



AKADEMIA FINANSÓW i BIZNESU VISTULA

CENTRUM STUDIÓW PODYPLOMOWYCH i SZKOLEŃ

STUDIA PODYPLOMOWE

Testerzy Oprogramowania

Agnieszka Kondrat

PRACA DYPLOMOWA

SPECYFIKA TESTOWANIA OPROGRAMOWANIA

W PROJEKCIE OUTSOURCINGOWYM

Promotor: Radosław Smilgin

WARSZAWA 2015

Spis treści

WSTĘP	1
ROZDZIAŁ I TESTOWANIE OPROGRAMOWANIA	3
1.1. Czym jest testowanie?	3
1.2. Co to jest outsourcing?.....	10
1.2.1 Szanse i zagrożenia związane z outsourcingiem	11
1.2.2 Outsourcing zagraniczny	13
1.3. Testowanie w outsourcingu.....	14
1.3.1 Korzyści testowania w outsourcingu	15
1.3.2 Modele występujące w outsourcingu.....	15
1.3.3 Jakie procesy przekazać do outsourcingu?	17
1.3.4 Współpraca z wykonawcą	18
ROZDZIAŁ II SPECYFIKA TESTOWANIA NA PRZYKŁADOWYM PROJEKCIE OUTSOURCINGOWYM	27
2.1. Wprowadzenie do projektu	27
2.2. Zarys produktu oraz technologie użyte w projekcie	28
2.3. Cykl produkcji.....	28
2.4. Struktura zespołu projektowego.....	30
2.5. Środowisko testowe	31
2.6. Wykonywane testy	32
2.7. Aktywności i produkty	34
2.8. Współpraca z wykonawcą.....	35
2.9. Zaobserwowane ryzyka i zagrożenia	39
ZAKOŃCZENIE	41
BIBLIOGRAFIA	42
SPIS RYSUNKÓW	45

WSTĘP

Z dnia na dzień informatyzacja przybiera coraz większy, globalny zasięg. Aby systemy softwarowe spełniały swoją rolę, muszą działać zgodnie z tym, do czego zostały przeznaczone. Niestety często tak się nie dzieje, dlatego testowanie oprogramowania powinno być nierozłączną częścią procesu tworzenia aplikacji. Od systemów informatycznych może zależeć wiele, np. ludzkie życie lub sukces misji kosmicznej. Wystarczy wspomnieć kilka słynnych przypadków awarii, np. lot sondy Mariner 1 czy rakiety nośnej Ariane 5, aby zdać sobie sprawę, jak istotne jest przetestowanie każdego, nawet najmniejszego elementu struktury. Dzięki testowaniu możemy w dużym stopniu zweryfikować czy system spełnia zakładane wymagania oraz dodatkowo zminimalizować ryzyko wystąpienia awarii.

Rynek zaczyna rozumieć, że są pewne aktywności, z którymi ciężko jest się uporać samemu i w takich sytuacjach warto skorzystać z pomocy doświadczonych w tej materii firm trzecich. Dlatego też przekazanie odpowiedzialności testowania do specjalistycznej firmy staje się coraz bardziej popularną praktyką.

Punktem wyjścia niniejszej pracy jest poszukiwanie odpowiedzi na pytanie jak wygląda współpraca klienta zlecającego testowanie oraz firmy zajmującej się testowaniem w outsourcingu. **Celem pracy** natomiast była teoretyczna analiza zasad współpracy outsourcingowej począwszy od wybrania dostawcy, poprzez sprecyzowanie umów i kontraktów, aż do finalizacji współpracy oraz empirycznej diagnozy stanu powyższej współpracy na przykładowym projekcie.

Opisywany w niniejszej pracy projekt został użyczony do analizy i badań pod warunkiem zachowania anonimowości dotyczącej nazw organizacji w nim uczestniczących oraz nazwy projektu. Dodatkowym warunkiem udostępnienia informacji było przekazanie finalnej wersji pracy do wglądu organizacji zlecającej testowanie oraz firmy testującej produkt. Firma testująca dostarczyła autorce pracy potrzebne materiały i dane do przeprowadzenia analizy projektu, a w zamian oczekiwała otrzymania informacji, w jaki sposób może ulepszyć procesy i procedury, które są aktualnie wykonywane.

Autorka pracy podpisała klauzulę poufności, na podstawie, której zobligowana jest do nieujawniania wspomnianych nazw oraz danych osobowych, do których otrzymała dostęp.

Badania oraz zbieranie potrzebnych informacji w głównej mierze polegały na

obserwowaniu oraz rozmowach z uczestnikami procesu testowego. Badanie przeprowadzone zostało w biurze dostawcy odpowiedzialnego za testy i trwało ono 5 dni roboczych. Po tym okresie czasu na wszystkie nowo pojawiające się pytania, odpowiedzi od dostawcy testów były udzielane za pomocą poczty elektronicznej. Całkowity czas badania i gromadzenia potrzebnych do pracy danych trwał 10 dni roboczych.

Rozdział I pracy definiuje testowanie wraz z jego podziałem oraz ilustruje proces testowy wraz z jego produktami. Opisany został również outsourcing, jego modele, a także korzyści oraz zagrożenia, jakie może nieść ze sobą. Ukazane zostało teoretyczne spojrzenie na współpracę z wykonawcą wraz z istotnymi aspektami kontraktu.

W rozdziale II zaprezentowana została charakterystyka badanego projektu, wykorzystane technologie, przedstawiony cykl produkcji oraz środowisko testowe. Ponadto opisane zostały elementy współpracy klienta i dostawcy outsourcingowego na przykładzie konkretnego projektu. Celem tej części pracy było przedstawienie poziomu oraz jakości prowadzonej współpracy. Wskazane zostały również działania naprawcze mające na celu wyeliminowanie zdiagnozowanych problemów.

ROZDZIAŁ I

TESTOWANIE OPROGRAMOWANIA

1.1. Czym jest testowanie?

Testowanie oprogramowania we współczesnym świecie odgrywa coraz większą rolę. Czym jest więc testowanie i dlaczego jest takie ważne?

Systemy softwarowe spotykamy praktycznie wszędzie. Począwszy od aplikacji biznesowych, np. w bankowości lub przedsiębiorstwach, przez aplikacje publiczne, np. państwowe systemy wspierające redukcję bezrobocia, po produkty użytkowe, np. samochody. Niestety w wielu przypadkach systemy te nie działają prawidłowo. Defekty występujące w systemach, które nas otaczają, mogą prowadzić do wielu problemów, od straty pieniędzy lub utraty wizerunku biznesowego po utratę zdrowia lub życia.

Człowiek jest istotą omylną. Może popełnić błąd, którego efektem będzie powstanie defektu w kodzie programu lub w dokumencie. Wykonanie kodu zawierającego defekt może skutkować powstaniem awarii. System nie wykona tego co powinien lub wykona, ale w inny sposób niż powinien. Usterki, które znajdziemy w oprogramowaniu, systemie lub dokumencie, mogą powodować awarie, jednak nie każda usterka do tego prowadzi.

Jest wiele powodów prowadzących do powstania defektu. Główne powody wpływające na błędy popełniane przez człowieka:

- praca pod presją,
- zbliżające się terminy oddania pracy,
- praca ze skomplikowanym kodem w złożonej infrastrukturze,
- ciągle zmieniające się wymagania i technologie,
- duża interakcja między systemami.

Niekiedy awarie są uzależnione od warunków środowiskowych. Pole magnetyczne i elektryczne, promieniowanie lub zanieczyszczenia mogą prowadzić do awarii wbudowanego oprogramowania lub wpływać na oprogramowanie przez zmianę warunków działania sprzętu¹.

¹ Stowarzyszenie Jakości Systemów Informatycznych, *Certyfikowany tester, Plan poziomu podstawowego*, http://sjsi.org/wp-content/uploads/2013/08/sylabus-poziomu-podstawowego-wersja-2011.1.1_20120925.pdf [dostęp: 16.09.2015].

Testowanie oprogramowania to proces ściśle powiązany z wytwarzaniem oprogramowania. Profesjonalne, dobrze zaplanowane testowanie systemów oraz dokumentacji pozwala w znaczącym stopniu zmniejszyć ryzyko awarii systemu, jeśli znalezione usterki zostaną usunięte w procesie debugowania².

Głównym zadaniem testowania jest weryfikacja i walidacja oprogramowania. **Weryfikacja** kontroluje czy produkt wytwarzany jest prawidłowo, zgodnie ze sztuką tworzenia oprogramowania oraz czy jest zgodny ze specyfikacją. Natomiast **walidacja** pozwala ocenić czy produkt spełnia wymagania postawione przez klienta.

Proces testowania oprogramowania może być wdrożony w dowolnym momencie produkcji, aczkolwiek często jest uzależniony od konkretnej metody wytwarzania oprogramowania.

Testy oprogramowania można sklasyfikować na dwóch płaszczyznach. Testowanie jako proces można podzielić na etapy, gdzie każdy z nich ma własną specyfikę, cel, zakres, wymagania i obiekty, których dotyczy. Jeśli spojrzymy na testy oprogramowania jako działania, możemy tutaj wyodrębnić podział na typy testów.

Zgodnie z materiałami przygotowawczymi do najbardziej rozpoznawalnego w świecie IT certyfikatu ISTQB, a właściwie z sylabusem dla poziomu podstawowego (ISTQB CTFL)³, wyróżniamy podział na poziomy:

- testy modułowe,
- testy integracyjne,
- testy systemowe,
- testy akceptacyjne.

W wyżej wymienionym sylabusie podział na typy testów wygląda następująco:

- testowanie funkcji (testowanie funkcjonalne),
- testowanie atrybutów нефunkcjonalnych (testowanie нефunkcjonalne),
- testowanie struktury/architektury oprogramowania (testowanie strukturalne),
- testowanie związane ze zmianami: testowanie potwierdzające (akceptacyjne) oraz regresyjne.

² ROMAN A., *Testowanie i jakość oprogramowania. Modele, techniki, narzędzia*, PWN, Warszawa 2015, s. 4.

³ Stowarzyszenie Jakości Systemów Informatycznych, *Certyfikowany tester, Plan poziomu podstawowego*, http://sjsi.org/wp-content/uploads/2013/08/sylabus-poziomu-podstawowego-wersja-2011.1.1_20120925.pdf [dostęp: 16.04.2015].

Dokładniejsza analiza klasyfikacji testów w kontekście outsourcingowym została przedstawiona w dalszych podpunktach niniejszego rozdziału.

Opis procesu testowania oprogramowania został przedstawiony w sylabusach ISTQB. Niestety na chwilę obecną nie jest on spójny. Wspomniane sylabusy rozwijane są w sposób niezależny, dlatego pojawiły się rozbieżności. Grupy tworzące sylabusy próbują ujednoczyć oba dokumenty, mając cały czas na uwadze zmieniający się kontekst projektowy.

„Sylabus ISTQB Certyfikowany tester. Plan poziomu podstawowego. Wersja 2011.1.1” pochodzi z roku 2011, a „Sylabus ISTQB Poziomu Zaawansowanego Kierownik Testów” z roku 2012. Różnica wydaje się niewielka, jednak należy zwrócić uwagę na fakt, że prace nad drugim dokumentem trwały od roku 2007. W związku z tym przedstawienie procesu testowania w jednym i drugim dokumencie jest rozbieżne⁴.

Opis procesu w sylabusie poziomu podstawowego przedstawia rysunek 1.1., jest on podzielony na następujące fazy:

- planowanie i kontrola,
- analiza i projektowanie,
- implementacja i wykonanie,
- ocena kryterium zakończenia i raportowanie,
- czynności zamykające test

Rys. 1.1. Podział procesu testowego wg sylabusa poziomu podstawowego



Źródło: <http://testerzy.pl/baza-wiedzy/jak-sylabus-istqb-proces-testowy>, [dostęp: 16.09.2015].

⁴ <http://testerzy.pl/baza-wiedzy/jak-sylabus-istqb-proces-testowy>, [dostęp: 16.09.2015].

Schemat ten od momentu opublikowania był krytykowany za zbyt dużą kaskadowość oraz o to, że nie brał pod uwagę ogólnej specyfiki projektu informatycznego. Uważany był on za czysto teoretyczny, ponieważ łączył czynności, które mogły być realizowane przez różne role oraz w różnych etapach projektowych.

W sylabusie poziomu zaawansowanego pojawia się schemat zmodyfikowany. Nastąpiło rozdzielanie czynności, a etapowość wskazuje na możliwość wykorzystania go w projektach zwinnych. Schemat przedstawiony jest na rysunku 1.2. Obejmuje on następujące fazy:

- planowanie, monitorowanie i kontrola,
- analiza,
- projektowanie,
- implementacja,
- wykonanie,
- ocena kryterium zakończenia i raportowanie,
- czynności zamykające testy.

Rys. 1.2. Podział procesu testowego wg sylabusu poziomu zaawansowanego



Źródło: <http://testerzy.pl/baza-wiedzy/jak-sylabus-istqb-proces-testowy>, [dostęp: 16.09.2015].

