodatkowych

opłat. Wyżej wymienione funkcje realizuje SAP Test Workbench zapewniając kontrolę nad procesem testowania nawet w dużych i skomplikowanych projektach. Wszystkie informacje składowane są na centralnym serwerze umożliwiając ich dalszą analizę.

Porównując kwestię łatwości tworzenia i modyfikowania skryptów testowych należy wyróżnić QTP za bardziej przejrzysty i intuicyjny interfejs użytkownika. Podstawowa funkcjonalność związana z tworzeniem skryptów testowych zarówno w eCATT jak i QTP opiera się na mechanizmie nagrywania czynności użytkownika. eCATT jest bardziej skomplikowany w użyciu ale daje większe możliwości analizy przebiegu i interakcji z testowaną aplikacją. Tester ma możliwość bezpośredniego dostępu do bazy danych, modułów funkcyjnych, zbiorów konfiguracji z poziomu skryptu za pomocą wstawek w języku ABAP. Dzięki temu możliwe są bardziej szczegółowe sprawdzenia niż tylko na poziomie interfejsu użytkownika.

Warto również zaznaczyć kwestie niezależności skryptów eCATT od języka. Wszystkie obiekty znajdujące się na ekranie identyfikowane są przez nazwy techniczne, w odróżnieniu od QTP, który identyfikuje je poprzez nazwy faktycznie wyświetlane. To rozróżnienie może mieć kluczowe znaczenie szczególnie w projektach typu roll-out, w których wykorzystywany jest cały zestaw testów z centralnego rozwiązania. Może to skutkować koniecznością przerabiania wszystkich skryptów testowych, jeżeli język w kraju docelowym jest inny niż język w którym przygotowywane były skrypty testowe dla rozwiązania korporacyjnego.

Biorąc po uwagę powyższe czynniki można przedstawić wniosek, że wybór narzędzia służącego do automatyzacji testów w środowisku SAP nie jest prostym zadaniem i uzależnione jest od wielu czynników. Narzędzie eCATT dostępne jest razem z serwerem aplikacji przez co nie wymaga dodatkowych inwestycji. Jest zatem dobrym wyborem dla projektów w których budżet przeznaczony na testowanie jest znacznie ograniczony. Jednocześnie należy pamiętać, że eCATT nie umożliwia testowania aplikacji poza systemami SAP. Jeżeli wykorzystywana jest duża liczba takich aplikacji, lub wybrany został model wdrożenia, w którym jedynie centralny system pochodzi od SAP, a pozostałe takie jak CRM, czy SRM od innych dostawców to lepszym wyborem jest inwestycja w narzędzie QuicktestPro.

5. Zakończenie

Automatyzacja testowania pozwala na:

- zmniejszenie kosztów związanych z błędnie działającym oprogramowaniem poprzez wczesne wykrycie błędów;
- zapewnienie powtarzalności procesu testowania i wyeliminowanie ew. błędów wprowadzonych przez człowieka;
- możliwość testowania regresyjnego w przypadku zmiany wersji lub wgrzywania pakietów serwisowych, przez co zapewnione jest poprawne i ciągłe działanie kluczowych procesów przedsiębiorstwa;
- bardziej optymalną użycie zasobów przez możliwość uruchamiania zestawów testów w momencie najmniejszego obciążenia np. w nocy lub w weekend.

W przypadku aplikacji tworzonych lub dostosowywanych w środowisku SAP możliwa jest automatyzacja na dwóch poziomach: testowania jednostkowego i testowania funkcjonalnego. Testowanie funkcjonalne zapewniają natomiast poprawność działania aplikacji i ich zgodność z koncepcją biznesową. Testowanie na tym poziomie powinno być ściśle skoordynowane z przebiegiem całego projektu. Skuteczność automatycznego

testowania funkcjonalnego uzależniona jest w znacznym stopniu od jakości planu testowania i opisu przypadków testowych powstających we wczesnych etapach projektu. Testowanie automatyczne może usprawnić jedynie proces wykonywania i oceny przypadków testowych czyniąc ich wykonani mniej czasochłonnym i bardziej powtarzalnym. Nie zastąpi to jednak procesu analizy i definiowania wymagań. Przystępując do testowania automatycznego należy mieć jasną wizję jaki procesy będą podlegały automatyzacji testowania i jakie są oczekiwane rezultaty z nimi związane. Testowanie automatyczne jest procesem kosztownym. Na jego koszt składają się:

