

TESTOWANIE OPROGRAMOWANIA

#2 Rodzaje błędów i ich raportowanie



AUTOR: PAULINA KUCHARCZYK



AGENDA

1

PO CO SĄ TESTY?

Przyczyny i źródła błędów

2

RODZAJE TESTÓW

Techniki, typy i poziomy

3

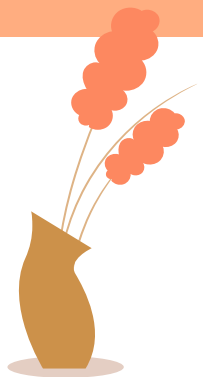
RODZAJE BŁĘDÓW

Typy i zgłoszenia

4

JAK ZGŁASZAĆ BŁĄD?

Zasady raportowania





TESTY?



- ...Testy
- Testy integracyjne
- Testy jednostkowe 
- Testy funkcjonalne
- Testy niefunkcjonalne
- Testy akceptacyjne
- Testy regresyjne



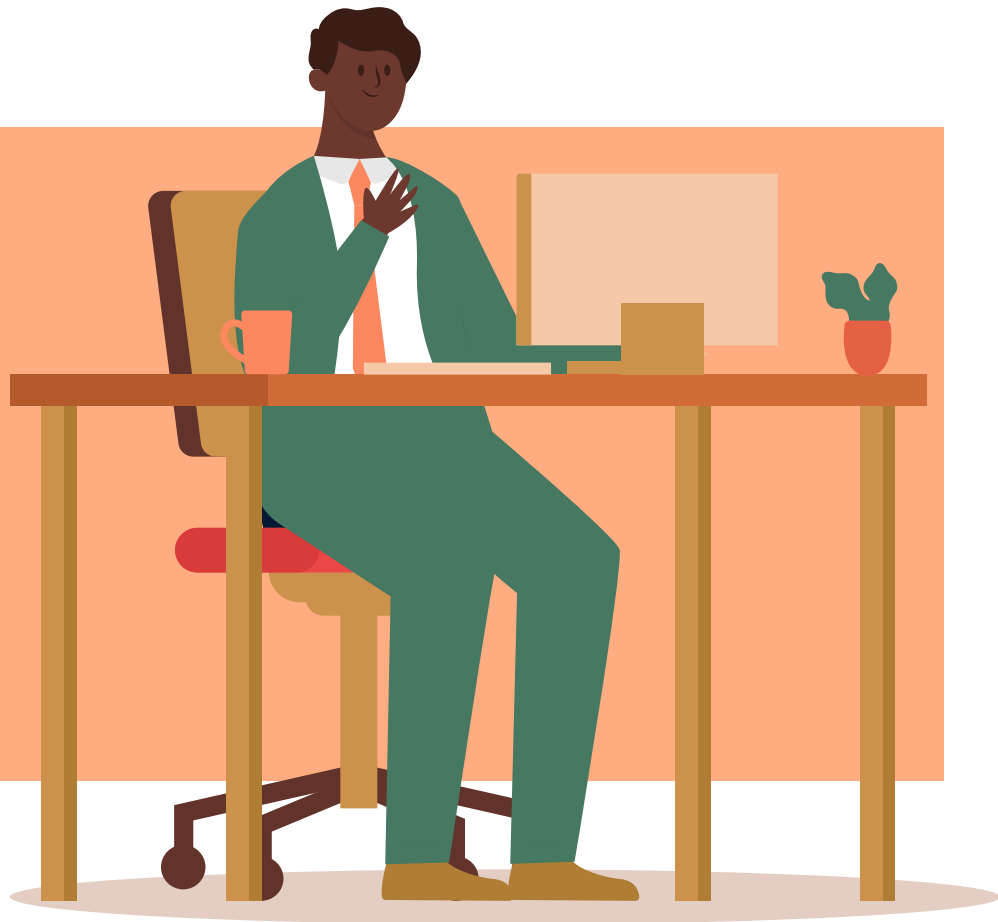
- Testy manualne
- Testy potwierdzające
- Testy automatyczne
- Testy dymne
- Testy strukturalne
- Testy białoskrzynkowe
- Testy czarnoskrzynkowe
- Testy...



01.

PO CO SĄ TESTY?