Separacja faz daje możliwość przypisania do nich konkretnego właściciela i roli odpowiedzialnej za wykonanie działań. Produkty wejścia i wyjścia występujące w fazach są teraz łatwiejsze do zdefiniowania.

Do końca roku 2015 powinien zostać opublikowany nowy sylabus poziomu podstawowego i być może proces testowy zostanie teraz ujednolicony.

Uruchomienie procesu testowania powoduje powstanie produktów takich jak:

- plan testów,
- raport z testów,
- raport defektów,
- scenariusze i przypadki testowe.

Planowanie jest najważniejszym elementem procesu testowego. Plan testów jest dokumentem, który powstaje w początkowej fazie testowania. Jest to „dokument opisujący zakres, metody, zasoby oraz harmonogram zamierzonych czynności testowych. Określa – między innymi – elementy testowe, testowane cechy, zadania testowe, kto będzie te zadania wykonywał, stopień niezależności testerów, środowisko testowe, technikę projektowania testów oraz kryteria wejścia i wyjścia, przesłanki ich użycia, a także ryzyka wymagające ciągłego planowania. Jest to zapis procesu planowania testów”⁵.

Przygotowując plan testów powinno się bazować na wytycznych normy IEEE 829-2008.

Według wielu osób raportowanie testów jest mniej istotne od ich uruchomienia. Jednak warto zaznaczyć, że niedokładny i źle przygotowany raport kończący fazę poprawnie wykonanych testów, będzie skutkował tym, że praca testera pójdzie na marne.

Norma ISO 29119, która jest następczynią normy IEEE 829 sugeruje używanie dwóch typów raportów. Pierwszy to raport statusu testów (z ang. Test Status Report), który zazwyczaj wykonywany jest kilkakrotnie w procesie testowania. Pokazuje on aktualny status testów. Drugi to raport kończący testowanie (z ang. Test Completion

⁵ <http://sjsi.org/slowo/plan-testow/> [dostęp: 18.09.2015].

Report), który dostarczany jest na koniec fazy testowania. Podsumowuje on cały proces oraz zakres uruchomionych testów.

Istotnym produktem testowania jest raport defektów. Raport ten zawiera podsumowanie cyklu testowego w postaci listy zgłoszonych defektów. Zgodnie z dobrymi praktykami prawidłowo zaraportowany defekt powinien zawierać:⁶

- tytuł – powinien przekazywać informacje o tym, co się wydarzyło przy użyciu możliwie najprostszycy słów, bez przekazywania emocji,
- kroki reprodukcji – dokładnie opisane kroki jak odtworzyć defekt,
- środowisko – wszystkie elementy składające się na środowisko, które mogą w jakikolwiek sposób wpływać na poprawne lub niepoprawne zachowanie aplikacji np. przeglądarka, system operacyjny itp.,
- oczekiwane zachowanie kontra rzeczywistość – czyli co powinno się wydarzyć, a co tak naprawdę się wydarzyło,
- załączniki – wszelkie pliki pomocne w zobrazowaniu wystąpienia defektu, np. zrzut z okna przeglądarki, gdzie wystąpił defekt, mały plik (nie większy niż 10mb) zapisany w skompresowanym formacie, który nie utrudnia odczytu informacji.

We wstępnej fazie testów tworzone są scenariusze testowe po to by zweryfikować czy zaimplementowane funkcjonalności oprogramowania działają poprawnie, czy przepływ procesów biznesowych odbywa się we właściwy sposób. Przy tworzeniu scenariusza testowego testerzy często korzystają z pomocy klienta, zaangażowanych stron oraz deweloperów.

Scenariusze pomagają zrozumieć testerom jak oprogramowanie powinno działać w prawidłowy sposób. Testują one przepływy biznesowe w aplikacji, dzięki nim potrafimy wykryć defekty, które nie byłyby wykryte za pomocą innych typów testów.

Scenariusz testowy może być pojedynczym przypadkiem testowym lub może być całą serią przypadków, które następują po sobie. Jest jak krótka historia, która tłumaczy, w jaki sposób aplikacja będzie używana przez końcowego użytkownika⁷.

⁶<http://testerzy.pl/artykuly/raportowanie-defektow-dla-nie-testerow> [dostęp: 15.09.2015].

⁷<http://www.softwaretestingmentor.com/articles/what-is-scenario-testing/> [dostęp: 16.09.2015].

Scenariusze testowe są głównie wykorzystywane podczas testów funkcjonalnych. Jednak jest możliwość zastosowania ich również w innych sytuacjach, np. podczas testów integracyjnych z innymi systemami⁸.

Jeśli chodzi o przypadek testowy to według „Słownika wyrażen związanych z testowaniem” jest to „*zbiór danych wejściowych, wstępnych warunków wykonania, oczekiwanych rezultatów i końcowych warunków wykonania opracowany w określonym celu lub dla warunku testowego, jak wykonanie pewnej ścieżki programu lub zweryfikowanie zgodności z konkretnym wymaganiem*”⁹.

Dodatkowo wspomniany wyżej słownik podaje definicje dwóch rodzajów przypadków testowych:¹⁰

- „*przypadek testowy niskiego poziomu – przypadek testowy z konkretnymi (na poziomie implementacji) wartościami wejściowymi i wynikami oczekiwanymi. Logiczne operatory z przypadków testowych wysokiego poziomu są zamieniane na konkretne wartości, które odpowiadają celom logicznych operatorów*”,
- „*przypadek testowy wysokiego poziomu – przypadek testowy bez konkretnych (poziom implementacji) wartości danych wejściowych i oczekiwanych rezultatów. Używane są operatory logiczne; rzeczywiste wartości nie są jeszcze zdefiniowane i/lub dostępne*”. Przypadek testowy wysokiego poziomu nazywany jest również czasami abstrakcyjnym przypadkiem testowym.

Aby proces testowania przebiegł prawidłowo, grupie testerskiej muszą zostać dostarczone odpowiednie dokumenty. Do podstawowych dokumentów wspomagających proces testowania możemy zaliczyć:

- dokumentację użytkownika aplikacji/produktu,
- dokument ze sprecyzowanymi wymaganiami aplikacji/produktu,
- specyfikację i instrukcję obsługi produktu,
- w przypadku aplikacji wykonujących skomplikowane obliczenia i działania techniczne – dokument z danymi testowymi.

⁸<http://ideas2action.pl/2012/10/02/jak-pisac-scenariusze-testowe/> [dostęp: 16.09.2015].

⁹ Stowarzyszenie Jakości Systemów Informatycznych, *Słownik wyrażen związanych z testowaniem*, wersja 2.3 (2014), <http://sjsi.org/wp-content/uploads/2014/10/slovník-termin%C3%B3w-testowych-ver-2.3-PL.pdf> [dostęp: 05.05.2015].

¹⁰ Stowarzyszenie Jakości Systemów Informatycznych, *Słownik wyrażen związanych z testowaniem*, wersja 2.3 (2014), <http://sjsi.org/wp-content/uploads/2014/10/slovník-termin%C3%B3w-testowych-ver-2.3-PL.pdf> [dostęp: 05.05.2015].

Jeśli chcemy mieć pewność, że jesteśmy odpowiednio przygotowani do rozpoczęcia procesu testowania musimy dokładnie wiedzieć co i w jaki sposób chcemy przetestować. Należy stworzyć odpowiedni, dobry plan testów, który powinien zostać zaakceptowany przez klienta, a następnie stopniowo realizowany przez grupę testerską.

1.2. Co to jest outsourcing?

Ogólna definicja outsourcingu mówi, że jest to wydzielenie ze struktury organizacyjnej przedsiębiorstwa niektórych realizowanych przez nie samodzielnie funkcji i przekazanie ich do wykonania innym podmiotom¹¹.

Dlaczego w niektórych przypadkach należy stosować outsourcing? Cytat znanego, amerykańskiego przemysłowca Henry'ego Forda powinien dać nam na to pytanie odpowiedź:

“Jeśli jest coś, czego nie potrafimy zrobić wydajniej, taniej i lepiej niż konkurenci, nie ma sensu, żebyśmy to robili i powinniśmy zatrudnić do wykonania tej pracy kogoś, kto robi to lepiej niż my”¹².

Outsourcing pozwala wykonywać pracę wydajniej oraz skuteczniej. Dobrym przykładem jest zlecenie zarządzania naszymi pieniędzmi doradcom inwestycyjnym. Znają oni rynki finansowe lepiej niż większość menedżerów. Potrafią lepiej dobrać akcje do portfela oraz odpowiednio je wyceniać. Nawet, jeśli potrafimy wydajnie kupować i sprzedawać to nasza wiedza może okazać się niewystarczająca do podjęcia dobrych i trafnych decyzji. Zlecając pracę ekspertom korzystamy z ich doświadczenia i wiedzy¹³.

Przedsięwzięcie outsourcingowe wymaga przeniesienia czynników produkcji i zasobów potrzebnych do wykonania pracy oraz prawa do podejmowania decyzji bądź też odpowiedzialności decyzyjnej. Organizacja transferująca te czynniki i zasoby jest klientem. Organizacja, która podejmuje decyzję i wykonuje prace jest dostawcą, a zakres prac zostaje zdefiniowany w formie projektu¹⁴.

Najczęstszym przedmiotem outsourcingu IT jest programowanie, jednak nic nie

¹¹ TROCKI M., *Outsourcing*, PWE, ISBN 83-208-1340-9, Warszawa 2001, s. 13.

¹² https://pl.wikiquote.org/wiki/Henry_Ford [dostęp: 30.07.2015].

¹³ POWER M.J., DESOUZA K.C., BONIFAZI C., *Outsourcing. Podręcznik sprawdzonych praktyk*, Warszawa 2010, s. 19.

¹⁴ POWER M.J., DESOUZA K.C., BONIFAZI C., *Outsourcing. Podręcznik sprawdzonych praktyk*, Warszawa 2010, s. 21.

stoi na przeszkodzie, aby testowanie również zostało przekazane na zewnątrz. Przed outsourcingiem testowania postawione zostały takie same wyzwania jak w przypadku outsourcingu programowania. Sukces w obydwu przypadkach uzależniony jest od solidnej podstawy, jaką są jasne wymagania¹⁵.

Czynnikiem decydującym o wdrożeniu outsourcingu w firmie powinna być chęć uzyskania lepszego produktu, a nie szukanie oszczędności.

1.2.1 Szanse i zagrożenia związane z outsourcingiem

Korzyści, które można osiągnąć dzięki prawidłowemu zastosowaniu outsourcingu:¹⁶

- dostęp do wysoko wykwalifikowanej i wyspecjalizowanej kadry testerskiej,
- uzyskanie zasobów, którymi przedsiębiorstwo nie dysponuje,
- po zakończeniu projektu odchodzi problem związany z rozwiązaniem umowy z zespołem,
- obniżenie kosztów stałych w przedsiębiorstwie, czyli pensje pracowników, powierzchnia biurowa, urządzenia,
- odchodzi część kosztów związanych z zarządzaniem personelem,
- możliwość skupienia się na podstawowej działalności firmy, np. firma księgowa skupia się głównie na rachunkowości, a stworzenie i przetestowanie aplikacji, na którą ma zapotrzebowanie zleca firmom outsourcingowym,
- dokładniejsze zaplanowanie budżetu, pod warunkiem, że kontrakt outsourcingowy będzie miał ustaloną wartość oraz nie będzie on renegotjowany w trakcie obowiązywania,
- obniżenie ryzyka niepowodzenia projektu, dzięki współdzieleniu odpowiedzialności klienta oraz dostawcy,
- własne zasoby możemy wykorzystać do innych celów,
- uporanie się z funkcją trudną do wykonywania lub niemożliwą do kontrolowania.

Wdrożenie outsourcingu jest procesem skomplikowanym i trudnym. Według raportów opublikowanych przez Gartnera, prawie 80% kontraktów outsourcingowych

¹⁵ WIEGERS K., BEATTY J., *Specyfikacja oprogramowania, Inżynieria wymagań*, Helion, Gliwice 2014, s. 430.

¹⁶ <https://pl.wikipedia.org/wiki/Outsourcing> [dostęp: 31.07.2015].

podlega renegotjacji. Spowodowane jest to przede wszystkim potrzebami klienta, które zmieniają się w trakcie trwania projektu. Jeden na osiem kontraktów zostaje renegotjowany w przeciągu pierwszych 12 miesięcy trwania projektu¹⁷.

Główne zagrożenia wynikające z zastosowania outsourcingu:¹⁸

- brak spodziewanej redukcji kosztów,
- obniżka jakości funkcji, które były dotychczas realizowane samodzielnie,
- współpraca z dostawcą może zakończyć się niepowodzeniem,
- konflikty na linii klient-dostawca, dotyczące głównie zastrzeżeń co do jakości oraz kosztów,
- szanse uzyskane z większej operacyjnej elastyczności mogą zostać niewykorzystane podczas zaspokojenia potrzeb klientów.