- Koszty projektowania przypadków testowych
Przypadki testowe muszą zostać zaprojektowane i udokumentowane. Należy określić ich przebieg i spodziewany rezultat. Jeżeli będą one wykonywane manualnie, to powstała dokumentacja staje się instrukcją dla osób testujących aplikację. W przypadku automatyzacji testowania dokumentacja ta staje się podstawą dla tworzenia skryptów testowych.
- Koszty narzędzi testowych
Koszty te mogą stanowić duży procent budżetu przeznaczonego na testowanie. Przed wyborem narzędzi wykorzystywanych do automatyzacji testowania należy głównie przeanalizować systemy które chcemy objąć tym procesem. Jeżeli w pejzażu systemów przeważają rozwiązania SAP wystarczające może okazać się skorzystanie z narzędzi dostępnych w ramach standardowej licencji, w innym wypadku należy rozważyć zakup takich pakietów jak np. QTP.
- Koszty implementacji przypadków testowych
Opracowywanie przypadków testowych wymaga zarówno nakładu czasowego jak i wiedzy eksperckiej z zakresu wykorzystywanego narzędzia. Najczęściej wymagana jest również wiedza programistyczna żeby rozszerzać podstawowe przypadki testowe pozyskiwane za pomocą nagrywania o kod sprawdzający określone warunki. Jeżeli przypadek testowy będzie wykonywany małą liczbą razy nakład związany z jego automatyzacją może być większy niż nakład związany z jego ręcznym wykonaniem. Im większa jest szansa na ponowne użycie automatycznego przypadku testowego tym bardziej opłacalne jest jego przygotowanie.
- Koszty utrzymania przypadków testowych
Koszty te uwidaczniają się szczególnie w późniejszych etapach projektu bądź podczas użytkowania systemu kiedy wykonywane jest testowanie regresyjne. W przypadku zmiany wymagań skutkującej zmianą przebiegu procesu lub w przypadku zmiany działania standardowej aplikacji SAP skrypty testowe stają się nieaktualne i wymagają dostosowania. Tylko dzięki temu można zapewnić stałe pokrycie testami kluczowych funkcjonalności i bezbłędne wykonywanie się skryptów. Na koszty związane z utrzymaniem przypadków testowych istotny wpływ ma sposób ich przygotowania. Wprowadzenie standardów kodowania dla skryptów testowych, ujednolicenie konwencji nazewnictwa, czy przygotowanie repozytorium wzorcowych przypadków testowych dla typowych scenariuszy może te koszty znacząco obniżyć.

Przed podjęciem decyzji o wprowadzeniu automatyzacji testów do procesów związanych z prowadzeniem projektu należy rozważyć powyższe czynniki i skonfrontować je z dostępnym budżetem. Procesy biznesowe zawarte w koncepcji należy uszeregować względem istotności dla prawidłowego działania przedsiębiorstwa. Pozwoli to zidentyfikować krytyczne obszary i objąć je automatycznym testowaniem w pierwszej kolejności. Co w rezultacie przyczynia się do zapewnienia zgodności przebiegu tych

procesów z wymaganiami oraz zapewnienia ciągłości operacyjnej przedsiębiorstwa w przypadku wprowadzania zmian w systemie w późniejszym okresie.

Literatura

1. Błażewicz J., Jankowski M., Klaus R.: Organizacja przedsięwzięcia wdrożeniowego SAP R/3, Komputerowo Zintegrowane Zarządzanie pod redakcją Ryszarda Knosali, WNT W-wa 2003, tom 1, str. 93-100, ISBN 83-204-2809-2.
2. Ehrensperger W., Staader J.: New Dynamic Test Tools for ABAP Developers, SAP AG, 2004.
3. Helfen M., Lauer M. Trauthwein M.: Testing SAP Solutions, Gaileo Press, 2007.
4. Karch S., Heilig L.: SAP NetWeaver Roadmap, Gaileo Press, 2005.
5. Klaus R., Maćkowski T.: Koncepcje i narzędzia automatyzacji testowania jednostkowego i funkcjonalnego aplikacji SAP, Raport Instytutu Informatyki Politechniki Poznańskiej, 2008.
6. Maidstone J., Staader J.: Dynamic Testing Throughout the ABAP Development Lifecycle, SAP AG, 2004.
7. Samól D.: Transformacja systemu klasy ERP, 2003.
8. SAP01 - mySAP Solution Overview, SAP AG, 2005.
9. Standish Group CHAOS Report on IT projects, <http://www.standishgroup.com/>, 2006.
10. Mercury QuickTest Professional User's Guide Version 9.2, MERCURY, 2007.

D inż. Rafał KLAUS
Mgr inż. Tomasz MAĆKOWSKI
Instytut Informatyki
Politechnika Poznańska
60-965 Poznań ul. Piotrowo 2
e-mail: rafal.klaus@cs.put.poznan.pl