Przyczyny i źródła testów



DLACZEGO WARTO TESTOWAĆ?

By zapobiec utracie:

ŚRODKÓW FINANSOWYCH

CZASU I ZASOBÓW

WIZERUNKU FIRMY

ZDROWIA LUB ŻYCIA



DLACZEGO WARTO TESTOWAĆ?

Testy należy robić przez wzgląd na **dobro potencjalnego odbiorcy** – jego wrażenia z użytkowania (**UX**), ale także jego **zdrowie** czy **życie**. Przy okazji budujemy w ten sposób jego **zaufanie** wobec naszego produktu i całej marki, co niewątpliwie zaowocuje w przyszłości.

Ostatecznie, zyskujemy poczucie, że po uwzględnieniu wszystkich etapów wytwarzania produktu, czyli w tym również testowania, stworzyliśmy wreszcie kompletny, pełnoprawny produkt.



JAKIE JEST ZADANIE TESTERA?

- Weryfikowanie poprawności oprogramowania
- Dostarczanie informacji na jego temat
- Nie dopuszczanie do utraty zaufania przez klientów
- Doprowadzenie do ukończenia procesu developmentu



CO TESTUJE GENERUJE PRACĘ TESTERA?

01.

KADRA

Im więcej ludzi, tym o błędy łatwiej. Wytwarzanie oprogramowania to cykl więc nawet jeśli coś już działa, w przyszłości może się to zmienić.

02.

DEADLINE

Testy należy uwzględnić jeszcze na etapie planowania, aby później nie starać się gorączkowo ich wcisnąć lub zrobić byle jak czy ominąć - bo termin goni. Crunch rodzi błędy.

03.

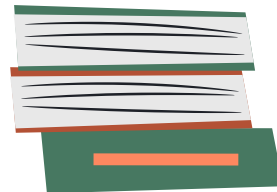
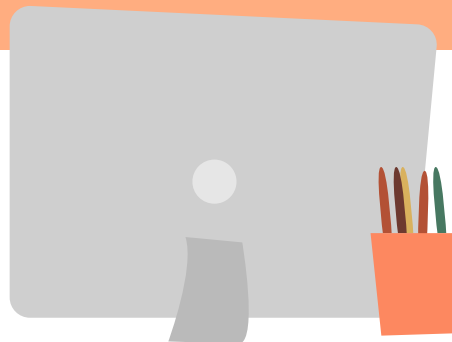
JAKOŚĆ

Mały błąd zbagatelizowany może przerodzić się w ogromny na dalszym etapie. Na produkcji może być już za późno by go złapać, zwłaszcza gdy zgłoszą go klienci.

04.

RYZYO

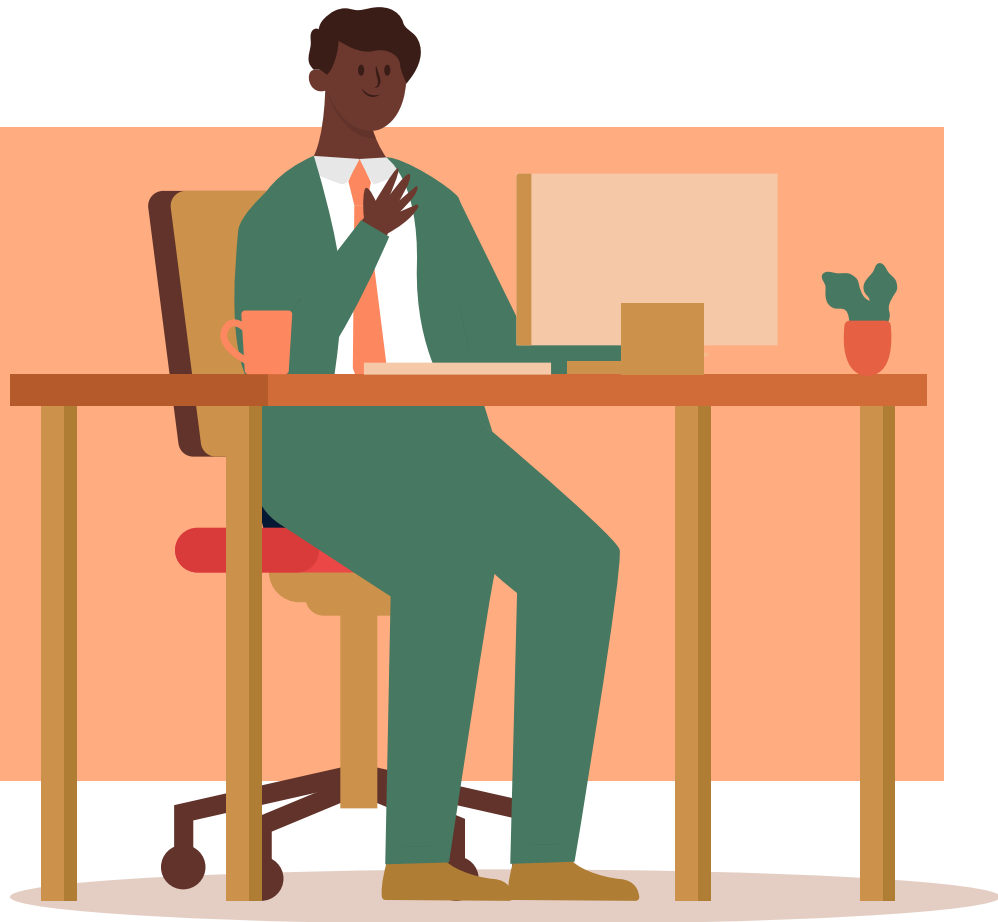
Wadliwy produkt może doprowadzić do poważnych wypadków jak utrata zdrowia bądź życia ale także do uszczerbku w morale, reputacji i konsekwencji prawne.



