

III faza konstrukcji

**Walentyn Baliuk
Anton Lunyov
Igor Tryhub**

1. Testy funkcjonalne

Warunek testowy	Przypadki testowe	Wynik testowania
Rejestracja użytkownika, wpisywanie adresu email	<ol style="list-style-type: none">1. Email:email@example.com, firstname.lastname@example.com, email@subdomain.example.com, firstname+lastname@example.com, email@123.123.123.123, email@[123.123.123.123], "email"@example.com, 1234567890@example.com, email@example-one.com, _____@example.com, email@example.name, email@example.museum, email@example.co.jp, firstname-lastname@example.com2. Email: plainaddress, #@%^%#\$@#\$@#.com, @example.com, Joe Smith <email@example.com>, email.example.com, email@example@example.com, .email@example.co	<ol style="list-style-type: none">1. Sukces, wszystkie dane są poprawne2. Wpisywanie tych emaili kończy się niepowodzeniem

	m, email.@example.co m, email..email@exam ple.com, あいうえお @example.com, email@example.co m (Joe Smith), email@example, email@-example.co m, email@example.we b, email@111.222.333 .44444, email@example..co m, Abc..123@example. com	
Rejestracja użytkownika, wpisywanie hasła	<ol style="list-style-type: none"> 1. 12345, qwerty, lolkek, q, abc, 54321, Anton 2. @nt0n#1337, .wr0c1ov3xxx, !nzyn13ri@, BardzoTrudneHaslo 12345 	<ol style="list-style-type: none"> 1. Takie hasła nie są wystarczająco bezpieczne wg wymagań, więc będą odrzucone 2. Takie hasła są dobre i przejdą walidację
Wpisywanie danych płatności	<ol style="list-style-type: none"> 1. Zła/nieistniejąca kombinacja numeru karty, daty wygasania karty oraz kodu CVV2, np 1234567890111213 , 12/12, 123 2. Dane istniejącej karty płatniczej (przykład nie będzie podany w celu bezpieczeństwa) 	<ol style="list-style-type: none"> 1. Płatność kończy się niepowodzeniem 2. Płatność kończy się sukcesem
Wykonywanie dozwolonych i	<ol style="list-style-type: none"> 1. Komentarz pod dowolnym postem 	<ol style="list-style-type: none"> 1. Taka funkcja jest dozwolona

niedozwolonych akcji przez zwykłego użytkownika	2. Dodawanie nowego produktu w sklepie	zwykłego użytkownikowi, więc on może to zrobić 2. Taka funkcja jest dostępna tylko administratorom portalu
---	--	---

2. Na podstawie norm ISO/IEC 9126 i 25 000 (przesłane materiały) opracowanie pomiarów mających na celu sprawdzenie czy system spełnia opisane w poprzednich dokumentach wymagania niefunkcjonalne.

Model jakościowy dostarczony przez standard ISO/IEC 9126-1, definiuje jakość produktu jako poniższy zbiór cech:

Funkcjonalność - zestaw atrybutów, które wiążą się z istnieniem zbioru funkcji i ich określonych właściwości. Funkcje te spełniają określone lub dorozumiane potrzeby.

- Stosowność
- Precyzja
- Interoperacyjność
- Bezpieczeństwo
- Zgodność z funkcjonalnością

Niezawodność - zestaw atrybutów, które zależą od zdolności oprogramowania do utrzymania pewnego poziomu wydajności w pewnych warunkach przez określony czas.

- Odporność na awarie
- Zdolność do odzyskania
- Zgodność z niezawodnością

Użyteczność - zestaw cech, które zależą od wysiłku potrzebnego do użycia produktu, oraz od indywidualnej oceny takiego użycia, przez określony lub dorozumiany zestaw użytkowników.

- Wyrozumiałość
- Operatywność
- Atrakcyjność
- Zgodność użyteczności

Wydajność - zestaw atrybutów, które zależą od poziomu wydajności oprogramowania i ilości wykorzystanych zasobów w określonych warunkach.

- Zachowanie czasowe
- Utylizacja zasobów

- Zgodność z wydajnością

Łatwość utrzymania - zestaw atrybutów, które zależą od wysiłku potrzebnego do wprowadzenia określonych modyfikacji.