Podjmując decyzję o oddelegowaniu części pracy na zewnątrz, należy również zwrócić szczególną uwagę na dziesięć pułapek outsourcingu, które możemy napotkać podczas jego stosowania:¹⁹

1. Brak zaangażowania menedżerów.
2. Brak planu komunikacji.
3. Zbyt mała wiedza o metodach outsourcingu.
4. Nierozpoznanie ryzyka biznesowego.
5. Niekorzystanie z zewnętrznych źródeł wiedzy.
6. Nieangażowanie w projekt najlepszych menedżerów.
7. Nadmierny pośpiech we wdrażaniu projektu.
8. Niedocenienie różnic kulturowych.
9. Ignorowanie czynników decydujących o produktywności dostawcy.
10. Złe zarządzanie relacjami.

W dzisiejszym świecie outsourcing jest strategicznym narzędziem biznesowym. Wiele firm nie wyobraża sobie funkcjonowania bez niego. Patrząc na jego zalety i wady, musimy wyobrazić sobie jak przełożą się one na naszą działalność. Musimy

¹⁷ <http://www.zdnet.com/article/gartner-highlights-outsourcing-problems> [dostęp: 02.08.2015].

¹⁸ <https://pl.wikipedia.org/wiki/Outsourcing> [dostęp: 02.08.2015].

¹⁹ http://webdb.ucs.ed.ac.uk/operations/stratops/papers/Mark_power_2004.pdf [dostęp: 02.08.2015].

odpowiedzieć sobie na pytania typu: Czy dostarczy on nam dużo korzyści? Czy ryzyko nie będzie za duże? Czy możemy zgodzić się na utratę kontroli nad pewnymi elementami? Jeśli odpowiemy sobie na te pytania, decyzja o wprowadzeniu outsourcingu nie powinna być trudna.

1.2.2 Outsourcing zagraniczny

Firmy szukając oszczędności często decydują się na outsourcing. Aby korzyści finansowe były jeszcze bardziej doraźne, w niektórych przypadkach wybierany jest outsourcing zagraniczny. W krajach wysokorozwiniętych koszt roboczogodziny dla specjalistów IT ustalony jest na dość wysokim poziomie. Przenosząc część obowiązków do krajów o niskich kosztach pracy, takich jak Indie, Chiny, Filipiny, kraje Ameryki Łacińskiej, Polska,²⁰ czy w ostatnim czasie Rumunia²¹, można spodziewać się znaczącej optymalizacji kosztów.

Mimo dużego potencjału i możliwości produkcyjnych, klienci ostatnio rezygnują ze współpracy z firmami outsourcingowymi z krajów Afryki Północnej, w szczególności Egiptu. Przyczyną tego jest niestabilna sytuacja geopolityczna oraz częste zamachy terrorystyczne²².

Czynniki decydujące o wyborze konkretnego kraju na dostawcę mogą być różnorakie. Niektórzy klienci szukają dostawców w krajach o zbliżonej kulturze, dla innych może mieć wpływ lokalizacja, która ma znaczący wpływ na koszty oraz zmęczenie pracownika podczas częstych podróży służbowych. Najczęściej jednak o wyborze decydują najniższe koszty, które niestety często nie idą w parze z zadowalającą jakością produktu czy usługi.

Jeśli zdecydujemy się zlecić testowanie zagranicznym firmom outsourcingowych, to musimy być świadomi niebezpieczeństw, jakie możemy napotkać.

Należy zwrócić uwagę na strefy czasowe klienta i dostawcy, które mogą się różnić porą dnia. Może to spowodować utrudnioną i opóźnioną komunikację.

Bariery językowe mogą powodować problemy z prawidłową interpretacją wymagań. Podczas tłumaczenia dostawca może interpretować je bardzo dosłownie i nie

²⁰ http://www.tholons.com/nl_pdf/Whitepaper_December_2013.pdf [dostęp: 03.08.2015].

²¹ <http://www.zdnet.com/article/it-outsourcing-as-romania-vies-to-be-the-new-india-can-the-country-keep-up/> [dostęp: 03.08.2015].

²² <https://clutch.co/outourcing-location/egypt> [dostęp: 03.08.2015].

dostrzeże niuansów znaczeniowych.

Innym istotnym czynnikiem mogącym zaburzyć współpracę są różnice kulturowe. Osoby pochodzące z pewnych kręgów kulturowych mają problemy z krytyką pracy wykonanej przez kogoś innego, nawet, jeśli jest to krytyka konstruktywna. Natomiast, jeśli komentarze są skierowane do pracy takich osób, mogą oni przyjąć to bardzo osobiście. Niestety efektem tego może być brak reakcji na niepokojące rzeczy, aby nie urazić autora przeglądanych materiałów. Zachowanie takie jest przejawem grzeczności i szacunku, jednak nie przysłuży się do osiągnięcia wspólnego celu²³.

Jeśli po stronie klienta osobą zarządzającą projektem jest kobieta, a dostawcą jest firma zlokalizowana w krajach arabskich, współpraca może okazać się niemożliwa do zrealizowania. Mężczyźni pracujący w firmie dostawcy mogą nie zaakceptować takiego stanu rzeczy²⁴.

Klient decydując się na outsourcing, często ma możliwość wyboru osób, które będą realizować projekt. Podczas doboru odpowiednich, zagranicznych kandydatów należy pamiętać o różnicach w wykształceniu oraz tytułach, jakimi się legitymują. Może zdarzyć się, że kandydat posiada tytuł, który brzmi jak magister, a w rzeczywistości może okazać się prostym certyfikatem²⁵.

Decydując się na outsourcing zagraniczny, jeśli chcemy, aby był on efektywny i spełnił pokładane w nim nadzieje, należy się zapoznać z kulturą pracy, religią i obyczajami kraju dostawcy. Pozwoli nam to na łatwiejszą komunikację, efektywniejszą współpracę oraz pozwoli uniknąć nieporozumień i nietaktownych sytuacji.

1.3. Testowanie w outsourcingu

Outsourcing testowania jest bardzo szeroką tematyką, tak szeroką jak wiele jest działań związanych z testowaniem, które mogą zostać wykonane przez osoby trzecie, od strategii testów, po testy penetracyjne. Testowanie ma coraz większą rolę w procesach tworzenia produktu. Wskazywać może na to fakt, że outsourcingiem testowania zajmują się nie tylko niezależne firmy testerskie, ale do swej oferty wprowadziły go również duże,

²³ WIEGERS K., BEATTY J., *Specyfikacja oprogramowania, Inżynieria wymagań*, Helion, Gliwice 2014, s. 429-433.

²⁴ http://www.stickyminds.com/sites/default/files/presentation/file/2013/11STRWR_T6.pdf [dostęp: 06.08.2015].

²⁵ http://www.stickyminds.com/sites/default/files/magazine/file/2012/XDD12261filelistfilename1_0.pdf [dostęp: 06.08.2015].

odgrywające istotną rolę na rynku firmy outsourcingowe.

Omawiając specyfikę testowania w outsourcingu, należy pamiętać o tym, że większość, jeśli nie wszystkie z zasad, które odnoszą się do zwykłego outsourcingu, mogą znaleźć zastosowanie również tutaj, tak więc muszą być brane pod uwagę.

1.3.1 Korzyści testowania w outsourcingu

Poprzez outsourcing testowania mogą być osiągnięte następujące korzyści:²⁶

- przyspieszenie procesu testowania – jeśli proces programowania jest opóźniony, aby w pełni wykorzystać dostępny czas i przygotować się do testowania, potrzeba łatwo dostępnego zespołu ludzi. Wesprzeć może nas tutaj firma outsourcingowa, która dostarczy potrzebne zasoby. W niektórych przypadkach, testowanie może zostać przyspieszone poprzez wykorzystanie różnic stref czasowych. Np. programiści mogą kodować aplikację w dzień, a następnie testerzy w innej strefie czasowej będą przeprowadzać testy w nocy,
- podniesienie jakości – konieczność przestrzegania formalnego procesu testowania i dostarczenia właściwej podstawy do testów w postaci dobrze udokumentowanych i stabilnych wymagań, w rezultacie mogą skutkować znalezieniem większej ilości defektów, co z kolei wpływa na poprawę jakości produktu końcowego. Ponadto, niektóre defekty zostaną zlokalizowane wcześniej, podczas analizy wymagań lub przeglądu przypadków testowych,
- redukcja kosztów testowania – głównym czynnikiem wpływającym na wybranie outsourcingu jest często redukcja kosztów, poprzez przeniesienie działań do krajów o niższych kosztach pracy. Zarówno testowanie, kosztowne IT i zasoby biznesowe mogą zostać uwolnione od powtarzalnych i czasochłonnych aktywności testowych, jakimi są np. testy regresyjne. Podniesienie jakości oprogramowania również prowadzi do oszczędności.

1.3.2 Modele występujące w outsourcingu

Podczas stosowania outsourcingu w testowaniu, możemy wyodrębnić kilka typów modeli, podzielonych w zależności od ich charakteru.

Klient wybierający outsourcing decyduje, jakie procesy zostaną przekazane

²⁶ <http://www.methodsandtools.com/archive/archive.php?id=74> [dostęp: 10.08.2015].

dostawcy do realizacji. W związku z tym w outsourcingu IT mamy do czynienia z następującymi modelami:

- pełny outsourcing IT – polega na całkowitym powierzeniu firmie zewnętrznej gruntownego przetestowania aplikacji,
- selektywny outsourcing IT – powierzenie firmie zewnętrznej przetestowania części aplikacji. Np. testy systemowe można powierzyć dostawcy, a testy akceptacyjne wykonać we własnym zakresie,
- outsourcing IT na życzenie – jest to stosunkowo nowy model, gdzie główne wewnętrzne procesy operacyjne firmy, zostają przeniesione na dostawcę. Jest on rozliczany na podstawie dostarczonych usług²⁷.

W outsourcingu, realizacja powierzonych zadań odbywa się poza siedzibą firmy klienta. Możemy więc wyodrębnić podział na model miejsca realizacji:

- zespół outsourcingowy pracuje w siedzibie swojej firmy, która jest wyposażona w odpowiednią infrastrukturę IT,
- zespół outsourcingowy jest rozproszony, realizując zadania zdalnie.

Jednym z głównych celów outsourcingu jest obniżenie kosztów. W związku z tym klient, decydujący się na outsourcing, może wybrać sposób rozliczenia z dostawcą dostosowany do swoich potrzeb. Najczęściej występujące modele cenowe:

- projekty o określonej, stałej cenie (z ang. fixed price) – ten typ modelu jest idealnym mechanizmem cenowym dla projektów, które mają jasny cel, ustalone metody zarządzania projektem oraz stabilny zestaw wymagań²⁸,
- projekty rozliczane według czasu pracy i zużytych zasobów (z ang. time & materials) – model ten jest popularnym modelem cenowym, najczęściej stosowanym w sektorze IT w projektach dotyczących rozwoju i utrzymania aplikacji, które są zaplanowane na dłuższy okres czasu. Umowy tego typu wymagają od usługodawców składania ofert na projekt na podstawie wymagań projektu klienta, zakresu działania, ilości pracy oraz stopnia pokrycia²⁹,

²⁷ https://en.wikipedia.org/wiki/On-demand_outsourcing [dostęp: 11.08.2015].

²⁸ <http://outsourcing-journal.org/outsourcing-management/management-transformation/849-outsourcing-pricing-models-part-2-fixed-price-fp-model> [dostęp: 11.08.2015].

²⁹ <http://outsourcing-journal.org/outsourcing-management/management-transformation/855-outsourcing-pricing-models-part-4-time-and-materials-t-m-model> [dostęp: 13.08.2015].

- projekty, za które zwracane są poniesione koszty (z ang. cost-reimbursable) – model gdzie dostawca usług dostaje zwrot poniesionych kosztów w ramach ustalonych limitów oraz dodatkową płatność, która jest jego przychodem³⁰.

1.3.3 Jakie procesy przekazać do outsourcingu?

Wydzielając do firmy zewnętrznej aktywności testowe, które pochłaniają zasoby i czas, nie tylko uzyskujemy większą wydajność, skalowalność oraz elastyczność, ale również mamy dodatkowe korzyści w postaci niezależnej weryfikacji. Testowanie nie jest wykonywane tylko przez zespół programistów tworzących oprogramowanie, lecz także przez zewnętrzny zespół testowy.