02.

RODZAJE TESTÓW

Techniki, typy i poziomy



PODZIAŁY TESTÓW

Ze względu na zastosowanie:



TECHNIKI

Styczne
(przeglądy, analiza kodu),
Dynamiczne
(testy gotowego systemu)



TYPY

Funkcyjne (czarnoskrzynkowe),
Niefunkcyjne (atrybutów),
Strukturalne (białoskrzynkowe),
Związane ze zmianą
(potwierdzające lub regresywne)



POZIOMY

Jednostkowe,
Integracyjne,
Systemowe,
Akceptacyjne

PODZIAŁY TESTÓW

Ze względu na wykonanie:



MANUALNE

Wykonywane przez człowieka, czasochłonne, ale skuteczne przy unikatowych przypadkach testowych.

Skuteczne dla testów: eksploracyjnych, użyteczności, ad hoc - bez planu.



AUTOMATYCZNE

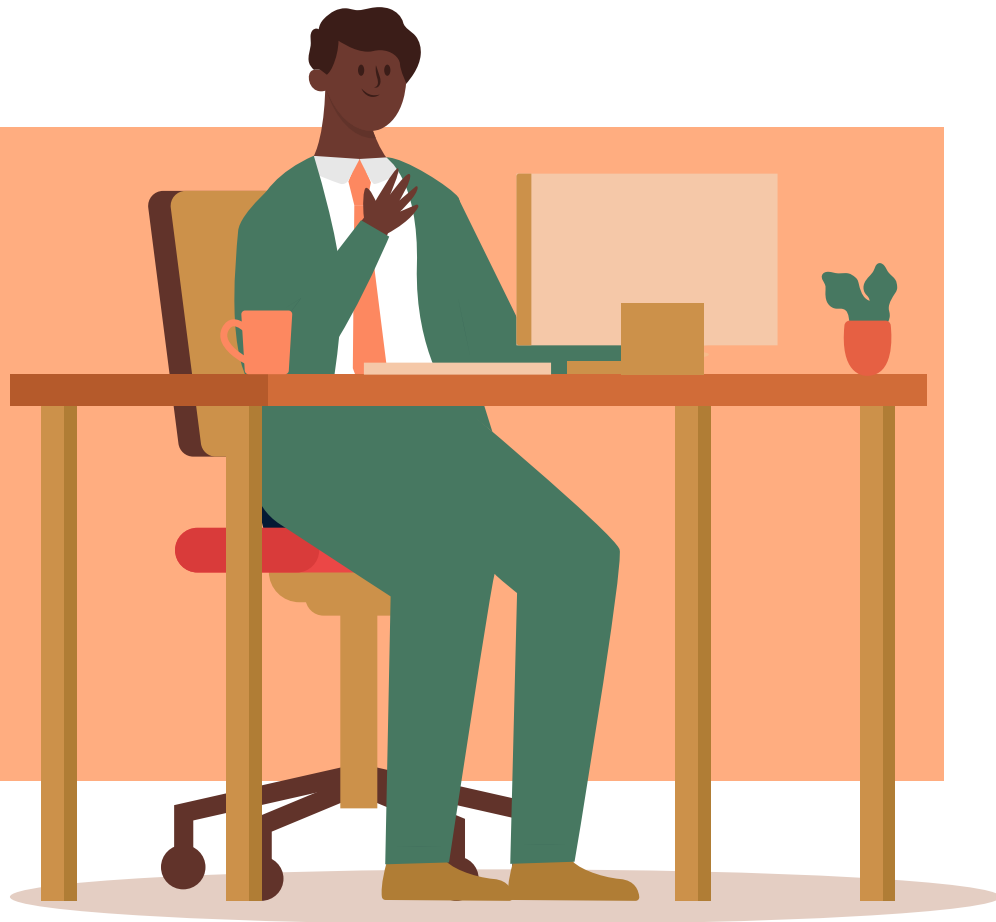
Wykonywane przez skrypt, szybkie ale trzeba je napisać i sprawdzają się przy powtarzalnych przypadkach testowych.

