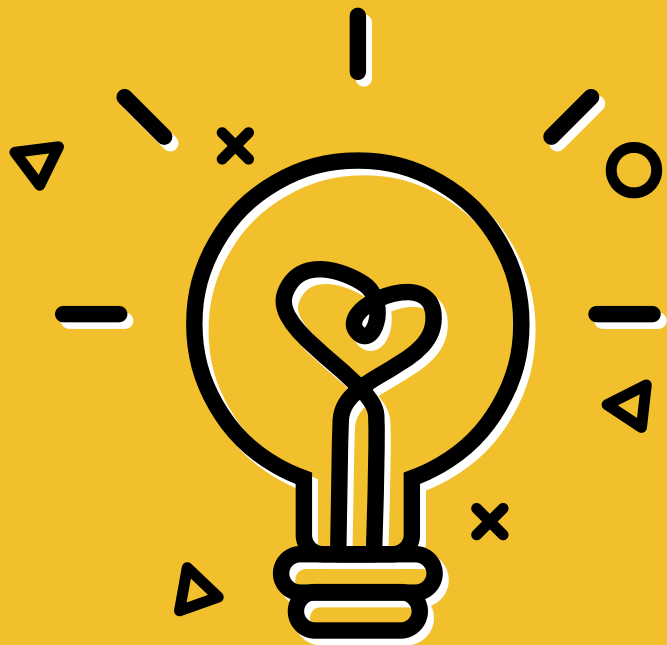




Systemy raportowania błędów

Przeglądanie kodu oraz obsługa błędów

Opracowanie: inż. Grzegorz Petri



„f u cn rd ths, u cn gt a gd jb n sftwr tstng” (Anonymus)



*„If debugging is the process of removing bugs, then
programming must be the process of putting them in.”
(Edsger Dijkstra)*



01

Systemy raportowania błędów

Czyli jak się w tym wszystkim nie pogubić





1) bugzilla

2) lunchpad

3) github

4) gitlab

5) bitbucket

6) jira



7) flyspray

8) ~~google code~~

9) sourceforge

10) trello

11) orts

12) redmine

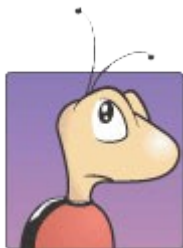


«stackoverflow»





Bugtrackery



bugzilla
bugtracker

Web-serwis

Wersja: 5.0.6

Licencja: MPL

Mozilla

Foundation



Bugzilla - Main Page

[Home](#) | [New](#) | [Browse](#) | [Search](#) | [Search](#) | [\[?\]](#) | [Reports](#) | [New Account](#) | [Log In](#) | [Forgot Password](#)

Welcome to Bugzilla



[File a Bug](#)



[Search](#)



[Open a New Account](#)



[Documentation](#)

Enter a bug # or some search terms [Quick Search](#) [\[?\]](#)

Common Queries:

Bugs reported in the [last 24 hours](#)  | [last 7 days](#) 

Bugs changed in the [last 24 hours](#)  | [last 7 days](#) 

[Home](#) | [New](#) | [Browse](#) | [Search](#) | [Search](#) | [\[?\]](#) | [Reports](#) | [New Account](#) | [Log In](#) | [Forgot Password](#)

Bugzilla Appliance - Powered by [TurnKey Linux](#)





Bugtrackery



lunchpad collaboration

Web-serwis

Wersja: aktywna

Licencja: GNU/GPL

Canonical Ltd.



Launchpad is a software collaboration platform that provides:

- Bug tracking
- Code hosting using Bazaar
- Code reviews
- Ubuntu package building and hosting
- Translations
- Mailing lists
- Answer tracking and FAQs
- Specification tracking
- Take the tour!

