

# Bugzilla - wybrane narzędzie zarządzania błędami

Kamil Anikiej,  
Piotr Dittwald,  
Karol Kurach,  
Krzysztof Pawłowski

Uniwersytet Warszawski

29 IV 2008

# System zarządzania błędami

- ang. *Bug-tracking system*
- aplikacja przeznaczona do zapewnienia jakości wytwarzanego oprogramowania poprzez utrzymywanie porządku w zgłaszanych błędach i zadaniach
- przeważnie zintegrowana z innymi narzędziami do zarządzania jakością

## System zarządzania błędami - c.d.

- opiera się na bazie danych, przechowującej informacje o błędach (np. czas zgłoszenia, sugestie dotyczące naprawy błędów, informacje identyfikujące autora zgłoszenia)
- typowo:
  - wspiera tzw. cykl życia błędu (proces podczas którego zgłoszony błąd posiada jeden ze zdefiniowanych statusów pozwalających określić etap przetwarzania błędu)
  - pozwala administratorom nadzorować dozwolone zmiany statusów pomiędzy stanami

# System zarządzania błędami - użycie

- 1 generowanie raportów odnośnie skuteczności programistów przy rozwiązywaniu problemów (oczywiście raporty te wymagają należytej interpretacji - różne błędy mają różny stopień złożoności)
- 2 Local bug trackers (LBT's)
  - programy komputerowe używane do komunikacji pomiędzy pracownikami działu pomocy bezpośredniej (*help desk*) a programistami
  - pozwalają na komunikację w języku wygodnym dla każdej ze stron

# Przykłady narzędzi do zarządzania błędami

## 1 IBM Rational ClearQuest

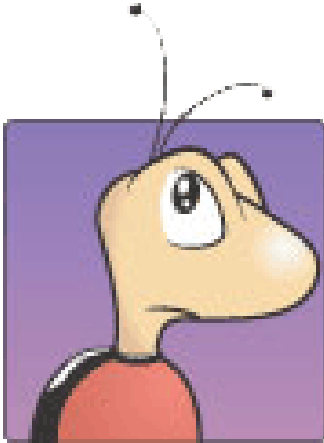
- m.in. Linux, SUN Solaris, Windows
- interfejs WWW
- integracja IDE takimi, jak np. z Eclipse i Microsoft .NET.
- rozbudowana, elastyczna obsługa przepływu pracy (dostarczany z predefiniowanymi przepływami odzwierciedlającymi podstawowe procesy zmian w organizacji)
- łatwa konfiguracja

# Przykłady narzędzi do zarządzania błędami - c.d.

## 1 JIRA

- stworzona na bazie J2EE
- zarządzanie błędami, zadaniami i projektami
- rozszerzalność i integracja z innymi systemami (email, RSS, Excel, XML ...)
- działa na wielu (cyt. *almost any*) systemach operacyjnych
- używana przez ponad 7.700 organizacji z 90 krajów

# Wprowadzenie



- oprogramowanie stworzone i używane przez Fundację Mozilla
- *open source* software
- używana zarówno przy projektach komercyjnych jak i niekomercyjnych
- porównuje się do niej inne systemy zarządzania błędami (jest więc w pewnym sensie standardem)

## Krótką historia

- 1998r. - Bugzilla, napisana przez Terry'ego Weissmana (w języku skryptowym TCL) dla powstającego projektu Mozilla.org
- Bugzilla 2.0 - rezultat przepisania na język Perl, wersja udostępniona użytkownikom przez anonimowe CVS
- pod zwierzchnictwem Tary Hernandez (od IV 2000r.) rozwój Bugzilli przybrał bardziej społecznościowy charakter - większą odpowiedzialność za projekt wzięli na siebie zwykli użytkownicy
- od 2001r. projekt nadzoruje Dave Miller





## Główne cechy

- duża stabilność

## Główne cechy

- duża stabilność
- zintegrowany schemat bezpieczeństwa

## Główne cechy

- duża stabilność
- zintegrowany schemat bezpieczeństwa
- zależności między błędami; grafy błędów

## Główne cechy

- duża stabilność
- zintegrowany schemat bezpieczeństwa
- zależności między błędami; grafy błędów
- zaawansowane funkcje związane z raportowaniem błędów

## Główne cechy

- duża stabilność
- zintegrowany schemat bezpieczeństwa
- zależności między błędami; grafy błędów
- zaawansowane funkcje związane z raportowaniem błędów
- szybki, stabilny system RDBMS

## Główne cechy

- duża stabilność
- zintegrowany schemat bezpieczeństwa
- zależności między błędami; grafy błędów
- zaawansowane funkcje związane z raportowaniem błędów
- szybki, stabilny system RDBMS
- rozbudowana konfiguracja