Skuteczne dla testów: obciążeniowych, funkcjonalnych, powtarzalnych.

03.

RODZAJE BŁĘDÓW

Typy i zgłoszenia



JAKIE ROZRÓŻNIAMY BŁĘDY?

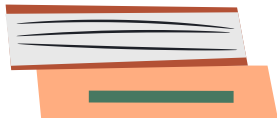
Według ISTQB

01.

POMYŁKA

Błąd ludzki, nieokreślone lub nieprawidłowe działanie człowieka prowadzące do defektu np. błąd projektanta lub programisty podczas pisania kodu.

Działanie człowieka powodujące powstanie nieprawidłowego rezultatu.



02.

DEFEKT

Usterka, błąd mający namacalną postać w kodzie lub projekcie prowadzący do powstania awarii np. Nieprawidłowy znak porównania w warunku.

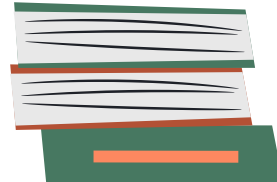
Niepoprawne wykorzystanie narzędzi deweloperskich

03.

AWARIA

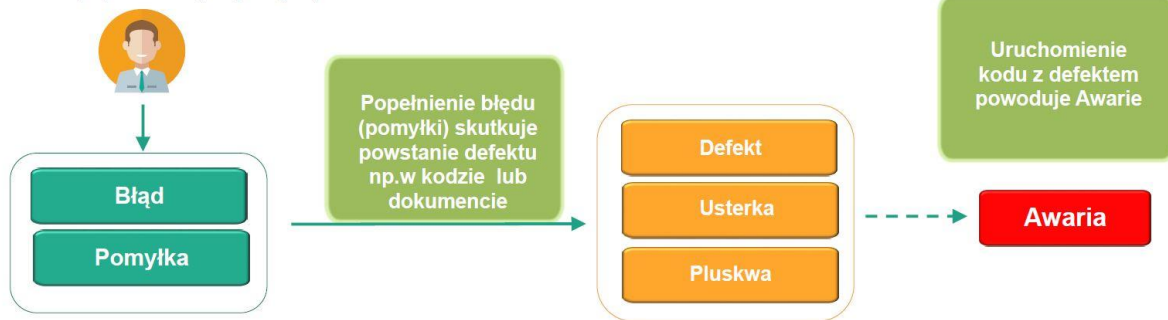
Odchyłka modułu lub systemu od oczekiwanego zachowania lub rezultatu działania np. po kliknięciu w przycisk Koszyka nic się nie dzieje.

Nieprawidłowe wykonanie funkcjonalności oprogramowania



JAK POWSTAJE PROBLEM?