Kiedy decydujemy, które części testowania chcemy wydzielić i przekazać na zewnątrz, musimy spojrzeć na testowanie z różnych perspektyw. W szczególności musimy zwrócić uwagę na poziomy testowania, typy testów oraz aktywności testowe:³¹

- poziomy testowania – testowanie niskiego poziomu, czyli testy modułowe i integracyjne są wykonywane przez zespół programistów. Jeśli proces wytwarzania produktu jest oddelegowany na zewnątrz, aktywności te są wykonywane razem z tym procesem. Testy systemowe są idealnym kandydatem do wykonywania przez firmę outsourcingową. Z kolei testy akceptacyjne wymagają znajomości biznesowego “know-how”³² oraz środowiska testowego zbliżonego maksymalnie do środowiska produkcyjnego. Testy na tym poziomie nie są łatwe do realizacji dla firmy zewnętrznej.
- typy testów – ogólnie mówiąc, wszystkie typy testów zarówno funkcjonalne jak i niefunkcjonalne mogą być przedmiotem outsourcingu. Testy regresyjne, zgodnie z ich naturą, są idealnym kandydatem, ponieważ są to testy cykliczne i zautomatyzowane. Testy obciążeniowe i wydajnościowe oraz użyteczności i bezpieczeństwa też najlepiej zlecić specjalistycznej firmie, która specjalizuje się w tych konkretnych typach.
- aktywności – szukając odpowiedzi na pytanie: Jakie aktywności związane z procesem testowania powinniśmy zlecić firmie zewnętrznej (np. planowanie testów, specyfikacja, wykonanie testów i raportowanie)?, musimy podjąć strategiczną decyzję jak dużo kontroli i wiedzy chcemy oddać w obce ręce. Możemy przekazać cały proces

³⁰ <http://outsourcing-journal.org/outsourcing-management/management-transformation/851-outsourcing-pricing-models-part-3-cost-reimbursable-model> [dostęp: 13.08.2015].

³¹ <http://www.methodsandtools.com/archive/archive.php?id=74> [dostęp: 16.08.2015].

³² <https://pl.wikipedia.org/wiki/Know-how> [dostęp: 16.08.2015].

testowania, bądź zlecić samo wykonanie testów.

Musimy zdecydować nie tylko o tym, które elementy projektu nadają się do outsourcingu, ale również o tym czy sam projekt jest odpowiednim kandydatem. Jeśli projekt jest zupełnie nową aplikacją, istnieje prawdopodobieństwo, że nie jest to idealny kandydat. Natomiast, jeśli projekt jest kolejną wersją stabilnego produktu, który ma zdefiniowane przypadki testowe, może być on zdecydowanie lepszym kandydatem. Tak więc, decydując się na outsourcing testowania projektu, musimy przeanalizować kilka czynników: złożoność techniczną projektu, wewnętrzne "know-how", dojrzałość wymagań i przypadków testowych oraz dostępność danych testowych³³.

1.3.4 Współpraca z wykonawcą

Jak zarządzać procesami przekazanymi do outsourcingu?

Stosowanie outsourcingu nigdy nie jest łatwym zadaniem. Czasami cały wysiłek idzie na marne, ponieważ outsourcing jest niedokładnie zarządzany. Oczekiwane założenia mogą być nierealne do spełnienia, można dokonać złego wyboru partnera biznesowego, mogą pojawić się problemy z komunikacją. Wszystkie te problemy mogą prowadzić do niepowodzenia. Konieczne jest zatem, aby dobrze zaplanować i przemyśleć wszystkie procedury, korzystać z najlepszych praktyk oraz być świadomym poziomu ryzyka.

Na początku musimy się zastanowić jak duży wysiłek przyjdzie nam włożyć w zastosowanie outsourcingu. Poza decyzją o tym, co chcemy przekazać do outsourcingu, musimy zdać sobie sprawę z dojrzałości naszego procesu testowego. Outsourcing, rzadko kiedy jest rozwiązaniem na problemy z wewnętrznymi procesami. Wręcz przeciwnie, gdy problem przekazany do outsourcingu pozostaje nierozwiązany, może stać się on dla nas dużo poważniejszy.

Przechodząc po kolei przez wszystkie fazy (rys. 1.3.), na początku musimy dokonać orientacyjnej kalkulacji dotyczącej naszego przypadku biznesowego. Rozważyć różne opcje outsourcingu, od zatrudnienia zewnętrznych specjalistów w dziedzinie testowania, do utworzenia własnego centrum zagranicznego (z ang. off-shore). Wszystko po to, aby znaleźć najbardziej optymalną kosztowo alternatywę.

Gdy podejmiemy już decyzję o wprowadzeniu outsourcingu, możemy zacząć

³³ <http://www.methodsandtools.com/archive/archive.php?id=74> [dostęp: 16.08.2015].

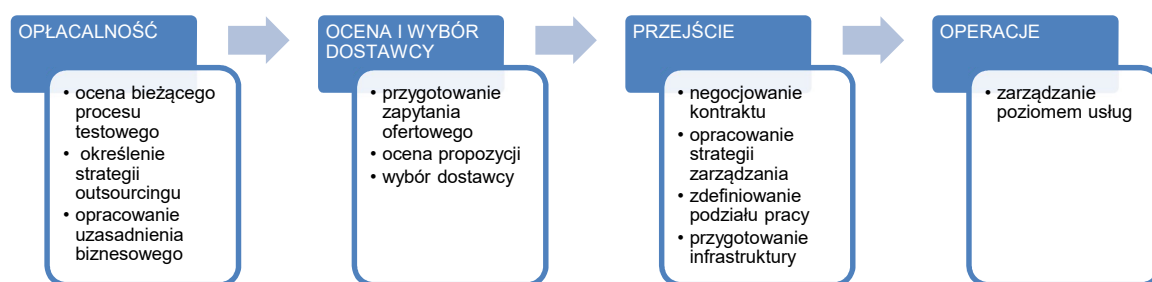
szukać odpowiedniego partnera. Ważnym jest, aby podczas poszukiwań nie patrzeć tylko na doświadczenie partnera w testowaniu, ale również na jego znajomość produktu oraz wiedzę technologiczną. Istnieje duże prawdopodobieństwo, że dzięki temu nasza współpraca będzie sprawniejsza oraz bardziej owocna. Jeśli już znajdziemy odpowiedniego kandydata, możemy go poprosić o złożenie oferty, która będzie w stanie zrealizować nasze potrzeby.

Dobre praktyki mówią, aby nie pozwolić lisowi pilnować kurnika, czyli innymi słowy nie powierzać testowania tej samej firmie, która zajmuje się produkcją danego oprogramowania. Sytuacja taka jest oczywiście korzystna dla dostawcy, który może wpływać na wyniki testowania tak, aby przedstawiały tworzone przez niego oprogramowanie w lepszym świetle^{34 35}.

Wybór dostawcy będzie wstępem do fazy przejścia. Faza ta obejmuje cały proces przygotowawczy, tj. przygotowanie dokumentacji, wzajemne poznanie się partnerów biznesowych. Podczas, gdy prawnicy będą przygotowywać dokumenty, można rozpocząć definiowanie zarządzania oraz rozdzielić pracę, a także skonfigurować infrastrukturę.

Jeśli wszystko będzie realizowane zgodnie z zakładanym planem, faza przejścia doprowadzi nas do stanu stabilności, zazwyczaj dzieje się to w przeciągu 6-9 miesięcy³⁶.

Rys. 1.3. Fazy wdrożenia outsourcingu



Źródło: <http://www.methodsandtools.com/archive/archive.php?id=74> [dostęp: 18.08.2015].

³⁴ <http://www.businesswire.com/news/home/20050525005488/en/Knowledge-Transfer-Objective-Evaluation-Successful-Outsourcing-Engagements#.VdAzk3WUfrc> [dostęp: 17.08.2015].

³⁵ <http://www.sourcingmag.com/9-tips-about-outsourcing-your-software-testing/> [dostęp: 17.08.2015].

³⁶ <http://www.methodsandtools.com/archive/archive.php?id=74> [dostęp: 18.08.2015].

Wszystkie kluczowe składniki sukcesu w codziennych operacjach są wynikiem efektywnego zarządzania poziomem usług. Należy zidentyfikować kluczowe wskaźniki wydajności oraz ustalić poziom usług, który chcemy osiągnąć. Warto, w umowie o gwarantowanym poziomie świadczenia usług (SLA³⁷), zdefiniować odpowiednie i dokładne miary. Poza tym należy pamiętać o standardowych metrykach, takich jak dostawa na czas, precyzja estymacji, retencja personelu oraz o metrykach związanych z testami³⁸.

Produkty testowania w outsourcingu

Celem każdego procesu outsourcingowego jest dostarczenie klientowi zamówionego produktu lub usługi. W przypadku zleconego testowania, produktem wynikowym są raporty z testów oraz raporty defektów, które określają, na jakim poziomie jest jakość dostarczonego do testów oprogramowania. Raporty te są przekazane klientowi. Oprócz raportów w procesie testowania pojawiają się jeszcze inne produkty:

- plan testów,
- przypadki testowe,
- bazy danych z przykładowymi danymi testowymi,
- skrypty automatyczne oraz programy,
- narzędzia zaprojektowane dla testów, np. narzędzie dla testów obciążeniowych.

O tym, które z powyższych pozycji będą stanowić dostawę dla klienta oraz kto będzie posiadał prawa do tych produktów, powinny zdecydować zapisy w kontrakcie.

Monitorowanie procesu outsourcingowego

Kiedy po wstępnej fazie wdrażania projektu, który wymagał baczego śledzenia i koordynacji, relacje outsourcingowe się normalizują, należy przejść do rutynowego i zautomatyzowanego procesu monitorowania. Gdyby klient nieustannie musiał patrzeć dostawcy na ręce i angażować w to cenny czas swojego zespołu, to projekt outsourcingowy nie miał by zbyt dużego sensu. Sukcesy może przynieść tylko skuteczne i wydajne monitorowanie współpracy³⁹.

³⁷ https://pl.wikipedia.org/wiki/Service_Level_Agreement [dostęp: 18.08.2015].

³⁸ <http://www.methodsandtools.com/archive/archive.php?id=74> [dostęp: 18.08.2015].

³⁹ POWER M.J., DESOUZA K.C., BONIFAZI C., *Outsourcing. Podręcznik sprawdzonych praktyk*, Warszawa 2010, s. 175.

Istotne elementy, które warto nadzorować podczas zlecenia testowania do outsourcingu:⁴⁰

- powinno się replikować wszystkie defekty, które zostały zgłoszone przez dostawcę, aby upewnić się czy są one uzasadnione,
- warto również wykonać audyt planu testów tak, aby odnaleźć ewentualne braki,
- monitorowanie reakcji programistów na zgłoszone defekty,
- monitorowanie postępu prac w stosunku do harmonogramu,
- monitorowanie postępu testów w stosunku do reszty projektu, czy testowanie przebiega szybciej lub wolniej niż pozostałe procesy,
- jeśli w dostawie do klienta znajdują się testy automatyczne, należy sprawdzić czy będą one przydatne w późniejszych iteracjach. Pracownicy powinni mieć na tyle dużo informacji, aby w łatwy sposób móc użyć ich ponownie,
- warto weryfikować czy nie ma za dużych rotacji personalnych po stronie wykonawcy⁴¹.

Proces zarządzania zmianami

Podczas współpracy biznesowej często dochodzi do zmian wymagań, które jeśli są źle zarządzane mają olbrzymi wpływ na czas, jakość oraz poniesione w związku ze zmianami koszty dostawcy.

Aby wspomóc wdrażanie zmian warto pomyśleć o kompleksowym systemie zarządzania zmianami. Na początku projektu opracować możliwy do przyjęcia przez obie strony proces sprawowania kontroli nad wprowadzaniem zmian. Proces ten będzie korzystny dla wszystkich uczestników projektu. Należy wyznaczyć osoby podejmujące decyzje dotyczące proponowanych zmian oraz opracować mechanizmy, z których będzie można korzystać, aby dać gwarancję dotarcia właściwych informacji do właściwych osób⁴². Przykład typowego scenariusza wygląda tak. Klient żąda jednej lub więcej zmian w projekcie. Przekazuje ją do dostawcy, który analizuje jak wszystkie zmiany wpłyną na terminy, zasoby i koszty oraz ocenia możliwość realizacji z technicznego punktu widzenia. Po tej kompleksowej ocenie zmian, informacja zwrotna trafia do klienta, który decyduje czy zmiana ma zostać wdrożona.

⁴⁰ <http://kaner.com/pdfs/outsource.pdf> [dostęp: 22.08.2015].

⁴¹ <http://conference.eurostarsoftwaretesting.com/wp-content/uploads/cte2.pdf> [dostęp: 22.08.2015].

⁴² WIEGERS K., BEATTY J., *Specyfikacja oprogramowania, Inżynieria wymagań*, Helion, Gliwice 2014, s. 433.

Umowa SLA

Umowa o gwarantowanym poziomie świadczenia usług powinna definiować minimalny poziom usług, jaki powinien być dostarczony przez firmę outsourcingową. Umowa powinna być obiektywna, mierzalna oraz nie powinna zawierać dwuznaczności. Pozwoli to na udaną, długoterminową współpracę.