Recent Launchpad blog posts

[The information sharing feature is complete](#) – 06 Nov 2012

Launchpad's bug and branch privacy features were replaced by information sharing that permits project maintainers to share kinds of confidential information with people at the project level. No one needs to manage bug and branch subscriptions to ensure trusted users have access to confidential information. The Disclosure features Disclosure is a super feature composed on [...]

[Information sharing is now in beta for everyone](#) – 28 Aug 2012

Launchpad's bug and branch privacy features are being replaced by information sharing that permits project maintainers to share kinds of information with people at the project level. No one needs to manage bug and branch subscriptions to ensure trusted users have access to confidential information. Maintainers can share and unshare their project with people Project [...]

[Project maintainers can see private bugs](#) – 23 Jul 2012

Project maintainers can now see all the private bugs in their project. While Launchpad tried to ensure the proper people could see private bugs in the past, the old subscription mechanism was brittle. Users could unsubscribe themselves and lose access, or retarget a bug to another projects which does not update bug subscriptions. The Purple [...]

Search Launchpad

34,305 projects, 1,284,452 bugs, 796,133 branches, 2,376,074 translations, 247,862 answers, 59,108 blueprints, and counting...

Get started

Creating an account allows you to start working within Launchpad.

Learn more about Launchpad in the [user guide](#) or try it for yourself in our sandbox environment.

Featured projects



The Zeitgeist Project

The Zeitgeist event aggregation framework and related projects.

Bazaar

Do

Drizzle

gDesklets

Gufw

Inkscape

MySQL

OpenStack

Shutter

Steadyflow

Terminator

Ubuntu

Zope.org

DHIS

Docky

Exaile

The Gearman Project

Gwibber

Linaro

OpenShot Video Editor

SchoolTool

Silva CMS

Stoq

Tomdroid

Unity

[Browse all 34305 projects!](#)



Bugtrackery



GitHub

prj dev mng bugtrc
Website

Wersja: aktywna

Licencja: komercyjna

Microsoft

The screenshot shows the GitHub repository page for `jasonetco/octocat-classifier`. The repository has 4 watchers, 2 stars, and 0 forks. The `Code` tab is selected, showing a file list and the `README.md` content.

File List:

File	Commit Message	Time Ago
assets	Support Octocats shared on Twitter	2 minutes ago
tests	Support Octocats shared on Twitter	2 minutes ago
LICENSE	Init to win it	2 minutes ago
README.md	Updated README.md	2 minutes ago
index.js	Created index.js	2 minutes ago
package.json	Bump lodash from 4.17.19 to 4.17.20	2 minutes ago

README.md Content:

Octocat Classifier 🐱🔍

build passing coverage 90% dependencies up to date

As the name suggests, Octocat Classifier is used to determine whether a given image contains an Octocat. It is trained with images from the [Octodex](#), images shared with [#MyOctocat](#) on Twitter, and [photographs of laptops with stickers on them](#).

Installation

```
git clone https://github.com/jasonetco/octocat-classifier
```





Bugtrackery



Bitbucket

prj dev mng bugtrc
Website

Wersja: aktywna

Licencja: komercyjna

Atlassian

The screenshot shows the Bitbucket web interface. On the left is a blue sidebar with the Bitbucket logo and navigation links: My work, Repositories, Projects, Pull requests, Issues, and Snippets. The main content area is titled 'Repositories' and features a search bar and filters for Project, Owner, and Watching. Below this is a table of repositories.