- Analizowalność
- Zmienność
- Stabilność
- Testowalność
- Zgodność z łatwością utrzymania

Przenośność - zestaw atrybutów, które zależą od możliwości przeniesienia oprogramowania z jednego środowiska na drugie.

- Zdolność adaptacji
- Instalowalność
- Współistnienie
- Wymienialność
- Zgodność przenośności

3. Opracowanie planu beta testowania.

Należy określić kilka cech beta testowania danego produktu:

- Beta test musi być zamknięty
- Dobrą praktyką jest prośenie testera o wysyłanie informacji zwrotnej, np przez checkbox
- Wiadomo że 1 z 5 beta testerów rzeczywiście wyśle jakąś informację zwrotną, więc potrzebujemy ok. 500 uczestników beta testu, z których ok. 100 będą wysyłali przydatną informację zwrotną

4. Sporządzenie planu zarządzania ryzykiem.

Plan zarządzania ryzykiem jest zbiorem czynności których celem jest skłonienie menedżera do stworzenia i zorganizowania procesu zarządzania ryzykiem. Plan zarządzania ryzykiem także musi doprowadzać do stworzenia infrastruktury organizacyjnej. Do zadań infrastruktury organizacyjnej zaliczamy działania zmierzające do przygotowania alternatywnych sposobów działania, izolowania i zmniejszenia ryzyka, eliminacji ryzyka jeśli to możliwe, określenia rezerw czasowych i pieniężnych w celach bezpieczeństwa przed zagrożeniami, które mogą się pojawić podczas planowania i wykonywania zadań w projekcie.

1. Opis:

- Cel
- Działanie
- Wymagania

2. Podsumowanie działania:

- Podsumowanie wymogów
- Zarządzanie
- Zintegrowany harmonogram

3. Uwarunkowania zarządzania ryzykiem:

- Strategia zarządzania ryzykiem
- Tolerancja grup interesu wobec ryzyka
- Schemat planu zarządzania ryzykiem w organizacji

4. Struktura zarządzania ryzykiem:

- Definicje
- Rozwiązania
- Synchronizacja w czasie
- Mierniki
- Wdrożenie (obserwacja, ocena, obowiązki, odpowiedzialność)

5. Problemy związane z realizacją:

- Identyfikacja ryzyka
- Klasyfikacja ryzyka
- Pomiar ryzyka
- Planowanie ryzyka
- Planowanie metod reagowania na ryzyko

6. Zatwierdzenia

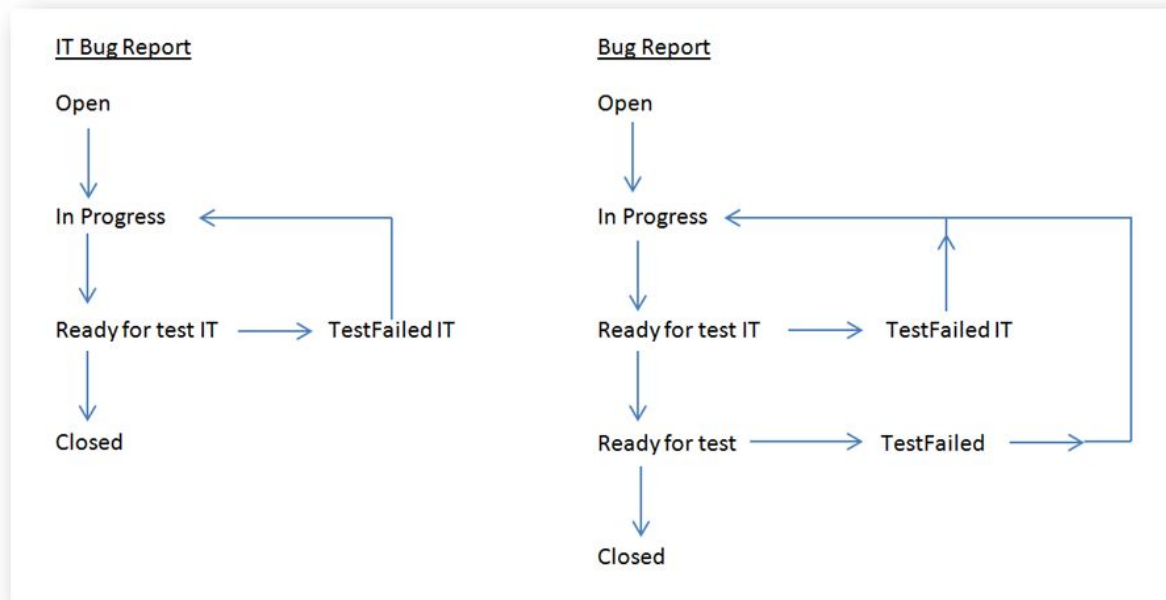
Ryzyko	Zdarzenie	Ppb (0-10)	Konsekwencje	Poziom tolerancji
--------	-----------	---------------	--------------	----------------------