Podczas przygotowywania SLA warto zwrócić uwagę na kilka istotnych elementów, które może zawierać umowa:⁴³

- dostawa na czas – dla wszystkich głównych dostaw terminy powinny być ustalone już na początku oraz powinny być dołożone wszelkie starania, aby zostały one dotrzymane. Jeśli terminy muszą być przesunięte należy wykorzystać do tego procesy kontroli zmian,
- satysfakcja klienta – powinny być prowadzone okresowe badania po to, aby upewnić się, że usługi dostarczane przez firmę outsourcingową są satysfakcjonujące dla klienta,
- skuteczność – metryki opisujące skuteczność to przede wszystkim metryki dotyczące obniżenia kosztów, zwiększenia zysku oraz poziomu dostosowywania transakcji handlowych,
- nakład pracy – nakład pracy jest często trudny do zdefiniowania. Dla projektów rozliczanych w trybie czasu pracy i zużytych materiałów nakład może być definiowany jako ilość wykorzystanych zasobów. Dla projektów o stałej cenie nakład może być określony poprzez liczbę dostaw. Metryka ta jest istotną częścią umowy SLA,
- wrażliwość – metryka wrażliwości określa ile czasu potrzebuje dostawca, aby obsłużyć żądanie klienta,
- przestoje systemu oraz zapewnienie dostępności – w outsourcingu zapewnienie 100% dostępności usług kosztuje znacznie więcej niż zagwarantowanie 99% czy 98% dostępności. Nie każda firma lub aplikacja wymaga 100% niezawodności. SLA powinna żądać takiej dostępności, aby specyficzne biznesowe potrzeby zostały spełnione.

Dobrze jest również dopilnować, aby umowa SLA, była powiązana z kontraktem na zasadzie ryzyka i zysku⁴⁴, tak by obie strony dążyły do jej prawidłowej realizacji⁴⁵.

⁴³ <http://www.tutorialspoint.com/white-papers/8.html> [dostęp: 19.08.2015].

⁴⁴ <http://financial-dictionary.thefreedictionary.com/Risk-Reward+Basis> [dostęp: 19.08.2015].

⁴⁵ <http://www.tutorialspoint.com/white-papers/8.html> [dostęp: 19.08.2015].

Wymagania i dokumentacja

Jeśli dokumentacja projektu do tej pory opierała się o nieformalne materiały, takie jak specyfikacja, plany testów itp., trzeba być świadomym, że firma outsourcingowa będzie potrzebować więcej detali. Opisane pobieżnie wymagania, które sprawdzają się w projektach własnych mogą okazać się niewystarczające w outsourcingu, ponieważ analitycy biznesowi nie mogą bezpośrednio odpowiadać na pytania, dzięki którym można by rozwiązać wątpliwości oraz uzupełnić luki. Niedostatecznie zdefiniowane wymagania są częstą przyczyną porażek projektów, których realizację zlecono na zewnątrz⁴⁶.

Jeśli oczekujesz wysokiej jakości zleconych usług, być może będzie trzeba zmienić praktyki i nawyki w firmie.

Pracownicy klienta często rozumieją wymagania dużo lepiej niż dostawcy. Dzieje się tak zazwyczaj w skomplikowanych projektach technicznych. Aby rozwiązać taką sytuację, warto w początkowej fazie trwania projektu nawiązać bliższą współpracę. Można spróbować połączyć na jakiś czas grupy klienta i dostawcy tak, aby przepływ wiedzy był skuteczniejszy i natychmiastowy.

Wiedza odnośnie produktu jest bardzo istotna dla testera. Nawet, jeśli tester posiada bardzo duże umiejętności automatyzacji testów, to bez odpowiedniej wiedzy nie wykorzysta ich w stu procentach. Dzięki wiedzy o produkcie będzie w stanie pokryć testami większy obszar aplikacji, bazując na przepływach biznesowych. Dobre testy automatyczne i regresyjne mogą powstać tylko w oparciu o dobrą wiedzę odnośnie produktu. Aby nabyć wiedzę o produkcie tester musi zrozumieć pewne kwestie, takie jak:⁴⁷

- Kim jest użytkownik?
- Jak użytkownicy chcą używać produktu?
- Czego można się spodziewać po projekcie, a czego nie?
- Jak zostały zrealizowane podobne produkty na rynku?

Warto zaznaczyć, że graficzne modele wymagań, znacząco poprawiają ich odbiór przez zespoły outsourcingowe. Korzystanie z modeli jest jeszcze bardziej korzystne

⁴⁶ WIEGERS K., BEATTY J., *Specyfikacja oprogramowania, Inżynieria wymagań*, Helion, Gliwice 2014, s. 429-430.

⁴⁷ <http://testhuddle.com/forums/topic/how-important-is-for-a-tester-to-know-the-characteristics-of-a-product/> [dostęp: 21.08.2015].

w przypadku, gdy współpracuje się z firmami zagranicznymi, które mają odmienny język oraz kulturę. Pozwala to na dodatkową weryfikację swoich interpretacji⁴⁸.

Wszystkie powyższe kwestie pokazują, jak ważna w projekcie jest dopracowana dokumentacja oraz jasno i klarownie zdefiniowane wymagania.

Eskalacja konfliktu

Podpisując umowę outsourcingu należy również pomyśleć o sytuacjach, w których klient nie może się porozumieć z dostawcą. W kontrakcie powinien być jasny i obiektywny opis eskalacji konfliktu oraz rozwiązania pojawiających się problemów uzgodniony na samym początku. Wczesna identyfikacja problemu, może być opisana w sekcji dotyczącej ryzyka projektowego. Powinna się tam też znaleźć informacja odnośnie sposobu łagodzenia problemu. Pozwoli nam to uchronić projekt przed całkowitym niepowodzeniem w trudnych, konfliktowych momentach. Niestety, często taki zapis jest pomijany w kontraktach. Jeśli istnieje zaufanie w relacji klient > dostawca współpraca ma większą szansę na pełen sukces⁴⁹.

Komunikacja

Warto na samym początku ustalić efektywne kanały komunikacji oraz mieć na uwadze to, że w niektórych przypadkach będzie konieczność ich dostosowania do potrzeb, których nie udało nam się przewidzieć. Należy zorganizować formalne punkty kontaktowe między zleceniodawcą, a wykonawcą. W przypadku braku komunikacji, prowadzonej w czasie rzeczywistym, twarzą w twarz, należy zwrócić się ku innym środkom, dzięki którym pozostaniemy na bieżąco z tym, co robi wykonawca⁵⁰.

Należy również zadbać o poprawną komunikację w projektach, gdzie klient zleca wykonanie oprogramowania jednemu dostawcy, a testowanie drugiemu. Komunikacja na linii tester > programista jest kluczowym elementem do osiągnięcia zadowalających efektów.

⁴⁸ WIEGERS K., BEATTY J., *Specyfikacja oprogramowania, Inżynieria wymagań*, Helion, Gliwice 2014, s. 431.

⁴⁹ <http://www.tutorialspoint.com/white-papers/8.html> [dostęp: 19.08.2015].

⁵⁰ WIEGERS K., BEATTY J., *Specyfikacja oprogramowania, Inżynieria wymagań*, Helion, Gliwice 2014, s. 431.

Bezpieczeństwo

Pracownicy dostawcy powinni podpisać klauzulę o poufności, po to, aby klient miał gwarancję uszanowania informacji poufnych, które zostały określone w kontrakcie. Biuro dostawcy powinno mieć zamknięty dostęp z zewnątrz. Byli pracownicy oraz osoby niezatrudnione nie powinni mieć wstępu do biura. Infrastruktura sieciowa powinna być odpowiednio zabezpieczona i odporna na próby włamania z zewnątrz. Usługodawca zazwyczaj chwali się normami związanymi z bezpieczeństwem, które spełnia ich firma. Jeżeli jednak jest inaczej, należy o to dopytać.

Jeśli rzeczywiście zależy nam bardzo na bezpieczeństwie, możemy przeprowadzić eksperyment. Poprosić dostawcę o oprowadzenie po biurze. Podczas tej wizyty należy obserwować uważnie wszystko dookoła, jeśli dostrzeżemy rzeczy, których nie powinniśmy zobaczyć, np. dane związane z innymi projektami, należy poważnie rozważyć decyzję o współpracy. Ewentualnie w takiej sytuacji możemy żądać od dostawcy gwarancji, że prace przy Twoim projekcie będą się odbywać w hermetycznym środowisku, np. odseparowane pomieszczenie bez dostępu osób nieupoważnionych. Tego typu zabezpieczenia mogą jednak generować dodatkowe koszty⁵¹.

Odpowiedzialność

Istnieje pewna istotna kwestia związana z odpowiedzialnością. Pojawia się ona zazwyczaj, gdy od testowanej aplikacji może zależeć ludzkie życie. W kontrakcie należy jasno sprecyzować, kto ponosi odpowiedzialność za awarię systemu. Zazwyczaj obarcza się nią klienta, który powinien zadbać o wysoki standard oprogramowania. Dostawca powinien się upewnić czy w umowie widnieje stosowna klauzula.

Własność intelektualna

Ochrona własności intelektualnej ma niebywałe znaczenie dla klientów outsourcingu. Kontrakt z firmą outsourcingową powinien wyraźnie i dokładnie określać czyją własnością będzie wykonana praca. Jeśli tego nie określimy, może dojść do konfliktu, a sprawa może trafić do sądu. Sąd może potraktować pracę firmy outsourcingowej jako "pracę na zlecenie" i prawa własności trafią do klienta. Jednak w większości przypadków doktryna "pracy na zlecenie", raczej nie będzie brana pod uwagę, co skutkuje tym, że

⁵¹ <http://kaner.com/pdfs/outsource.pdf> [dostęp: 22.08.2015].

dostawca przejmie prawa do własności intelektualnej. Aby uniknąć nieporozumień, należy więc w umowie precyzyjnie określić, kto będzie właścicielem produktów outsourcingu⁵².

Aby stosować outsourcing testowania należy się odpowiednio przygotować. Zdobyć wiedzę, która pomoże nam wdrożyć oraz prawidłowo przeprowadzić rzeczony proces. Trzeba mieć świadomość, że włożony wysiłek nie przyniesie efektów natychmiastowo. Zajmie to zapewne więcej czasu niż się tego można spodziewać, a osiągnięcie pełnych korzyści jeszcze dłużej. Należy cyklicznie sprawdzać postępy prac, konfrontować czy postępy idą zgodnie z planem i czy jakość usług odpowiada zapisom w umowie SLA. Dokładne, jednoznaczne wymagania, brak dwuznaczności, odpowiedni kontakt klienta z dostawcą, są kluczem do osiągnięcia celu.

W kolejnym rozdziale zostanie przedstawiona charakterystyka oraz specyfika przykładowego projektu, którego testowanie odbywa się w procesie outsourcingowym.

⁵² <http://kaner.com/pdfs/outsource.pdf> [dostęp: 22.08.2015].

ROZDZIAŁ II

SPECYFIKA TESTOWANIA NA PRZYKŁADOWYM PROJEKCIE OUTSOURCINGOWYM

W niniejszym rozdziale została zaprezentowana charakterystyka badanego projektu, wykorzystane technologie, przedstawiony cykl produkcji oraz środowisko testowe.

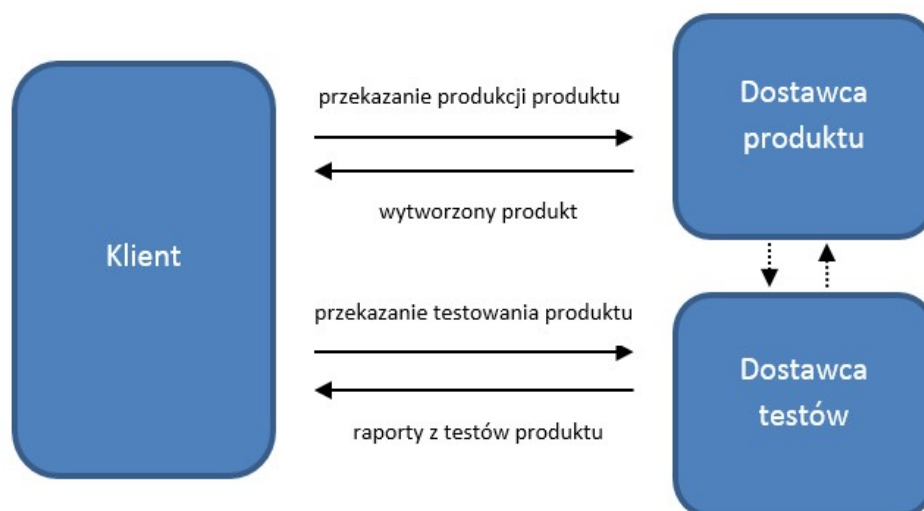
Celem rozdziału jest przedstawienie specyfiki testowania w przykładowym projekcie, wyciągnięcie wniosków oraz wskazanie sposobów, dzięki którym współpraca outsourcingowa może ulec znacznej poprawie.

2.1. Wprowadzenie do projektu

Opisywany projekt realizowany jest za pośrednictwem outsourcingu. Wytworzenie produktu zlecone zostało dwóm niezależnym podmiotom, z którymi klient podpisał dwie osobne umowy outsourcingowe. W całym procesie produkcji uczestniczą więc trzy organizacje (rysunek 2.1.):

- Klient – organizacja transferująca czynniki produkcji i zasoby potrzebne do realizacji pracy oraz prawa do podejmowania decyzji,
- Dostawca produktu – organizacja podejmująca decyzje i wykonująca prace produkcyjne,
- Dostawca testów – organizacja podejmująca decyzje i wykonująca prace związane z testowaniem produktu.

