

Warszawa, 24.05.2016

Web-Notifier

Plan zapewnienia jakości

Dokument dotyczy projektu realizowanego w ramach zajęć z Inżynierii Oprogramowania na wydziale MIM UW w roku akademickim 2015/16.

Wersja: 1.1

Zespół: Tomasz Kacperek, Marek Mystkowski, Adam Sołtysik

Klient: mgr Grzegorz Grudziński

Spis treści:

1. Wstęp.
2. Metodologia.
3. Weryfikacja dokumentacji.
4. Weryfikacja aplikacji.
5. Walidacja aplikacji.
6. Testy z udziałem użytkowników.

1. Wstęp.

Celem dokumentu jest przedstawienie metod mierzenia i testowania jakości projektu, które pomogą w otrzymaniu produktu końcowego na najwyższym poziomie.

2. Metodologia.

Zapewnienie jakości projektu będzie się składać z kilku etapów:

- weryfikacja dokumentacji
- weryfikacja aplikacji
- walidacja aplikacji
- testy z udziałem użytkowników

3. Weryfikacja dokumentacji.

Ten etap będzie polegał na sprawdzeniu zgodności i spójności dokumentacji.

Po przygotowaniu odpowiednich dokumentów należy wnikliwie zbadać cały zbiór dokumentów w celu znalezienia braku spójności np. wykluczających się wymagań.

W przypadku znalezienia takich niespójności, należy je przedyskutować z klientem i wyeliminować.

Następnie należy przedstawić klientowi całość dokumentacji w celu weryfikacji i potwierdzenia zgodności z jego wymaganiami.

Reguły pracy nad dokumentacją:

1. Przedstawiciel zespołu i przedstawiciel klienta opracowują pierwszą wersję dokumentacji.
2. Zespół weryfikuje dokumentację, a następnie zatwierdza ją lub wprowadza poprawki.
3. Klient weryfikuje dokumentację, a następnie zatwierdza (ostatecznie) lub sugeruje poprawki. Klient może wprowadzić poprawki co najwyżej dwukrotnie. Przy braku zatwierdzenia przechodzimy do punktu 2.
4. Po zatwierdzeniu przez obie strony, dokumentacja nie podlega dalszym zmianom.

4. Weryfikacja aplikacji.

Po zaimplementowaniu każdej funkcjonalności oraz po każdej iteracji należy upewnić się, czy wykonany fragment aplikacji jest zgodny z (zatwierdzoną wcześniej) dokumentacją i czy spełnia wszystkie wymagania.

Reguły weryfikacji:

1. Programista A implementuje funkcjonalność X i zgłasza, że X jest gotowa.
2. Programista B (inny niż A) sprawdza zgodność X z dokumentacją.

5. Walidacja aplikacji.

Ten etap polega na przeprowadzaniu testów oprogramowania wchodzącego w skład projektu po każdej iteracji. Pozwala to na wczesne wykrywanie błędów programistycznych.

W tym etapie będą wykorzystywane testy jednostkowe, testy wydajnościowe oraz sprawdzanie przypadków użycia. Ocenie będzie również podlegać wygoda i ergonomia użytkowania.

Co należy przetestować:

- kompletna i poprawna realizacja wymagań
 - wymagania funkcjonalne
 - wszystkie platformy
 - zachowanie dla niepoprawnych danych
- testy jednostkowe wszystkich funkcji realizujących pełne funkcjonalności (bez funkcji pomocniczych)
- wydajność
 - śledzenie 400 stron / s na serwerze
 - śledzenie 20 stron / s na każdym urządzeniu
- bezpieczeństwo
 - przesyłanie danych
 - przechowywanie danych
 - testy penetracyjne
- użyteczność i ergonomia
 - prawidłowe wyświetlanie na wszystkich urządzeniach
 - estetyka
 - wygoda obsługi

6. Testy z udziałem użytkowników.

Ten etap polega na udostępnieniu gotowej aplikacji zamkniętej grupie użytkowników, którzy nie są członkami zespołu projektowego. Zadaniem osób z grupy testowej będzie przekazywanie informacji zwrotnych o błędach oraz propozycji zmian.

Testerzy powinni zwrócić szczególną uwagę na:

- wygodę użytkowania
 - wygodny interface
 - szybkość działania
 - łatwość obsługi
- realizację funkcjonalności na wszystkich dostępnych platformach
 - testowanie przypadków użycia

W przypadku znalezienia błędu, wybrany programista¹ identyfikuje jego przyczynę, pisze test jednostkowy sprawdzający dany błąd, a następnie poprawia kod.

¹ w przypadku trudności w wyborze programisty, decydujący jest ciąg $\text{ceil}(\log_2 n)$ rzutów monetą, gdzie n - liczba programistów, orzeł - 1, reszka - 0, kolejność big-endian