Repository	Project	Owner	Privacy
1up Last updated 28 minutes ago	Project: Mars	Kevin Cambell	
3rd_library Last updated an hour ago	Ecosystem	Fabric-Dev	
3rd_party-analytics-plugins Last updated yesterday	Ecosystem	Fabric-Dev	
a11-dashboard-input Last updated 29 Sep 2017	Saturn Shuttle Lines	Saturn Labs	
a11-dashboard-webservice Last updated 7 Sep 2017	Spacez Plus	William Smith	
aac-maintenance-page Last updated 6 Sep 2017	CST Site reliability engineering	Jenifer Evans	
a11-dashboard-input Last updated 29 Sep 2017	Nucleus Cloud	Saturn Labs	
Analytics Last updated 12 Sep 2017	Nucleus	Fabric-Dev	





Flyspray

prj dev mng bugtrc

Web-serwis

Wersja: aktywna

Licencja: LGPL

Flyspray

Development Team



Pokaż zgłoszenie #

Admin (Nux) | Moje wyszukiwania | Narzędzia administratora | Wyloguj się

Nowy projekt | Przełącz | Przegląd | Lista zgłoszeń | Dodaj nowe zgłoszenie | Log zgłoszeń | Plan działań | Zarządzaj projektem

To jest nowy projekt, a to co widzisz, to tekst wprowadzony przez administratora tego projektu.
Administrator może zabrać Ci zabawki, więc lepiej mu nie podpadać 😊.

FS#3 - Mój rdzeń nie działa

Przydzielony do projektu: [Nowy projekt](#)
Zgłoszone przez Anonimowy zgłaszający - Sunday, 26 June 2011, 02:49 GMT

Typ zgłoszenia	Bug Report
Kategoria	Rdzeń
Status	Unconfirmed
Przydzielone do	Nikogo
System operacyjny	All
Ważność	Krytyczna
Priorytet	Normalny
Rozpoznane w wersji	1.0.0
Do realizacji w wersji	Nie zdecydowano
Do realizacji do dnia	Nie zdecydowano
Procent zaawansowania:	
Głosy	0 (Dodaj głos)
Prywatne	Nie (Zmień na prywatne)
Obserwowanie	Nie (Obserwuj zgłoszenie)

Opis

Kroki do powtórzenia

1. Zapuszczam rdzeń
2. Przekładam przekładnię
3. Wypisuję wypis

Powtarzalność: Problem występuje zawsze

Otrzymany wynik

Wypis jest zapisany.

Spodziewany wynik

Wypis powinien być wypisany.

Zgłoszenie zależy od

Komentarze (0) | Powiązane zgłoszenia (0/0) | Powiadomienia (0) | Przypomnienia (0) | Historia





Bugtrackery



Jira

prj dev mng bugtrc

Web-serwis

Wersja: 7.4.1

Licencja: komercyjna

Atlassian



Jira

prj dev mng bugtrc

Web-serwis

Wersja: 7.4.1

Licencja: komercyjna

Atlassian





Bugtrackery



RedMine

prj dev mng bugtrc

Web-serwis

Wersja: 4.2.1

Licencja: GNU/GPLv2

Jean-Philippe Lang

Name	Issue amount	Issue progress	Estimated hours (open)	Remaining budget
AlphaNodes	2	100%	1573.00	€23,340.00
Product: DB	6	20%	176.00	
Product: Passwords	4	0%	90.00	
Product: Reporting	35	0%	1310.00	€58,340.00
test 1	0	0%	0.00	
Hello World	0	0%	0.00	
00 Playground	5	0%	80.00	€1,700.00
Prueba de budget	0	0%	0.00	€7,600.00
Prueba de Proyecto	3	0%	100.00	€0,200.00
Test project #01	2	0%	200.00	





Bugtrackery



sourceforge
project mngmnt

Website

Wersja: aktywna

Licencja: komercyjna

VA Software

The screenshot shows the SourceForge homepage. At the top, there's a navigation bar with links: Browse, Blog, Deals, Help, Create, Join, Login. Below this is a secondary navigation bar with categories: Articles, Cloud Storage, Business VoIP, Internet Speed Test, and social media icons for Twitter, Facebook, Google+, and LinkedIn. A search bar is on the right. The main header features the SourceForge logo and the text "The Complete Open-Source Software Platform". Below this, it says "Create, collaborate & distribute to over 33 million users worldwide" and "This Week: 25,985,057 Downloads 20,769 Code Commits". A large search bar is in the center. The page is divided into two main sections: "Make Your Projects Come To Life" and "For Developers By Developers". The first section describes the platform's reach and provides a "Join & Create" button. The second section describes the community resource and provides a "Browse" button. The background of the lower section has faint icons of a bug, a gear, and a question mark.