Człowiek popełnia błąd / pomyłkę



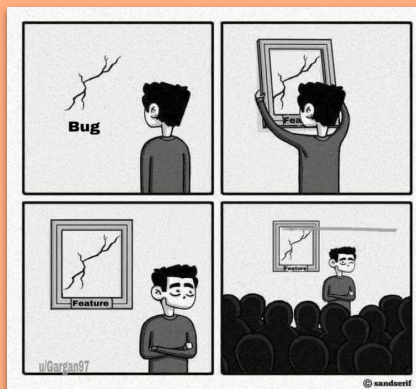
KIEDY SĄ PROBLEMY?

- Gry oprogramowanie nie działa zgodnie z oczekiwaniami (np. dokumentacji czy doświadczenia)
- Gdy wynik rzeczywisty różni się od wyniku oczekiwanego (np. $1+2 = 5$)
- Gdy wystąpiła utrata danych
- Gdy elementy graficzne dostępne w aplikacji źle się wyświetlają



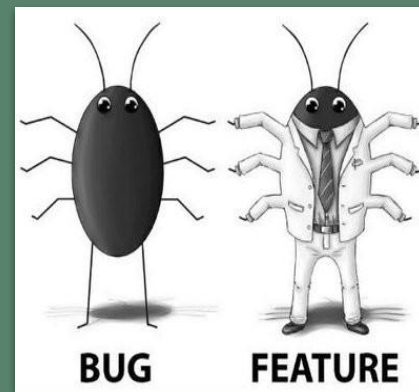
BŁĘDY (BUGS)

Istniejące już pomyłki, defekty
czy awarie.



SUGESTIE (FEATURES)

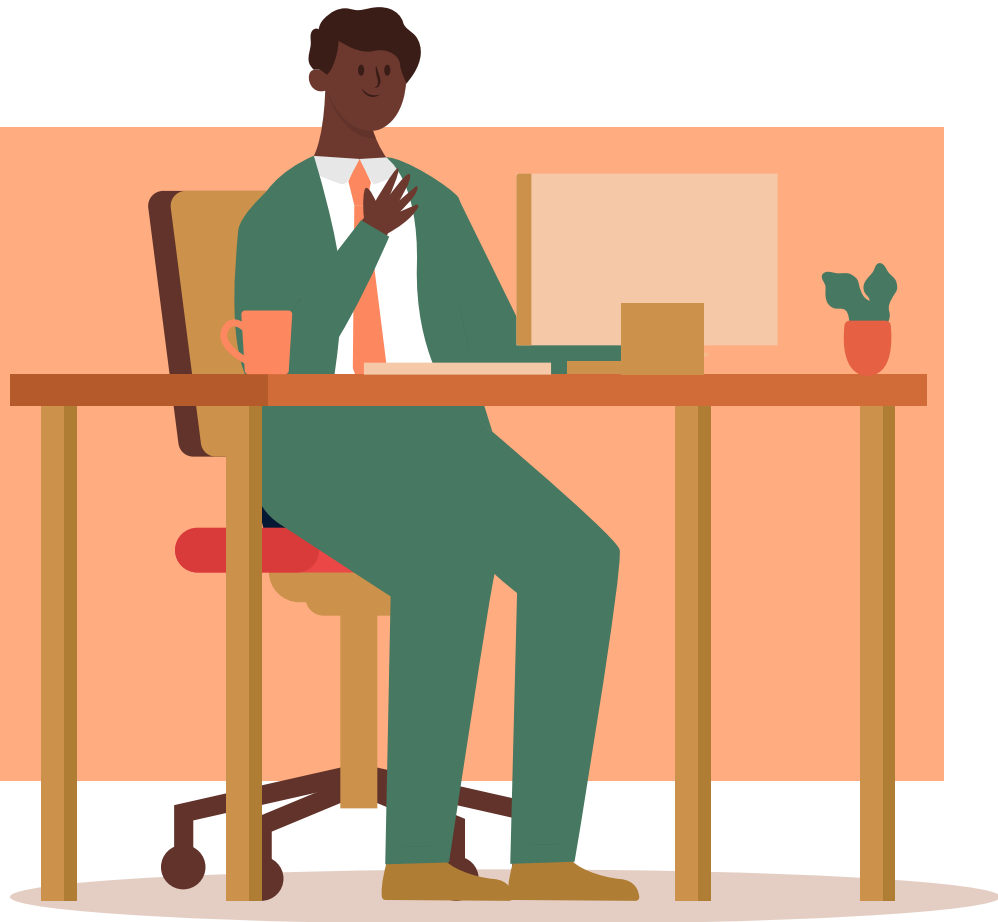
Sugerowanie zmiany mogącej
dać korzystny efekt.