Trudność realizacji logiki biznesowej	Nasze wymagania funkcjonalne są dość wygórowane i rozbudowane. Może się zdarzyć tak, że narzędzia, które zamierzamy użyć nie będą wspierały wszystkich funkcjonalności, a żeby je zrealizować trzeba będzie zainwestować nieodpowiednio duży wysiłek lub zmienić wcale stosowane narzędzia.	8	Krytyczne	Średni
Projektowanie interfejsu strony internetowej i jej implementacja	Jak wyżej, może się okazać, że Wordpress nie pozwala na realizację pewnych pomysłów design'owych dotyczących wyglądu stron portalu.	7	Marginalne	Średni
Błędy harmonogramowania	Projekt może zostać opóźniony w pewnym momencie, przez co trzeba będzie zatrudniać nowych pracowników, płacić stawki ponadnormowe, lub liczyć się z konsekwencjami prawnymi niedostarczenia aplikacji na czas.	4	Krytyczne	Wysoki
„Pełzający” zakres	Typowym niekorzystnym działaniem jest systematyczne rozszerzanie zakresu o dodatkowe wymagania lub swobodne modyfikowanie istotnych założeń projektowych, bez zwracania uwagi na konsekwencje. W takich działaniach celują często nieodpowiedzialni klienci, którzy „usztyniając” cenę i czas realizacji, oczekują spełnienia wszystkich, nawet najbardziej absurdalnych wymagań.	4	Katastroficzne	Niski

Czynnik ludzki	Pracownicy mogą opuszczać zespół przed zakończeniem projektu, wymagając nagle szukać zastępstwa na ich miejsce.	6	Pomijalne	Wysoki
Wydajność	Różnice pomiędzy zakładaną, a rzeczywistą wydajnością pracy mogą być duże, np. z powodu niedoszacowania na etapie planowania.	5	Marginalne	Wysoki
Nastawienie do pracy	Angażowanie do projektu pracowników o niewłaściwych lub niewystarczających kompetencjach do realizacji prac.	8	Krytyczne	Średni
Współpraca z klientem	Brak lub niewystarczające zaangażowanie klienta w prace projektowe, zagrażające pozytywnej akceptacji projektu przy końcowym odbiorze.	4	Krytyczne	Niski

5. Napisanie planu zarządzania jakością wytwarzania oprogramowania

- 1) Podział testów w zespole:
 - Programiści: testy jednostkowe/modułowe oraz testy „małe” integracyjne
 - Testerzy: testy systemowe, testy „duże” integracyjne, testy akceptacyjne
- 2) Harmonogram testowania – Gantt chart
- 3) Cykl życia błędu



4) Monitorowanie i nadzorowanie testów:

- Monitorowanie jaka część zaplanowanych przypadków testowych została przygotowana oraz wykonana
- Monitorowanie przebiegu wykonania testów (liczba wykonanych i niewykonanych przypadków testowych oraz liczba przypadków testowych które „przeszły” i „nie przeszły”)
- Statystyki o błędach (liczba znalezionych i naprawionych defektów oraz czas “życia błędu”)

5) Pie chart przebiegu wykonania testów (podzielonych według stanu)

6) Grafik liczby znalezionych błędów

7) Skład zespołu testowego

- Lider testów
- Testerzy (analitycy testów, programiści testów automatycznych, wykonujący testy)
- Inne osoby (programista – testy modułowe/integracyjne, specjalista biznesowy – testy akceptacyjne, zewnętrzne zasoby – testy wydajnościowe)

8) Raport z testów

- Lista błędów (wg. Poziomu krytyczności oraz wykonywania przez testerów)
- Raporty czasowe

6. Sporządzenie dokładniejszego planu wykonania produktu, dokładniejsza ocena pracochłonności, dokładniejszego harmonogramu.