SOURCEFORGE

Browse Blog Deals Help Create Join Login

Articles Cloud Storage Business VoIP Internet Speed Test

Twitter Facebook Google+ LinkedIn

Search for software or solutions

The Complete Open-Source Software Platform

Create, collaborate & distribute to over 33 million users worldwide

This Week: 25,985,057 Downloads 20,769 Code Commits

Search for Software, Business Solutions, or Resources

Make Your Projects Come To Life

With the tools we provide, developers on SourceForge create powerful software in over 430,000 projects; we host over 3.7 million registered users. Our popular directory connects over 33 million visitors and serves more than 4.5 million software downloads a day.

[Join & Create](#)

For Developers By Developers

SourceForge is an Open Source community resource dedicated to helping open source projects be as successful as possible. We thrive on community collaboration to help us create a premiere resource for open source software development and distribution.

[Browse](#)



02 Wytyczne raportowania błędów

Czyli jak zrobić TO dobrze i dostać TO czego oczekujemy





- Pomocne zasady



- 1) Zasada pewności
- 2) Zasada pojedynczości
- 3) Zasada tytułu
- 4) Zasada duplikatu
- 5) Zasada prostoty
- 6) Zasada wagi
- 7) Zasada pilności
- 8) Zasada nieraportowania
- 9) Zasada niereprodukowalności





Zasada pewności

Dlaczego:

Jednym z najczęstszych błędów popełnianych przez testerów jest raportowanie błędów, które nie są błędami.

Wynika to z naszego niezrozumienia kontekstu lub nieznajomości specyfikacji.

Przed zaraportowaniem błędu, poprzez analizę, odpytanie eksperta lub też eksperyment, musimy więc odpowiedzieć sobie na pytanie, czy to na pewno jest błąd.



Rozwiązanie:

- analiza błędu
- upewnienie się, że to błąd oprogramowania.

Wyjątek:

Jesteśmy niedoświadczonymi jeszcze testerami, a nasze błędy będą przechodziły przez dodatkowy punkt weryfikacji.





Zasada pojedynczości

Dlaczego:

Jeśli raport zawiera opis wielu zgłoszeń, to naprawienie jedynie części z nich zaczyna generować problemy.

Nie możemy zamknąć zgłoszenia, ponieważ nie zostało rozwiązane w całości.

Musimy przekazać informację o konieczności retestu, ale jak to zrobić jeśli nie możemy go przekazać w części do testera?

Poza tym tracimy informację o metrykach. Co nam da informacja o ilości zgłoszonych defektów jeśli nie wiemy, ile w jednym zgłoszeniu może być defektów.

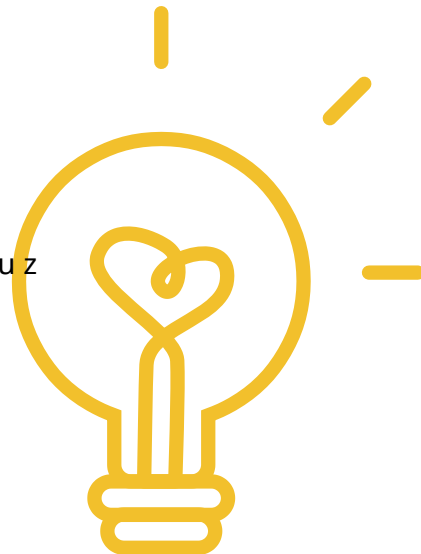


Rozwiązanie:

- Jeden błąd = jeden raport.