04.

JAK ZGŁASZAĆ BŁĄD?

Zasady raportowania



RAPORT BŁĘDU

Zbiór różnych informacji, pozwalających na zdefiniowanie oraz odtworzenie błędu. Raport powinien zawierać w sobie wiele informacji pozwalających w łatwy sposób na analizę i identyfikację potencjalnego defektu. Wszystko zależy jednak od tego, jak wygląda proces zarządzania incydentami w naszej organizacji i jakie elementy są wymagane do zamieszczenia w raporcie.





ZAWARTOŚĆ PERFEKCYJNEGO RAPORTU



- Identyfikator
- Data zgłoszenia
- Autor zgłoszenia
- Typ zgłoszenia
- Priorytet
- Kategoria
- Aktualny status



- Środowisko testerskie
- Temat zgłoszenia
- Opis zgłoszenia
- Kroki do odtworzenia
- Oczekiwany rezultat
- Załączniki
- Wersja aplikacji



Bug #1276

Zwolnienie pracy komputera przy odinstalowywaniu aplikacji

Dodane przez Paulina Patrowicz 27 dni temu. Zaktualizowane 1 dzień temu.

Status: Feedback
Priorytet: High
Przypisany do: [Piotrek](#)
Wersja docelowa: 1.0.0.0
Instance: Staging
Operating System: Linux, MacOSX, Windows
Reproducibility: Always
Build Commit Hash:
Application Name:
Reach:

Data rozpoczęcia: 2019-05-15
Data oddania:
% Wykonania: 0%
Szacowany czas:
Impact:
Confidence:
Blocker: Nie
Client:
Known Bug Description:

Opis

Krótki opis błędu

Przy naprawianiu plików, w momencie gdy usuniemy więcej niż połowe plików, patcher zawiesza cały komputer i zwalnia jego pracę.

Spodziewane zachowanie

Patcher nie zmuli systemu.

Kroki do powtórzenia problemu

1. Włącz launcher i pobierz aplikację
2. Wyłącz patcher, wejdź w folder apps i usuń wszystko z wyjątkiem jsona
3. Włącz launcher i pozwól patcherowi naprawić pliki, efekt:

Dłgie Sprawdzanie integralności, Uninstall z zawiechami, po Uninstall około minuty not resnponding całego systemu a później 100% CPU dla patchera i bardzo wolne dalsze działanie wszystkiego.

Komentarz

Próbowaliśmy przebadac to z Piotrem ale nic sensownego nie wymyśliłiśmy. Problem zaczyna się pod koniec Odinstalowywania.

 [output_log.txt \(4,26 MB\)](#)  Paulina Patrowicz, 2019-05-15 11:18 



#	Projekt	Typ zagadnienia	Status	Temat
1321	Patcher Unity	Bug	New	Bardzo długi czas naprawiania plików
1276	Patcher Unity	Bug	Feedback	Zwolnienie pracy komputera przy odinstalowywaniu aplikacji
1255	Launcher Qt	Bug	Feedback	Możliwość włączenia wielu launcherów w offline mode
1256	Launcher Qt	Bug	Feedback	Nieprawidłowa akcja po uruchomieniu kolejnego launchera z rzędu
1110	Panel	Feature	New	Wcześniej pojawiający się komunikat o zablokowanych custom patcherach
1108	Panel	Feature	New	Brak możliwości wyjścia z okna po utworzeniu nowego konta
1109	Panel	Bug	New	404 przy odświeżeniu strony z aktywacją konta
1107	Panel	Bug	New	Komunikat o wysłaniu wkrótce wiadomości na maila, mimo wyłączonej opcji komunikatów mailowych.
1316	Tools	Bug	Feedback	Nieskończące się processing
1297	Patcher Unity	Bug	Resolved	Problemy z patcherem podczas testów node'ów





Wojciech G. (programista) (#2) Programista

Sep 7 ...

