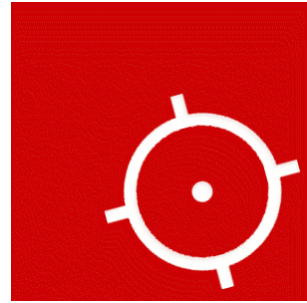


Dokumentacja testowa. Plan testów

[ang. Test Plan]

Plan testów jest jednym z podstawowych dokumentów w procesie testowym. Przedstawiamy wzór planu testów.



testerzy.pl

Zapraszamy do dyskusji o planie testów na forum strony <http://testerzy.pl/>

Plan testów – jest podstawowym dokumentem w procesie testowym. Poniżej prezentujemy wzór formularza dla planu testów dla metodyki tworzenia oprogramowania zgodnej z modelem V lub metodyką wodosпадową. Może on znaleźć zastosowanie również w metodach iteracyjnych.

Wzór planu zawiera klasyfikację defektów pod kontem ich ważności.

Nazwa produktu

Plan testów

Imię i nazwisko autora

Wersja 1.0

1. Wstęp

1. Tworzenie nazw plików

Standard nazywania plików

2. Wprowadzenie

Wprowadzenie do projektu

3. Cele

Co chcemy osiągnąć mówiąc o parametrach: jakość, plan i koszty

4. Podejście

Strategia testów mająca doprowadzić do osiągnięcia pożądanych celów

II. Opis testów

1. Testowany obiekt

Produkt lub aplikacja będąca celem testów

1.1. Obiekt: Aplikacja

- 1.1.1. Główny plik wykonalny
- 1.1.2. Instalacja/deinstalacja
- 1.1.3. Funkcje/Narzędzia
- 1.1.4. Pomoc online

1.2. Obiekt: Dodatki

- 1.2.1. Czcionka
- 1.2.2. Clip Art
- 1.2.3. Powiązane multimedia
- 1.2.4. Przykład/ tutorial
- 1.2.5. Czytajto
- 1.2.7. Inne

1.3. Obiekt: Dokumentacja

- 1.3.1. Przewodnik
- 1.3.2. CD
- 1.3.3. Opakowanie
- 1.3.4. Marketing/Informacje o produkcie/Pakiet reklam

2. Funkcjonalność do przetestowania

Lista rzeczy do przetestowania. Może zawierać opis środowiska testowego.

3. Funkcjonalność nie testowana

Lista rzeczy nie podlegająca testowaniu, lub nie uwzględnionych w tym planie testów.

4. Wymagania systemowe

WYMAGANIA KONFIGURACYJNE DLA OPROGRAMOWANIA I OSPRZĘTU

WYMAGANIA DLA ŚRODOWISKA KLIENTA

5. Wejście/Wyjście dla procesu tworzenia produktu

Opis kamieni milowych lub/i kryteriów akceptacyjnych dla aplikacji

6. Standardy/Bibliografia

- *IEEE Standard for Software Test Documentation* (ANSI/IEEE std 829).
- Kaner et al. *Testing Computer Software*, 2nd edition. New York: Wiley, 1993.
- *XXXX Test Matrix*.

[Lista wszystkich standardów, dokumentów potrzebnych do stworzenia planu testów]

7. Dostawy testowe

Lista materiałów przygotowanych przez grupę testową podczas cyklu testowego dostarczanych do projektu.

7.1 Plan testów

7.1.1 Wstępny plan testów (zaaprobowany)

Kompletny dokument zawierający potwierdzenie ważności

7.1.2 Dokumenty dodatkowe do planu wykonanych testów

Tablice przypadków testowych, matryce oraz inne materiały powiązane z testami z określonymi potwierdzeniami weryfikacji i ukończeniem testów.

7.1.3 Finalny plan testów(zaaprobowany)

Końcowy plan testów jest pomniejszą wersją planu testów programistycznych. Plan ten jest wyprodukowany i używany końcowych cyklach testów. Musi zawierać potwierdzenia zgodności.

7.1.4. Dokumenty dodatkowe do planu finalnych testów

Tablice przypadków testowych, matryce oraz inne materiały powiązane z testami z określonymi potwierdzeniami weryfikacji i ukończeniem testów.

7.2. System śledzenia defektów

7.2.1. Raporty defektów

- Lista podsumowująca znalezione defekty
- Pełny opis defektów

7.2.2. Baza danych defektów

Baza defektów znalezionych podczas cyklu testowego służąca zazwyczaj do wymiany informacji tester - programista.

7.3 Końcowy raport wydania

Raport , który powinien być wydany przed zakończeniem projektu. Raport ten jest dokumentem oceniającym jakość będącą w zakresie testowania projektu, kompletności testowania, rezultatów dla najważniejszych obszarów oraz rekomendacja lub jej brak do wydania produktu (klientowi lub na rynek)

III. Zarządzanie projektem testowym

1. Zespół projektowy

Lista członków zespołu projektowego wraz z ich rolami

2. Odpowiedzialni za testy

Kto będzie zarządzał wysiłkami testowymi? Inni ludzie i ich odpowiedzialność.