Wyjątek:

Jeśli poprawka dla defektu może zostać zrobiona za jednym razem, to możemy raportować je łącznie. Przykładowo możemy zaraportować komunikat błędu z kilkoma literówkami.





Zasada tytułu



Dlaczego:

Tytuł to element zgłoszenia, który będzie czytany najczęściej i musi jednoznacznie wskazywać, czego dotyczy błąd.

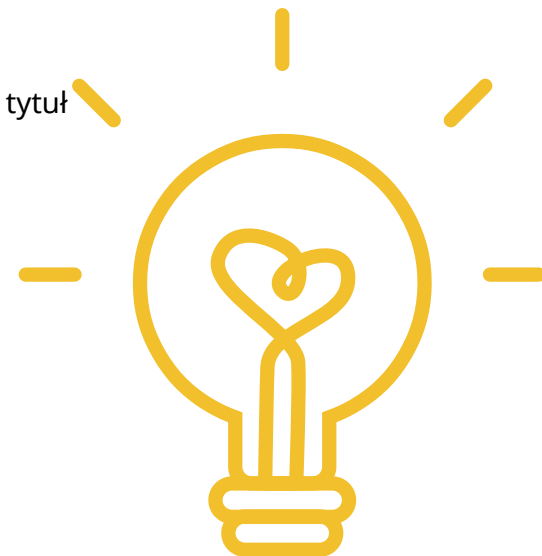
Kiedy ktoś będzie przeszukiwał bazę w poszukiwaniu defektów po to, aby np. by nie popełnić duplikatu, najczęściej przejrzy tytuły zgłoszeń.

Rozwiązanie:

- Najważniejszym elementem raportu jest tytuł
- Tytuł ma być prosty i jednoznaczny

Wyjątek:

BRAK





Pomocne zasady (4)



Zasada duplikatu



Dlaczego:

To kolejny przykład nieefektywności testerskiej.

Zgłaszanie po wielokroć tego samego błędu skutkuje nadmiarową pracą dla innych członków zespołu.

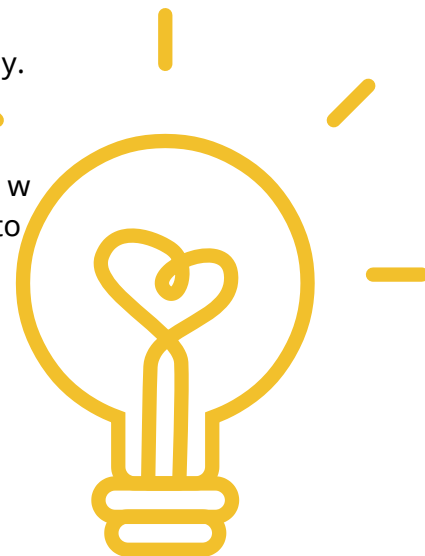
Przejrzyj repozytorium wszystkich defektów (np. trzy wyszukania) po słowach kluczowych błędu i upewnij się, że go tam nie ma.

Rozwiązanie:

- Nie raportuj błędu, który już został zaraportowany.

Wyjątek:

Jeśli coś wygląda jak duplikat, ale znajduje się w innej części aplikacji lub ma odmienne zachowanie, to może nie mieć tej samej przyczyny źródłowej. Zaraportuj z dopiskiem „podobny do zgłoszenia X”.





- Pomocne zasady (5)

Zasada prostoty

Dlaczego:

Nie masz czasu na rozpisywanie się, a programista nie ma czasu na czytanie elaboratów.

Krótki opis i zwięzłe kroki reprodukcji to podstawa. Nie tworzymy epopei narodowej - powtórzenia lub równoważniki zdań są bardziej niż mile widziane.

Rozwiązanie:

- Raport błędu w żołnierskich słowach powinien przekazać, co się wydarzyło i jak do tego dojść.

Wyjątek:

Jeśli raportujemy do osoby nietechnicznej (np. kogoś „z biznesu”), która może nie zrozumieć języka technicznego, musimy dodać więcej szczegółów.