4) [Pilny] Napis "Kliknij" nie znika [Mechanika]

Kliknąć na ekerkę można nawet w tym przypadku, kiedy temu przeszkadzają inne przedmioty.

Kroki:

1. Wejść w pokój fizyki
2. Kliknąć na ołówek
3. Rzucić ołówek
4. Spojrzeć w inną stronę

Cel: Napis "Kliknij" znika, kiedy nie ma przedmiotu, który można kliknąć/wziąć/wcisnąć.



image.png · 106 KB · [View full-size](#) · [Download](#)

5 comments



TESTERSKIE LIFEHACKI?

Lifehacki

PRZYPADKI TESTOWE

Zbiór danych wejściowych, warunków wykonania, oczekiwanych rezultatów i warunków końcowych testowania danej funkcjonalności.

CHECKLISTA

Lista kroków szczegółowego testowania danej funkcjonalności składająca się na przypadek testowy.



TESTERSKIE LIFEHACKI?

Lifehacki

PRZYPADKI TESTOWE

Zbiór danych wejściowych, warunków wykonania, oczekiwanych rezultatów i warunków końcowych testowania danej funkcjonalności.

CHECKLISTA

Lista kroków szczegółowego testowania danej funkcjonalności składająca się na przypadek testowy.



[PK-023: Data Transfer Usage indicator](#)

[PK-024: User is allowed to upgrade account](#)

PK-019: Patcher file fixing

Requirements: Patcher should re-download a broken file [before game can be played](#).

Details

If any installed game file accidentally gets deleted or changed, patcher should detect that fact and re-download that file without the need of re-downloading the whole game.

To test it out:

1. Install the game
2. Close the patcher
3. Remove or replace any random file inside the game's directory
4. Start the launcher
5. See that patcher is re-downloading some data to fix that file

Objective: Fix a broken game.

CHECKLISTA – POPRAWNE RAPORTOWANIE ZGŁOSZEŃ

1 Czy uzupełniłeś wszystkie wymagane pola (*tytuł, Opis, Środowisko Testów, Priorytet, Autor Zgłoszenia, Wersja testowanej aplikacji, kategoria incydentu*) ?

☐

2 Czy w opisie są potrzebne wszystkie kroki potrzebne do powtórzenia defektu – wraz z używanymi danymi testowymi ?

☐

3 Czy w zgłoszeniu znajduje się opis tylko jednego defektu ?

☐

4 Czy w zgłoszeniu znajdują się opisane zrzuty ekranu ułatwiające identyfikację defektu ?

☐

5 Czy sprawdziłeś czy na liście zgłoszeń nie znajduje się już wykryty defekt ?

☐

6 Czy tytuł/temat zgłoszenia jest zwięźle napisany i odnosi się do wykrytego defektu ?

☐

7 Czy opisałeś dokładnie środowisko na którym wykryłeś defekt (*Wersja przeglądarki, System Operacyjny, Rodzaj urządzenia mobilnego itp.*) ?