Rys. 2.1. Relacje outsourcingowe między organizacjami



Źródło: Opracowanie własne

W dalszej części rozdziału przedstawiona została przede wszystkim relacja między klientem a dostawcą testów. Istnieje również w mniejszym zakresie nawiązanie do relacji dostawcy testów z dostawcą oprogramowania.

Dostawca testów rozlicza się z klientem na podstawie popularnego modelu cenowego, czyli według zużytych zasobów i czasu pracy, a zdefiniowaną jednostką pracy jest tutaj roboczogodzina.

2.2. Zarys produktu oraz technologie użyte w projekcie

Produktem, na podstawie którego została przeprowadzona analiza procesu testowego jest zintegrowany system klasy ERP, bazujący na rozwiązaniach klient/serwer.

Jest to elastyczny system, za pomocą którego można planować i zarządzać procesami w przedsiębiorstwie. Podzielony jest on na kilka modułów, które w niezależny sposób mogą ze sobą współpracować.

Dodatkowo aplikacja integruje się z systemami zewnętrznymi w celu pobrania danych potrzebnych do realizacji konkretnych zadań w projekcie.

Jak wspomniano, produkt jest typową aplikacją klient/serwer. Strona serwerowa aplikacji pisana jest w języku programowania Java 1.7. Dodatkowe biblioteki wspierające część serwerową to Spring oraz Hibernate. Część kliencka oparta jest na framework-u GWT oraz dodatkowej bibliotece rozszerzającej jego możliwość – SmartGWT. Aplikacja współpracuje ze wszystkimi popularnymi, współczesnymi przeglądarkami w nowych wersjach, takimi jak: Firefox, Chrome czy Internet Explorer.

Do przechowywania danych wykorzystana została relacyjna baza danych PostgreSQL, a do usług integracyjnych wykorzystano protokół SOAP.

Serwer aplikacyjny, do którego dostosowany jest produkt, to JBoss w wersji 7.1.1.

2.3. Cykl produkcji

Tworzenie produktu odbywa się według metodyki zwinnej, a dokładniej metodyki Scrum⁵³ oraz jej techniki programowania sterowanego testami (z ang. Test Driven Development)⁵⁴.

⁵³ <https://pl.wikipedia.org/wiki/Scrum> [dostęp: 21.09.2015].

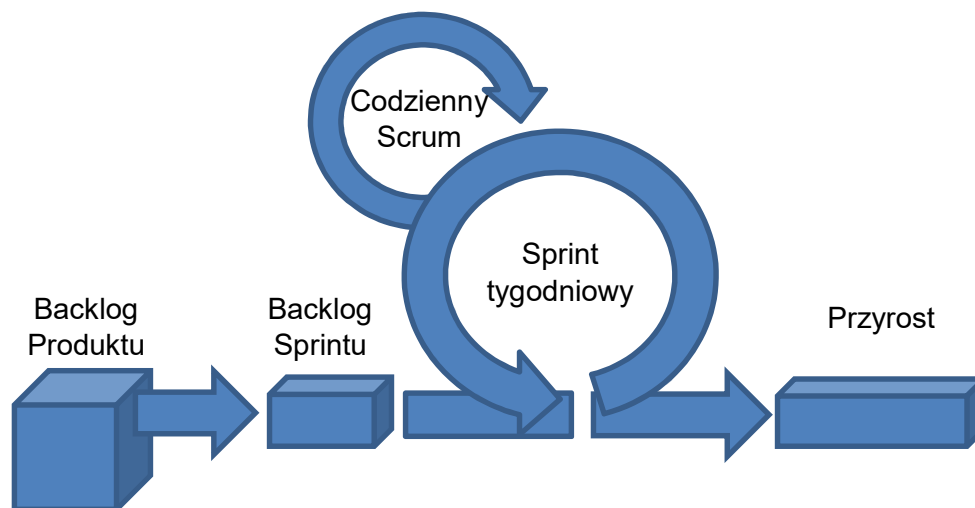
⁵⁴ https://pl.wikipedia.org/wiki/Test-driven_development [dostęp: 20.09.2015].

Produkcja oprogramowania opiera się na cotygodniowych sprintach, w których realizowane są ustalone zadania przebiegu, które ustalane są podczas planowania przebiegu. Jeśli podczas przebiegu nie uda się zrealizować niektórych zadań, przenoszone są one do kolejnego cyklu.

Codziennie rano odbywają się krótkie spotkania projektowe, w których każdy z programistów przedstawia co zrealizował w dniu poprzednim oraz mówi o problemach, które pojawiły się podczas pracy. Następnie cały zespół wspólnymi siłami próbuje znaleźć najlepsze rozwiązanie zaistniałego problemu.

Średnio raz na 2 miesiące wypuszczana jest wersja testowa aplikacji dla klienta, który przeprowadza na niej testy akceptacyjne i decyduje o podmianie wersji na środowisku produkcyjnym. Testy akceptacyjne realizowane są dopiero po wcześniejszym spełnieniu kryterium akceptacji. Informacja o tym jest dostarczana w postaci raportu od dostawcy testów. Rysunek 2.2. prezentuje cały cykl produkcji, natomiast na rysunku 2.3. przedstawiony jest szczegółowy cykl tygodniowy.

Rys. 2.2. Cykl produkcji oprogramowania zgodny z metodyką Scrum

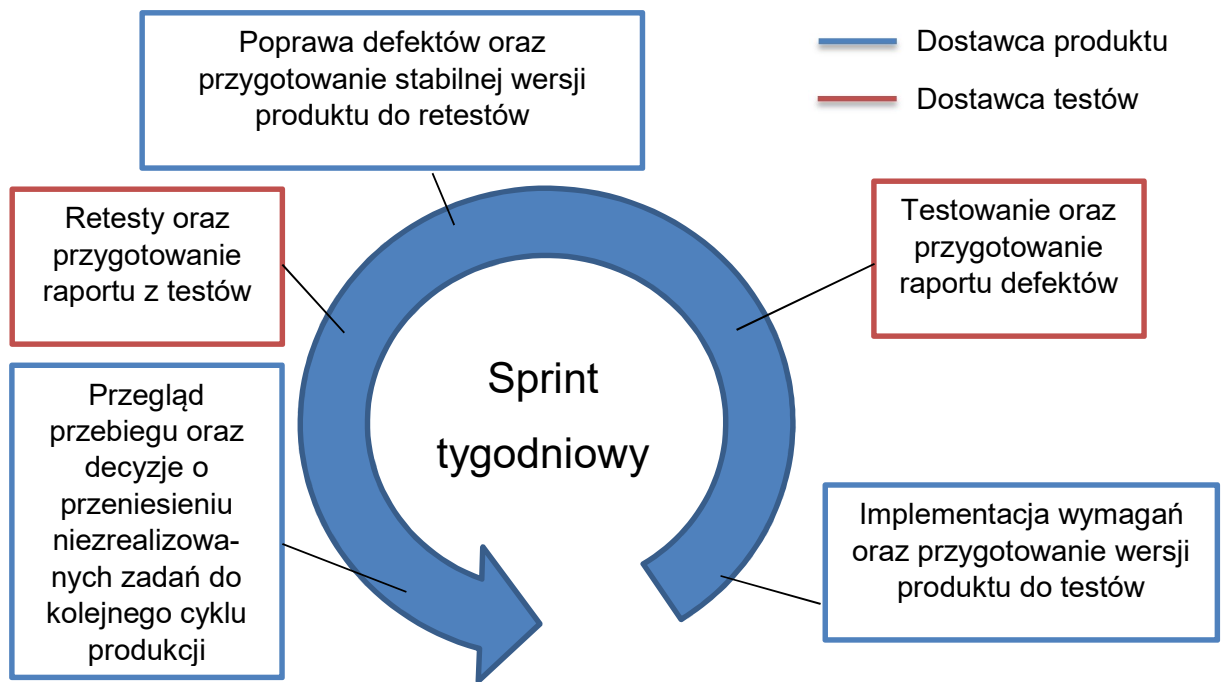


Źródło: Opracowanie własne na podstawie

http://www.moznainaczej.com.pl/Download/Konwersatoria/Rok_2014/Scrum-Guide-PL-v-July-2013.pdf

[dostęp: 16.09.2015].

Rys. 2.3. Szczegółowy sprint tygodniowy



Źródło: Opracowanie własne

Proces budowy wersji wykorzystuje narzędzie Jenkins, które wspiera proces ciągłej integracji (z ang. Continuous integration)⁵⁵.

W związku z tym, że do dostawcy testów oddelegowane są tylko niektóre procesy, w prezentowanym projekcie mamy do czynienia z selektywnym outsourcingiem IT.

2.4. Struktura zespołu projektowego

Struktura zespołu dostarczającego produkt jest dość standardowa jeśli chodzi o metodykę Scrum. Nie odbiega od często spotykanych rozwiązań. Możemy w niej wyodrębnić osoby, które pełnią określone role:

- Właściciel produktu – osoba reprezentująca klienta. Definiuje ona potrzeby biznesowe klienta. Jest w ciągłym kontakcie z przedstawicielem dostawcy, szybko reagując na nowe i zmienione wymagania. Zebrane informacje dostarczane są Scrum masterowi w dokumencie o ustalonym wcześniej formacie,
- Scrum master – odpowiada za usuwanie przeszkód, które utrudniają zespołowi

⁵⁵ https://en.wikipedia.org/wiki/Continuous_integration [dostęp: 21.09.2015].

wykonanie zadania oraz za poprawną implementację procesu i metod,

- Członkowie zespołu – aktualnie produkt tworzony jest przez grupę 8 programistów o różnym poziomie doświadczenia. W skład grupy wchodzi 3 programistów o zaawansowanych umiejętnościach, 3 programistów ze średnim stażem oraz 2 praktykujących studentów.

W zespole tworzącym oprogramowanie nie ma zatrudnionych testerów.

Zespół dostawcy testów pracuje w siedzibie swojej firmy outsourcingowej. W początkowych iteracjach, które wymagały większych nakładów pracy, zespół składał się z lidera testów oraz 4 testerów. W momencie przeprowadzanego badania, kiedy produkt był już w wersji stabilnej, zespół testowy został zredukowany do 2 testerów oraz lidera. Większość produktów powstałych w iteracji jest testowana w takim właśnie składzie. W przypadku testów wersji produktu, która posiada więcej obszarów podlegających testowaniu zespół zasila jeden lub dwóch dodatkowych testerów.

Klient nie wybierał konkretnych członków zespołu testerskiego. Decyzja o wyborze osób pozostała w rękach dostawcy testów. W umowie między klientem a dostawcą testów nie ma zapisu dotyczącego liczebności zespołu testowego. Jest natomiast wpis mówiący o tym, że dostawca musi zaangażować odpowiednią ilość zasobów, aby można było zrealizować cele w ustalonym terminie. Dostawca testów nie ma obowiązku informowania klienta o rotacjach w zespole.

Takie rozwiązanie daje możliwość dynamicznego przydziału członków zespołu. Sytuacja wydaje się korzystna z punktu widzenia dostawcy testów. Może on, bez żadnych konsekwencji, oddelegować testera do innych, bardziej pilnych zadań w firmie. Sytuacja klienta natomiast jest mniej komfortowa, ponieważ każda zmiana w zespole testerskim powoduje niepotrzebne komplikacje, wymaga dodatkowego czasu na wdrożenie nowego pracownika, co finalnie może skutkować wzrostem kosztów poniesionych przez klienta oraz wydłużeniem czasu zakończenia testów.

2.5. Środowisko testowe

Środowisko testowe ulokowane jest całkowicie po stronie dostawcy produktu. Dostawca testów uzyskuje jedynie dostęp do środowiska za pomocą wirtualnej sieci prywatnej

(z ang. VPN)⁵⁶. Każdy z testerów ma przypisane indywidualne konto z danymi dostępowymi, a po zalogowaniu do sieci ma dostęp do środowiska testowego.

Środowisko to wykorzystuje trzy serwery:

- serwer z wersją dzienną, która budowana jest codziennie o północy,
- serwer, na którym znajdują się wersje przygotowane podczas tygodniowych cykli produkcyjnych,
- oraz serwer testowy, na którym znajduje się wersja stabilna, przeznaczona głównie do testów akceptacyjnych.

System do zarządzania pracą oraz raportowania defektów to Redmine⁵⁷. Aplikacja ta również znajduje się na serwerach dostawcy produktu. Klient oraz testerzy mają utworzone indywidualne konta pozwalające na dostęp do systemu.

2.6. Wykonywane testy

Testy produktu wykonywane są przez każdą z organizacji uczestniczących w projekcie.

Podczas implementacji na wczesnym etapie produkcji testowanie opiera się głównie na testach modułowych wykonywanych przez dostawcę produktu. Programiści z zespołu dostawcy dodatkowo we własnym zakresie wykonują testy manualne implementowanych funkcjonalności.

Tworzona aplikacja integruje się z usługami Webservice. Testowanie tych usług sieciowych wywoływane jest w sposób manualny przez programistów podczas wytwarzania wersji.