Zasada wagi

Dlaczego:

Podanie wagi przez testera już przy pierwszym raportowaniu przynosi wymierne korzyści i możliwość sortowania zgłoszeń od tych, które otrzymały największą wagę. Jest to pierwsza estymacja (szacowanie) i ustalenie priorytetów.

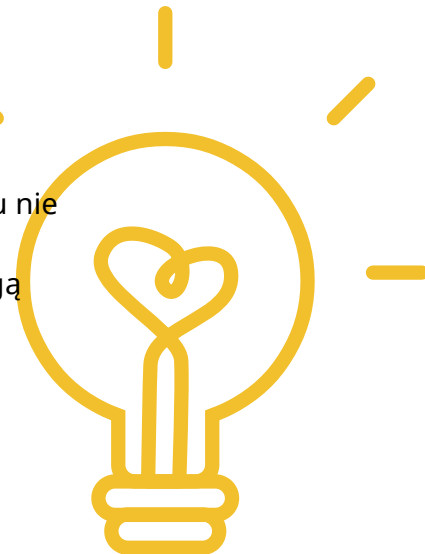
Jeśli tester ma i wiedzę biznesową, i techniczną, jego zgłoszenia wnoszą dużą wartość do dalszej pracy nad błędem.

Rozwiązanie:

- Waga jest ważna
- Powinna być częścią opisu błędu.

Wyjątek:

Jeśli uznajemy, że w naszym oprogramowaniu nie ma miejsca na żadne błędy, to nie ma większego znaczenia, w jakiej kolejności będą naprawiane. Mogą być więc niewartościowane wagą.





Zasada pilności



Dlaczego:

W projektach ważne jest, aby szybko dostarczać informacje o jakości.

Jeśli w oprogramowaniu pojawia się błąd wielkiej wagi, to chcemy poinformować o nim możliwie jak najszybciej, aby szybko go naprawić.



Rozwiązanie:

- Pilny defekt powinien być zgłoszony pilnie.

Wyjątek:

Gdy wiemy, że nasze zgłoszenie nie będzie szybko odczytane (np. raportujemy w weekend, gdy nikt nie pracuje), możemy odłożyć raportowanie w czasie.





Zasada nieraportowania

Dlaczego:

Choć w wielu źródłach przeczytacie, że wszystkie rzeczy należy zgłaszać, to jest to zazwyczaj opinia teoretyków. W praktyce nie zawsze będzie czas na raportowanie, nie zawsze będzie czas na naprawianie.

Część defektów zdewaluuje się zanim je zgłosimy. Część jest tak trywialnych, że zapewne nikt ich nie zauważy, oprócz wprawnego oka testera.

Jeśli masz czas i nieskończony budżet, raportuj wszystko. Jeśli jednak masz ograniczenia, raportuj tak, by uzyskiwać poprawki.

Rozwiązanie:

- Nie każdy błąd musi być raportowany.

Wyjątek:

Jeśli mamy małą wiedzę o tym, co dzieje się w oprogramowaniu i nie do końca możemy ocenić, co jest ważne, a co nie, wtedy raportujemy wszystko.





Zasada niereprodukowalności



Dlaczego:

Defekty niereprodukowalne to często symptom czegoś naprawdę poważnego.

Znalezienie czegoś takiego powinno zaświecić testerowi czerwoną lampkę w głowie i zachęcić do zbadania problemu.



Rozwiązanie:

- Nie zawsze da się zreprodukować defekt, ale to nie znaczy, że nie należy próbować.

Wyjątek:

Możemy nie raportować defektu jeśli mamy uzasadnione przekonanie, że niekoniecznie wynika on z błędu oprogramowania, np. ktoś mógł wrzucić aktualizację na środowisko testowe.





Issues?

report@gplweb.pl
edu.gplweb.pl