☐

8 Czy defekt jest zgłoszony w sposób obiektywny ?

☐

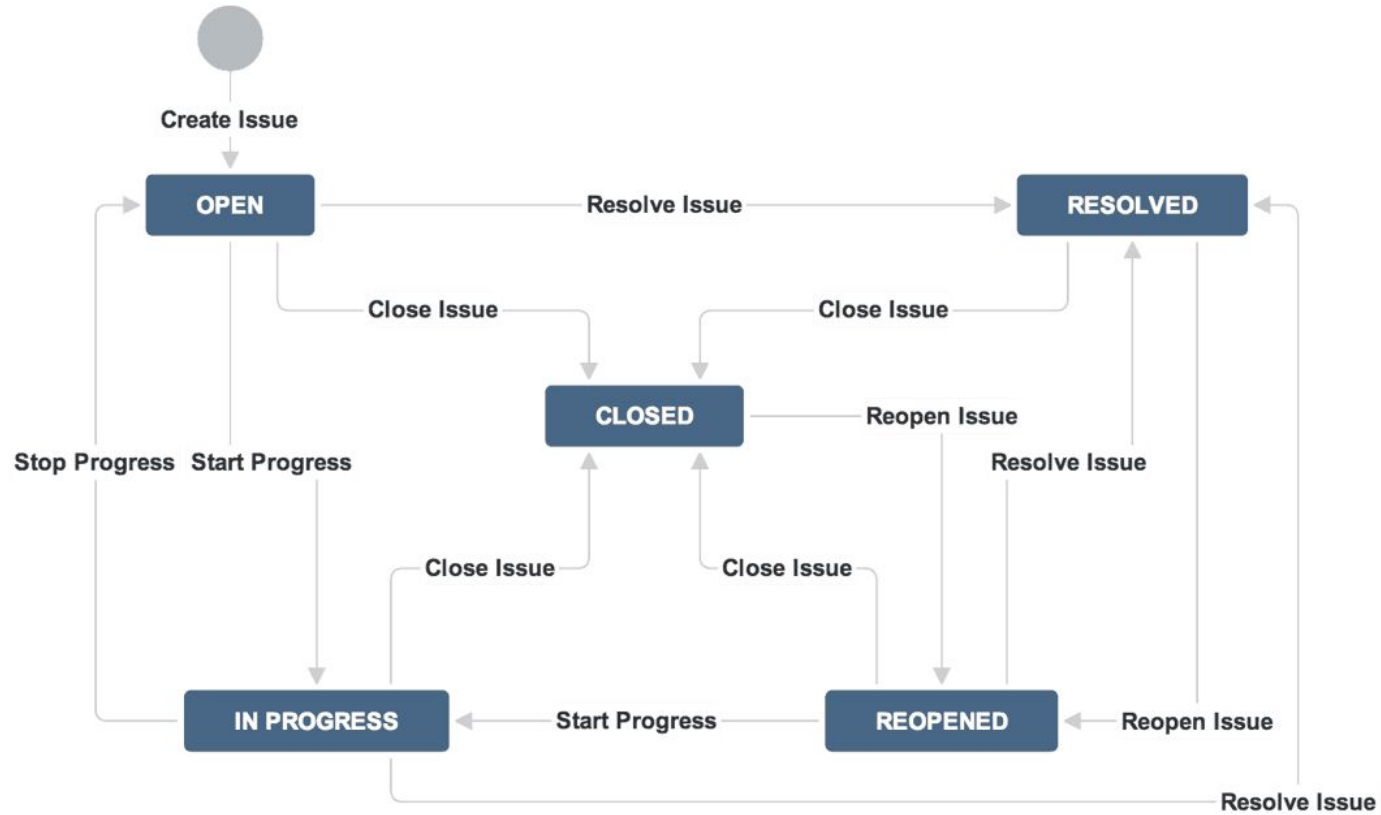
9 Przeczytaj jeszcze raz zgłoszenie i wprowadź niezbędne poprawki

☐

PODSUMOWANIE RAPORTU

Testerski rachunek sumienia

- ❑ Odpowiedź: Czy udało nam się zrealizować zaplanowane testy?
- ❑ Odpowiedź: Czy wystąpiły jakieś trudności niebędące defektami? (np. problemy ze środowiskiem, które opóźniły albo skróciły testy)
- ❑ Odpowiedź: Czy wyrobiliśmy się w estymacji czasu pracy?
- ❑ Informacja o możliwości weryfikacji raportu (np. odnośnik do przypadków testowych tworzonych pod testy akceptacyjne)
- ❑ Wnioski na przyszłość (pamiętajmy, aby uczyć się na błędach to jedna z ważniejszych czynności w IT).



ZADANIE PRAKTYCZNE



Myszka w dłoń!
i zapraszam do testowania gry oraz
sporządzania raportu błędów:

- bit.ly/3jheLff (ver 1)
- bit.ly/3cnIHDP (ver 8)

W dalszej części zajęć wspólne
omówienie wyników.





DZIĘKI ZA UWAGĘ!

Więcej materiałów znajdziesz na
kucharczyk.dev

Kontakt
pkucharczyk@technikumkreatywne.pl

CREDITS:

This presentation template was created by **Slidesgo**,
including icons by **Flaticon**, and infographics &
images by **Freepik**.

Please keep this slide for attribution.

Materiały i inspiracje

Prezentacja powstała w oparciu o serwisy:

TestNaTestera.pl

Testuj.pl

Testerzy.pl

Kodilla.pl

ToNieBug.pl

...oraz własną wiedzę i doświadczenie ;)