Pozostałe testy, które są wykonywane w trakcie cykli produkcyjnych nowej wersji oddelegowane zostały do zespołu dostawcy testów.

Przez dostawcę testów wykonywane są dwa typy testów: testy regresyjne oraz testy systemowe funkcjonalne. Grupa testująca dostawcy testów wszędzie tam gdzie jest to możliwe stara się tworzyć i wykonywać przypadki testowe korzystając z narzędzia automatyzacji Selenium IDE⁵⁸. Tam gdzie nie jest możliwa automatyzacja przypadki

⁵⁶ https://pl.wikipedia.org/wiki/Virtual_Private_Network [dostęp: 21.09.2015].

⁵⁷ <https://en.wikipedia.org/wiki/Redmine> [dostęp: 20.09.2015].

⁵⁸ <http://www.seleniumhq.org/projects/ide/> [dostęp: 03.10.2015].

testowe są wykonywane w sposób manualny.

W opisywanym projekcie testerzy nie zawsze testują tylko kompletną, nową wersję produktu. Zdarza się, że testy wymagań funkcjonalnych przeprowadzane przez testerów odbywają się na wersjach dziennych, w których zaimplementowane zostały tylko niektóre wymagania. Dzieje się tak, ponieważ cykl produkcji wersji jest dość krótki (7 dni) i czasem pod jego koniec brakuje czasu na poprawę defektów oraz kolejne retesty. W przypadku, gdyby wersja zawierała wiele zmian i nowych funkcjonalności, a testerzy czekali na dostarczenie pełnej wersji produktu do testów, zaplanowane zadania przebiegu mogłyby zostać niezrealizowane.

Iteracyjna metodyka oraz krótkie cykle produkcji, w których dostarczane są kolejne funkcjonalności mogą wpłynąć negatywnie na istniejące już i działające prawidłowo części systemu. W związku z tym dostawca testów wykonuje testy regresyjne, aby zweryfikować poprawność wcześniejszych implementacji. Jeśli nastąpiła zmiana w logice, której powyższe testy dotyczą, następuje dostosowanie przypadków testowych do wprowadzonych zmian.

Testy regresyjne wykonywane są przez dostawcę testów na wersji stabilnej produktu dostarczonej przez dostawcę produktu. Wersja ta ma zazwyczaj zaimplementowane wszystkie wymagania ustalone w przebiegu. Jeśli jednak któregoś wymagania brakuje, oznacza to, że zostało one przeniesione do kolejnego cyklu produkcyjnego.

Jeśli wynik testów regresyjnych jest negatywny, dostawca produktu musi wprowadzić niezwłocznie poprawki w kodzie i dostarczyć do testów wersję pozbawioną znalezionych defektów.

Do czasu, aż testy regresyjne nie zakończą się sukcesem kompletne testy systemowe funkcjonalne nie są wykonywane. Testy systemowe funkcjonalne sprawdzają czy system spełnia wymagania zapisane w specyfikacji. Dzięki nim można znaleźć braki w implementacji lub niezgodność funkcji z wymaganiami. Testy akceptacyjne wykonywane są we własnym zakresie przez klienta. Klient realizuje testy na tym poziomie po wcześniejszym przekazaniu pozytywnej opinii o wersji produktu przez zespół testerów.

Dużą bolączką wydaje się brak testów integracyjnych wykonywanych przez testerów. Testy wykonywane przez dostawcę produktu mogą okazać się niewystarczające. Klient powinien wymagać od dostawców ich przeprowadzenia. Testy te powinny być wykonane

przynajmniej na dwóch poziomach. Testy integracyjne modułów sprawdzające interakcje między modułami oprogramowania, dostawca produktu powinien wykonać po zakończeniu testów modułowych. Natomiast testy integracyjne systemów, które sprawdzają interakcję między systemami mogą zostać wykonywane przez dostawcę testów po zakończeniu testowania systemowego. Testy te powinny dostarczyć informacji o aktualnej jakości integracji w produkcji.

Wartym odnotowania jest fakt, dotyczący tego, iż aplikacja nie została poddana istotnym typom testów нефункциональных, które mogłyby poinformować o potencjalnych zagrożeniach i awariach.

W badanym produkcie brak jest testów obciążeniowych i wydajnościowych, czyli nie wykonano pomiaru zdolności systemu do przesyłania i przetwarzania w sensie fizycznym określonej ilości informacji w danej jednostce czasu. Dzięki tym testom można by było stwierdzić czy system będzie w stanie obsłużyć zakładaną liczbę użytkowników oraz na jakim poziomie znajduje się jego wydajność.

Klient nie ma świadomości wagi i konieczności prowadzenia charakterystyki bezpieczeństwa, w związku z tym testy bezpieczeństwa nie są prowadzone. Przez bezpieczeństwo systemu rozumiemy odporność na różnego rodzaju próby uzyskania nieautoryzowanego dostępu do aplikacji lub jej danych. Potencjalne defekty w aplikacjach webowych umożliwiające dostęp niepowołanych osób mogą prowadzić do poważnych konsekwencji. Chodzi przede wszystkim o wykradanie, podmianę lub usuwanie danych, włamania na serwery, podmianę haseł użytkowników lub podmianę treści na stronach WWW.

Jeśli testy нефункциональные wykażą nieprawidłowości w odniesieniu do wymagań zawartych w specyfikacji lub ogólnie uznanych standardów, to dostawca produktu będzie musiał wykonać niezbędne poprawki optymalizujące produkt.

2.7. Aktywności i produkty

Każdy tester musi określić w wewnętrznym systemie raportowania pracy ile czasu potrzebuje na konkretne aktywności testowe. Po skończonej pracy musi skorygować estymowaną wartość tak, aby odzwierciedlała stan faktyczny. Na podstawie tych danych przygotowywane są raporty dotyczące czasu pracy testerów.

Klient nie posiadając odpowiednich zasobów do wdrożenia kompletnego procesu testowania w ramach własnej organizacji, zdecydował się oddelegować część aktywności

testowych na zewnątrz. Aktywności testowe, które zostały przekazane do dostawcy testów są następujące:

- wykonanie testów regresyjnych,
- wykonanie testów systemowych funkcjonalnych,
- wykonanie testów dokumentacji i wymagań,
- przygotowywanie raportów z testów,
- przygotowywanie raportów z defektami.

Przed każdym cyklem testowania, klient określa zakres testów, na podstawie którego dostawca testów szacuje czas oraz koszt wykonania testów.

Każdy proces wytwórczy niesie za sobą wytworzenie produktu. Nie inaczej jest też w przypadku badanego procesu testowania. Dostawca testów podczas realizacji zleconych zadań wytwarza produkty:

- cykliczny raport statusu testów wraz z raportem defektów,
- raport kończący 2 miesięczny etap testowania,
- scenariusze i przypadki testowe,
- skrypty SQL tworzące testowe dane w bazie danych.

Plan testów wykonany został przez przedstawiciela klienta oraz lidera dostawcy testów. Na jego podstawie zespół testerski uruchamia czynności testowe zgodnie z założeniami oraz ustalonym harmonogramem.

2.8. Współpraca z wykonawcą

Komunikacja wewnątrz projektu

Komunikacja w projekcie jest bardzo istotnym elementem. Jeśli komunikacja jest utrudniona, może to odbić się negatywnie na jakości produktu oraz opóźnić czas oddania realizowanych zadań.

W prezentowanym projekcie kontakt dostawcy testów z klientem odbywa się głównie za pośrednictwem telefonu oraz poczty elektronicznej. Organizowane są również spotkania na zakończenie kolejnych faz procesu testowania. Spotkania te odbywają się zarówno w siedzibie klienta jak i w siedzibie dostawcy testów.

Kontakt na linii tester > programista to przede wszystkim komunikacja za pośrednictwem systemu raportowania defektów Redmine. Z relacji członków zespołu

dostawcy testów wynika, że kontakty z programistami z zespołu tworzącego produkt nie należą do zbyt łatwych. Zgłaszanie defektów przez testera niekiedy traktowane jest przez programistę jako atak na jego umiejętności. Zdarzały się również przypadki, że zgłoszony defekt tak naprawdę nie powstał na skutek błędu programisty. W związku z tym programista czuł się poszkodowany i próbował w dosadny sposób wytłumaczyć to testerowi, który defekt zgłosił, co z kolei prowadziło do sytuacji konfliktowych.

Komunikacja między dostawcami wydaje się niewystarczająca i należałoby ją wzbogacić o inne media, np. telefon, który w pilnych przypadkach mógłby przyspieszyć realizację zadań. Dobrym pomysłem wydaje się zorganizowanie spotkania integracyjnego grupy programistów i testerów. Znajomość osobista członków dwóch oddzielnych grup kooperujących ze sobą, w większości przypadków przekłada się na lepszą wydajność projektową oraz płynniejszą współpracę. Szczególnie, jeśli mamy do czynienia z dwoma kluczowymi dostawcami. Ponadto, należy wspomnieć, iż często relacje między programistą i testerem przedstawiane są jako nieustanna walka⁵⁹. Nie inaczej jest w badanym projekcie, dlatego być może dzięki wspomnianym integracjom, udałoby się zamienić walkę programistów z testerami na wspólną walkę zespołów z problemami i defektami.

Własność intelektualna

Umowa między klientem, a dostawcą nie precyzuje, do kogo należą prawa intelektualne do produktów powstałych w procesie testowania. Niestety brak tego zapisu może skomplikować współpracę obydwu stron. Aby uniknąć ewentualnych problemów i sporów po zakończeniu procesu testowego, należałoby w trybie pilnym, jasno określić, kto jest właścicielem tych produktów. Jeśli w chwili obecnej w umowie nie ma tego zapisu, wydaje się rozsądnym podpisanie aneksu do aktualnego kontraktu, który by uregulował tę brakującą kwestię.

Monitorowanie procesu outsourcingowego

Klient otrzymuje okresowe raporty po każdym tygodniowym przebiegu. Co dwa miesiące, przed wypuszczeniem na produkcję nowej wersji aplikacji otrzymuje on również raport kończący etap testowania.

⁵⁹ <http://www.4pm.pl/artykuly/tester-oprogramowania> [dostęp: 03.10.2015].

Klient posiada dostęp do centralnej bazy danych Redmine, gdzie znajdują się zaraportowane przez testerów i programistów defekty. Dodatkowo Redmine daje możliwość wygenerowania raportów oraz zestawień statystycznych bazujących na zgłoszonych defektach.

Defekty znalezione na poziomie testów akceptacyjnych wykonywanych przez klienta również trafiają do tej samej bazy danych. Dodatkowo niezależnie od wyników testów akceptacyjnych tworzony jest raport z tych testów, który jest przekazywany do dostawcy produktu oraz dostawcy testów.

Zespół tworzący oprogramowanie nie w każdym przypadku raportuje defekt. Jeśli defekt pojawia się w testach modułowych niskiego poziomu, zazwyczaj naprawiany jest on zaraz po wystąpieniu, bez formalnego raportowania. Natomiast, jeśli znaleziony defekt, niezależnie od poziomu, na którym go wykryto, ze wstępnej analizy wygląda na skomplikowany i czasochłonny, jest on raportowany w systemie Redmine.

Testerzy z firmy dostarczającej testy skrupulatnie raportują każdy, nawet najmniejszy znaleziony defekt.

Zarządzanie zmianami

W projektach zwinnych zdarza się, że zmianom podlegają wymagania zaimplementowanych rozwiązań. Jeśli taka sytuacja pojawia się w badanym projekcie outsourcingowym, dostawca testów dostosowuje do zmian istniejące testy regresyjne. Na przypadki testowe zostają naniesione niezbędne poprawki, a czasochłonność zmian podliczona i przedstawiona klientowi.

Umowa SLA

W prezentowanym projekcie między klientem, a dostawcą testów umowa SLA nie została podpisana. Większość ustaleń odbyło się na zasadzie ustnego porozumienia.

Aby usługi były świadczone na wysokim poziomie na każdym etapie testów, klient powinien zadbać o podpisanie z dostawcą rzeczzonego dokumentu.

Jasne określenie nawet bardzo rygorystycznych parametrów może świadczyć o odpowiedzialności dostawcy, który uważa, że będzie w stanie wykonać zlecone zadania, oraz niejako daje gwarancję na wykonaną pracę.

Podpisując umowę SLA zleceniodawca również wykazuje się świadomością i odpowiedzialnością, ponieważ zdaje sobie sprawę ze stopnia skomplikowania

systemów informatycznych i z wielu czynników, które mogą wpłynąć na realizację zleconych prac.

Wymagania i dokumentacja

Dokumentacja bazuje w dużej mierze na projektach wstępnych. Projekty te przygotowywane są przez zespół programistów, a następnie przekazane klientowi do akceptacji. Jeśli klient ma jakieś uwagi, wtedy są nanoszone poprawki, a następnie projekt trafia do ponownej oceny oraz ewentualnej akceptacji.

Dostarczona przez klienta zespołowi testerów dokumentacja stoi na dość wysokim poziomie. Na jej podstawie można przeprowadzić skutecznie testowanie. Testerzy, którzy wykonują testy twierdzą, iż jest ona wystarczająco dokładna i wyczerpująca.