3. Zadania testowe

- Stworzenie planów testów wraz z przypadkami testowymi, matrycami i planami etc.
- Poddanie planu testów procesowi przeglądu wraz z uzyskaniem odpowiednich zgód.
- Uzyskać wymagania dla sprzętu/oprogramowania/narzędzia.
- Stworzenie bazy danych defektów.
- Przeprowadzenie testów.
- Zaraportowanie błędów.
- Przeprowadzenie spotkania dotyczącego defektów.
- Stworzenie cotygodniowego raportu.
- Stworzenie końcowego raportu.

4. Plan testów oraz harmonogram

Co będzie dostawą testów? Kiedy poszczególne dostawy będą realizowane?

5. Kryteria wejścia i wyjścia dla kamieni milowych projektu

Definicje, opisy oraz mierzalne kryteria dla kamieni milowych.

6. Harmonogram planu testów

6.1. Harmonogram

Lista grup zadań testowych wraz z opisem. Wstępne przypisanie zadań do osób.

6.2. Ocena ilości ludzi

Ocena ilości ludzi potrzebnych do ukończenie projektu.

7. Potrzeby treningowe

Zidentyfikowane potrzeby doształcenia osób w projekcie.

8. Potrzeby środowiskowe

8.1. Komponenty testowe

Lista wszystkich komponentów sprzętowych oraz oprogramowania potrzebnych do ukończenia projektu. Ich dostępność oraz strategia w ich pozyskiwaniu.

- Sprzęt
- Oprogramowanie
- Konta

8.2. Narzędzia testowe

- Narzędzia do kupienia na zewnątrz
- Wewnętrzne narzędzia
- Narzędzia, które należy stworzyć

8.3. Budynki, sprzęt, serwisy

Testowanie odbędzie się w laboratorium w firmie [nazwa firmy]. W przypadku konieczności zatrudnienia firmy zewnętrznej, plan zostanie zmieniony.

9. Plan integracji

Czy jest plan integracji? Jeśli tak, to jak się on ma do strategii testów?

10. Zawieszenie i ponowne rozpoczęcie testów

Kiedy testowanie powinno zostać zawieszone? Kiedy zawieszony proces testowania powinien być ponownie rozpoczęty?

11. Kryterium zakończenie testów

Kiedy kończy się testowanie?

12. Proces śledzenia defektów

12.1. Proces

Opis procesu śledzenia defektów.

12.2. Narzędzie śledzenia defektów (baza danych)

Opis narzędzia do śledzenia defektów.

12.3. Definiowanie ważności defektów

Ocena ważności błędu jest subiektywną metodą używaną do zaraportowania stopnia ważności każdego zaraportowanego defektu. Błędy oceniane będą na podstawie następujących wytycznych.

12.3.1.1. Krytyczny

Ważność 1— *Defekt krytyczny* (ang. show-stopper - blokujący wydanie) występuje gdy brakuje podstawowej funkcjonalności, użyteczności lub wydajności produktu podczas normalnych operacji; nie ma możliwości stworzenia "obejścia" (ang. work-around)

Zaliczamy do tej kategorii również defekty nie związane z funkcjonalnością ale będące jasnymi pomyłkami w np. nazwie firmy, złym klipem video na ekranie początkowym, błędna instrukcja obsługi etc.

Kilka dodatkowych kategorii:

- Wyłączenie lub uszkodzenie aplikacji/sprzętu
- Utrata danych lub ich uszkodzenie
- Brak sukcesu przy wykonywaniu podstawowych funkcji

12.3.2.2. Poważny

Ważność 2— *Poważny defekt* dla najważniejszych funkcjonalności, które nie działają prawidłowo w określonych warunkach lub funkcjonalności drugiego rzędu, które w ogóle nie działają, lub słaba jakość funkcjonalności lub wydajności podczas normalnych operacji, trudności z użyciem podstawowych funkcjonalności etc.

Zasadniczo, tego rodzaju defekty powinny zostać naprawione w cyklu tworzenia oprogramowania. Jednak w ostatecznej fazie testowej mogą zostać zaakceptowane (znaczy to nienaprawione)

12.3.3. 3. Nie-krytyczny

Ważność 3— *Nie-krytyczne defekty* to te z nich, które wprowadzają pewne problemy w obsłudze, niezdarzające się zbyt często. Zaliczamy tutaj: pomniejsze problemy w wyświetlaniu, błędy językowe, pomniejsze problemy z dizajnem oraz podobne.

Zazwyczaj błędy tego typu powinny być naprawione kiedy czas pozwoli z minimalnym nakładem sił przez programistów.

Wiele małych błędów może się przełożyć na ogólną niską ocenę jakości produktu.

13. Śledzenie statusu testów i raportowanie

W jakiej formie będą prezentowane raporty?

Z jaką częstotliwością będą pojawiać się raporty?

Jakie formacje zostaną zaraportowane?

14. Ryzyko i incydenty

Ryzyko i możliwe przypadki zmiany planu testów.

15. Proces potwierdzania zgodności

15.1. Potwierdzanie zgodności planu testów

Jak przygotowuje się plan testów i kto za niego odpowiada?

15.2 Potwierdzanie zgodności końcowo wydanego produktu

Jak wygląda proces potwierdzania wydania produktu?

Bibliografia:

- *IEEE Standard for Software Test Documentation* (ANSI/IEEE std 829).
- Kaner et al. *Testing Computer Software*, 2nd edition. New York: Wiley, 1993.
- Hung Q. Nguyen *Testing Applications on the Web*, 1st edition. New York: Wiley 2001.