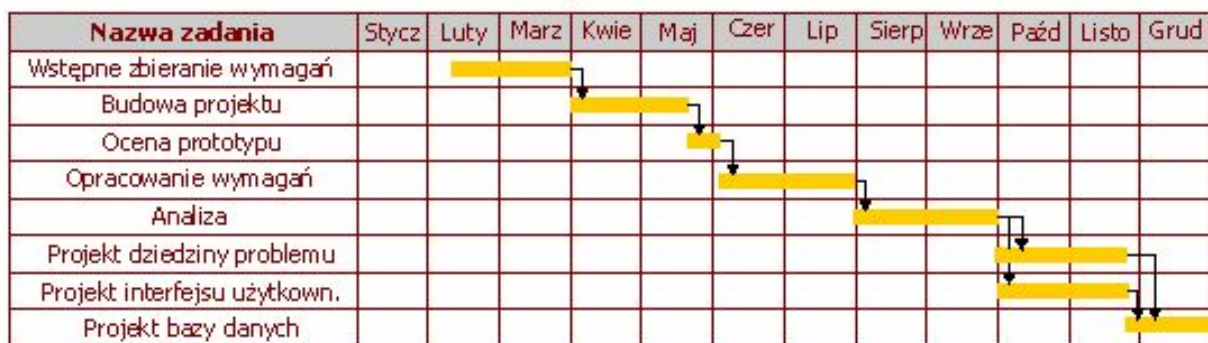
Aby zobrazować istotę szacowania pracochłonności wytwarzania oprogramowania, posłużymy się przykładem Gry Planistycznej, zaproponowanej w metodyce eXtreme Programming. Metoda ta ma na celu ułatwienie pertraktacji zakresu funkcjonalności, przeznaczonej do realizacji w ramach następnego przyrostu.

Pierwszym ogniwem jest Klient i wymagania przez niego sprecyzowane. Z perspektywy Klienta, wymagania mają przeważnie różne priorytety. Po drugiej stronie znajduje się Dostawca oprogramowania, który musi ocenić, czy jest w stanie zaimplementować daną funkcjonalność, w przewidzianym na to czasie. W tym właśnie miejscu pojawia się szacowanie pracochłonności. Rolą Klienta w Grze Planistycznej, jest sprecyzowanie wymagań i uszeregowanie ich według własnych priorytetów. Natomiast Dostawca oprogramowania powinien zbadać wykonalność i czas implementacji poszczególnych funkcji. Innymi słowy, należy oszacować pracochłonność, czyli określić sumaryczny czas potrzebny na wytworzenia oprogramowania.

Ostatnim etapem negocjacji z Klientem jest wybór funkcjonalności do następnego wydania. Dysponujemy priorytetami Klienta, które w połączeniu z oszacowaniem pracochłonności, umożliwiają dokonanie wyboru satysfakcjonującego obie strony.

Początkowa wersja harmonogramu w postaci Diagramu Gantta będzie po kolejnych etapach konsultacji z klientem uaktualniana:

Diagram Gantta



Jeżeli mówimy o procesie testowania, to podczas jego wykonania będziemy się sugerować listą następujących artefaktów będących produktami procesu testowego, np.:

- dokumentacja zarządcza – plan testów, szczegółowe harmonogramy

- specyfikacje testów – przypadki testowe, procedury, raporty, skrypty
- wyniki wykonywania testów
- dane testowe
- raporty z postępu realizacji testów

7. Ocena zgodności wykonanych prac z wizją systemu i specyfikacją wymagań.

Aby ocenić zgodność wykonanych prac z wizją systemu i specyfikacją wymagań, należy wykonać testy akceptacyjne, które zapewniają, że produkt jest gotowy do wykorzystania przez użytkownika końcowego. Zwykle takie testy są "black box" i nie są ściśle określone. Takie testy składają się z dwóch części: wewnętrzne oraz zewnętrzne testy akceptacyjne aka alpha i beta testy. Alpha testy są zwykle wykonane przez pracowników firmy, które brali udział w developmencie, ale nie bezpośrednio, np przez menadżerów. Beta testy są wykonane przez osoby, które nie brali udział w developmencie.