Należy zaznaczyć, że testerzy nie wykonują testów dokumentacji oraz wymagań, jednak podczas przeprowadzania testów innego typu, wychwytyją nieliczne defekty, które zgłaszają w systemie Redmine. W odniesieniu do całości dostarczonego materiału jest to jednak znikomy procent, który nie wpływa w dużym stopniu na obniżenie jakości dokumentacji. W sytuacji, gdy pojawiają się wątpliwości odnośnie wymagania czy zapisów w dokumentacji testerzy kontaktują się z klientem w celu wyjaśnienia. Klient w większości przypadków szybko reaguje na pytania testerów. Jednak w niektórych przypadkach, gdy również ma wątpliwości, musi nawiązać dialog z dostawcą produktu w celu wyjaśnienia niejasności.

Rozstrzygnięcie konfliktów

W umowie outsourcingowej nie ma zapisów odnośnie eskalacji konfliktów. Istnieje dość duże zaufanie między klientem, a firmą testującą. Do chwili obecnej podczas współpracy nie było sytuacji konfliktowych. Jednak, aby uniknąć potencjalnych nieporozumień w przyszłości, dobrze byłoby się zabezpieczyć na taką ewentualność i spróbować usprawnić proces poprzez ustalenie odpowiednich mechanizmów oraz wyznaczenie odpowiedzialnych osób, za pośrednictwem których można będzie przeprowadzać mediacje. Pozwoli to przygotować się w ten sposób na ewentualne problemy, które mogą wyniknąć w czasie trwania współpracy, a być może nawet pozwoli uchronić przed zerwaniem kontraktu i sądowym dochodzeniem roszczeń.

Bezpieczeństwo i odpowiedzialność

Testerzy pracujący w firmie dostawcy mają podpisaną klauzulę poufności dostarczoną przez klienta. Klauzula teoretycznie gwarantuje, że żaden z pracowników firmy outsourcingowej nie będzie rozpowszechniał danych otrzymanych od klienta. Jeśli taka sytuacja miałaby miejsce mogą zostać wyciągnięte konsekwencje prawne wobec tej osoby.

Zapewnienie bezpieczeństwa danych w chmurze leży po stronie dostawcy oprogramowania. Jak już wcześniej wspomniano, cała infrastruktura testowa znajduje się po stronie firmy tworzącej. Dostawca testów nie ma żadnego wpływu na bezpieczeństwo tych danych poza zachowaniem w poufności przydzielonych danych dostępowych do sieci VPN. O to muszą już zadbać indywidualnie pracownicy firmy testującej.

Dostęp do biura firmy testującej jest otwarty. Znajduje się portier weryfikujący przychodzące osoby, ale jest to zabezpieczenie niewystarczające. Z obserwacji wynika, że wejście do biura mimo obecności portiera jest bardzo łatwe. Wystarczy wspomnieć, że celem wizyty jest umówione spotkanie, a drzwi zostaną otwarte. Aby zapewnić wyższy poziom bezpieczeństwa można wprowadzić identyfikatory ze zdjęciem dla pracowników oraz weryfikować czy interesariusze mają rzeczywiście umówione spotkania. Zapobiegnie to dostaniu się do biura niepowołanych osób i tym samym utrudni dostęp do poufnych danych klientów.

2.9. Zaobserwowane ryzyka i zagrożenia

Dużym problemem w procesie testowym wydaje się uzależnienie od infrastruktury organizacji dostarczającej produkt. Testerzy nie mają bezpośredniego dostępu do logów serwerów oraz do bazy danych. Jest to duże utrudnienie spowalniające pracę. Przy każdorazowej potrzebie uzyskania takiego dostępu tester musi wysłać żądanie to wykonawcy. Wykonawca przygotowuje logi z określonego przedziału czasu i wskazanego serwera. Podobna sytuacja jest z bazą danych. Wykonawca przygotowuje zrzut bazy danych i przekazuje go testerowi.

Aby usprawnić proces przepływu danych warto pomyśleć o przygotowaniu infrastruktury testowej po stronie dostawcy testów. Dostęp testerów do logów i bazy danych byłby natychmiastowy, co mogłoby znacząco przyspieszyć procesy.

Warto wspomnieć, iż w przypadku awarii usługi VPN, systemu raportowania błędów Redmine, serwerów testowych bądź braku łączności z internetem, testy mogłyby

być przeprowadzone bez zakłóceń.

Dodatkowa infrastruktura wymaga utrzymania i związane są z nią dodatkowe koszty, jednak często firmy zajmujące się testowaniem, mają do dyspozycji dostępne zasoby i tak też jest w badanym przypadku. Klient powinien rozważyć czy koszty związane z wykorzystaniem infrastruktury dostawcy testów nie będą niższe niż ewentualne straty, które może ponieść w przypadku korzystania z rozwiązań dostawcy produktu.

ZAKOŃCZENIE

W niniejszej pracy dokonany został przegląd i analiza przekazanego do outsourcingu testowania. Podsumowując dotychczasowe rozważania można stwierdzić, że opisywana współpraca outsourcingowa ukazuje elementy, które są na wysokim poziomie i nie wymagają poprawy. Przede wszystkim chodzi tutaj o dostarczaną przez klienta szczegółową i dokładną dokumentację oraz wymagania. Monitorowanie procesu outsourcingowego również realizowane jest na odpowiednim poziomie.

Zaprezentowana w pracy analiza ukazuje również braki oraz nieścisłości. Do głównych zauważonych nieprawidłowości można zaliczyć: brak umowy SLA, brak ustalonych praw własności do treści intelektualnych, brak ustaleń dotyczących rozwiązywania konfliktów, słabą komunikację testerów z programistami, dynamiczny skład zespołu testowego, pominięcie wykonania ważnych typów testów.

Do dnia przeprowadzonych badań, między klientem a dostawcą testów nie stwierdzono konfliktów i nieporozumień. Jednak mając na uwadze występujące braki należy rozważyć uzupełnienie umowy outsourcingowej i kontraktu zapisami, które zostały pominięte. Zdaniem autorki pracy, pozwoli to zwiększyć szanse na zakończenie współpracy pełnym sukcesem.

Analizując pozytywne i negatywne aspekty współpracy outsourcingowej w prezentowanym projekcie, można dojść do wniosku, że współpraca ta jest udana. Znalezione defekty nie wydają się krytyczne i można je dość szybko i skutecznie poprawić.

Cel pracy, którym była teoretyczna i empiryczna analiza testowania w projekcie outsourcingowym został osiągnięty. Stan powyższej współpracy został zbadany, opisany i zaopiniowany.

Praca ukazuje teoretyczny i praktyczny obraz specyfiki testowania w projekcie outsourcingowym. W związku z tym, jej potencjonalnym zastosowaniem może okazać się wykorzystanie zawartych w niej wskazówek. Mogą one okazać się pomocne zarówno dla firm zajmujących się testowaniem w outsourcingu jak i ich potencjalnych klientów.

BIBLIOGRAFIA

1. Power M.J., Desouza K.C., Bonifazi C., *Outsourcing, Podręcznik Sprawdzonych Praktyk*, Warszawa 2010.
2. Roman A., *Testowanie i Jakość Oprogramowania. Modele, Techniki, Narzędzia*, PWN, Warszawa 2015.
3. Trocki M., *Outsourcing*, PWE, ISBN 83-208-1340-9, Warszawa 2001.
4. Wiegers K., Beatty J., *Specyfikacja Oprogramowania, Inżynieria Wymagań*, Helion, Gliwice 2014.

Źródła Internetowe

1. Stowarzyszenie Jakości Systemów Informatycznych, Certyfikowany tester, Plan poziomu podstawowego, http://sjsi.org/wp-content/uploads/2013/08/sylabus-poziomu-podstawowego-wersja-2011.1.1_20120925.pdf [dostęp: 16.09.2015].
2. Stowarzyszenie Jakości Systemów Informatycznych, Słownik wyrażen związanych z testowaniem, wersja 2.3 (2014), <http://sjsi.org/wp-content/uploads/2014/10/slownik-termin%C3%B3w-testowych-ver-2.3-PL.pdf> [dostęp: 05.05.2015].
3. <https://clutch.co/outsourcing-location/egypt> [dostęp: 03.08.2015].
4. <http://conference.eurostarsoftwaretesting.com/wp-content/uploads/cte2.pdf> [dostęp: 22.08.2015].
5. https://en.wikipedia.org/wiki/Continuous_integration [dostęp: 21.09.2015].
6. https://en.wikipedia.org/wiki/On-demand_outsourcing [dostęp: 11.08.2015].
7. <https://en.wikipedia.org/wiki/Redmine> [dostęp: 20.09.2015].
8. <http://financial-dictionary.thefreedictionary.com/Risk-Reward+Basis> [dostęp: 19.08.2015].
9. <http://ideas2action.pl/2012/10/02/jak-pisac-scenariusze-testowe/> [dostęp: 16.09.2015].
10. <http://kaner.com/pdfs/outsourc.pdf> [dostęp: 22.08.2015].
11. <http://outsourcing-journal.org/outsourcing-management/management-transformation/849-outsourcing-pricing-models-part-2-fixed-price-fp-model> [dostęp: 11.08.2015].
12. <http://outsourcing-journal.org/outsourcing-management/management->

- transformation/851-outsourcing-pricing-models-part-3-cost-reimbursable-model [dostęp: 13.08.2015].
13. <http://outsourcing-journal.org/outsourcing-management/management-transformation/855-outsourcing-pricing-models-part-4-time-and-materials-t-m-model> [dostęp: 13.08.2015].
14. <https://pl.wikipedia.org/wiki/Know-how> [dostęp: 16.08.2015].
15. <https://pl.wikipedia.org/wiki/Outsourcing> [dostęp: 02.08.2015].
16. <https://pl.wikipedia.org/wiki/Scrum> [dostęp: 21.09.2015].
17. https://pl.wikipedia.org/wiki/Service_Level_Agreement [dostęp: 18.08.2015].
18. https://pl.wikipedia.org/wiki/Test-driven_development [dostęp: 20.09.2015].
19. https://pl.wikipedia.org/wiki/Virtual_Private_Network [dostęp: 21.09.2015].
20. https://pl.wikiquote.org/wiki/Henry_Ford [dostęp: 30.07.2015].
21. <http://sjsi.org/slowo/plan-testow/> [dostęp: 18.09.2015].
22. <http://testerzy.pl/artykuly/raportowanie-defektow-dla-nie-testerow> [dostęp: 15.09.2015].
23. <http://testerzy.pl/baza-wiedzy/jak-sylabus-istqb-proces-testowy>, [dostęp: 16.09.2015].
24. <http://testhuddle.com/forums/topic/how-important-is-for-a-tester-to-know-the-characteristics-of-a-product/> [dostęp: 21.08.2015].
25. http://webdb.ucs.ed.ac.uk/operations/stratops/papers/Mark_power_2004.pdf [dostęp: 02.08.2015].
26. <http://www.4pm.pl/artykuly/tester-oprogramowania> [dostęp: 03.10.2015].
27. <http://www.businesswire.com/news/home/20050525005488/en/Knowledge-Transfer-Objective-Evaluation-Successful-Outsourcing-Engagements#.VdAzk3WUfrc> [dostęp: 17.08.2015].
28. <http://www.methodsandtools.com/archive/archive.php?id=74> [dostęp: 18.08.2015].
29. <http://www.seleniumhq.org/projects/ide/> [dostęp: 03.10.2015].
30. <http://www.softwaretestingmentor.com/articles/what-is-scenario-testing/> [dostęp: 16.09.2015].
31. <http://www.sourcimgmag.com/9-tips-about-outsourcing-your-software-testing/> [dostęp: 17.08.2015].
32. http://www.stickyminds.com/sites/default/files/magazine/file/2012/XDD12261filelistfilename1_0.pdf [dostęp: 06.08.2015].

33. http://www.stickyminds.com/sites/default/files/presentation/file/2013/11STRWR_T6.pdf [dostęp: 06.08.2015].
34. http://www.tholons.com/nl_pdf/Whitepaper_December_2013.pdf [dostęp: 03.08.2015].
35. <http://www.tutorialspoint.com/white-papers/8.html> [dostęp: 19.08.2015].
36. <http://www.zdnet.com/article/gartner-highlights-outsourcing-problems> [dostęp: 02.08.2015].
37. <http://www.zdnet.com/article/it-outsourcing-as-romania-vies-to-be-the-new-india-can-the-country-keep-up/> [dostęp: 03.08.2015].

SPIS RYSUNKÓW

1.1. Podział procesu testowego wg sylabusa poziomu podstawowego	5
1.2. Podział procesu testowego wg sylabusa poziomu zaawansowanego	6
1.3. Fazy wdrożenia outsourcingu	19
2.1. Relacje outsourcingowe między organizacjami	27
2.2. Cykl produkcji oprogramowania zgodny z metodyką Scrum	29
2.3. Szczegółowy sprint tygodniowy	30