## Główne cechy

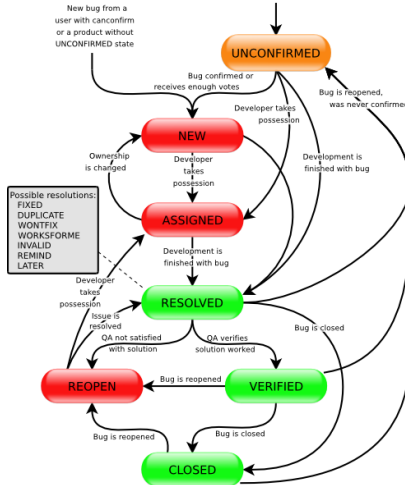
- duża stabilność
- zintegrowany schemat bezpieczeństwa
- zależności między błędami; grafy błędów
- zaawansowane funkcje związane z raportowaniem błędów
- szybki, stabilny system RDBMS
- rozbudowana konfiguracja
- API do e-maili, XMLa, konsoli i HTTP

## Główne cechy

- duża stabilność
- zintegrowany schemat bezpieczeństwa
- zależności między błędami; grafy błędów
- zaawansowane funkcje związane z raportowaniem błędów
- szybki, stabilny system RDBMS
- rozbudowana konfiguracja
- API do e-maili, XMLa, konsoli i HTTP
- możliwa integracja z automatycznymi aplikacjami konfiguracji zarządzania, takimi jak Perforce czy CVS (interfejs e-mail Bugzilli oraz skrypty zapisu/odczytu)



# Cykl życia błędu



## Cykl życia błędu - statusy

- UNCONFIRMED - błąd został zgłoszony, ale nie zweryfikowano jego istnienia
- NEW - błąd został dodany do bazy danych i potwierdzono jego istnienie
- ASSIGNED - błąd nie został jeszcze rozwiązany, ale przypisano go do właściwej osoby
- REOPENED - błąd został kiedyś rozwiązany, ale okazało się, iż to rozwiązanie nie było prawidłowe
- RESOLVED - rozwiązanie zostało znalezione i oczekuje na akceptację odpowiedzialnej za to osoby; błąd może zostać tu otworzony na nowo (REOPENED) lub zweryfikowany (VERIFIED)

## Cykl życia błędu - rozwiązania

Statusy resolved i verified mają następujące rozwiązania:

- FIXED - rozwiązanie jest sprawdzane i testowane
- INVALID - opisany problem nie jest błędem
- WONTFIXED - opisany problem jest błędem, ale nie zostanie nigdy rozwiązany
- DUPLICATE - zgłoszony błąd jest duplikatem już istniejącego
- WORKSFORME - wszelkie próby odtworzenia błędu nie powiodły się i nie są znane przyczyny wystąpienia danego błędu; gdy pojawią się dalsze informacje, błąd zostanie otwarty ponownie (REOPENED)
- MOVED - dany błąd jest charakterystyczny dla innego produktu i został przeniesiony do bazy danych błędów dotyczących tego produktu

# Wybrane projekty używające Bugzilli

- Projekty Programistyczne

- 1 Mozilla
- 2 Linux Kernel
- 3 Gnome
- 4 KDE
- 5 Apache Project
- 6 Open Office
- 7 Eclipse

## Wybrane projekty używające Bugzilli - c.d.

- Dystrybucje Linuksa
  - 1 Red Hat
  - 2 Mandriva
  - 3 Gentoo
  - 4 Novell
- Organizacje
  - 1 NASA
  - 2 Id Software

## Strona błędu

- Product and Component – określenie jakiego produktu i komponentu (elementu produktu) dotyczy błąd
- Status and Resolution – definiuje aktualny status błędu, począwszy od niepotwierdzonego błędu, poprzez błąd w trakcie rozwiązywania aż do potwierdzonego rozwiązania błędu
- Assigned To – osoba odpowiedzialna za poprawienie błędu
- URL – url związany z błędem, element opcjonalny
- Summary – jednozdaniowy opis problemu
- Status Whiteboard – krótka notatka dotycząca błędu
- Keywords – administrator może zdefiniować słowa kluczowe, które można użyć do etykietowania i kategoryzowania błędów, np. krytyczny błąd, itp.
- Platform and OS – środowisko, w którym błąd został wykryty

## Strona błędu

- Version – wersja produktu lub/i komponentu, do których odnosi się błąd
- Priority – używane do priorytetowania błędów
- Severity – określenia jak poważny jest błąd, od uniemożliwiającego poprawną pracę aplikacji do trywialnego
- Target – numer przyszłej wersji produktu, która będzie pozbawiona rozpatrywanego błędu
- Reporter – osoba opisująca/zgłaszająca błąd ulegnie zmianie
- Time Tracking – śledzenie, szacowanie czasu naprawy błędu

## Strona błędu

- Attachments – pliki dołączone, powiązane z błędem (np. przypadki testowe, łatki, inne pliki)
- Dependences – określone jeżeli błąd nie może być naprawiony dopóki inne błędy nie zostaną naprawione (depends on), lub rozpatrywany błąd blokuje naprawę innych błędów (blocks), wyliczone są numery błędów
- Votes – „głosy” oddane na rozpatrywany błąd przez użytkowników, mogą mówić o ważności naprawy błędu
- Additioal Comments – każdy użytkownik może dodać swoje zdanie do dyskusji na temat błędu



## O czym warto pamiętać

- 1 Zanim zgłosimy błąd sprawdzamy, czy ktoś przed nami nie zgłaszał już tego błędu (np. poprzez *“Query Existing bug reports”*),
- 2 Przed zgłoszeniem błędu warto przeczytać stosowne wytyczne napisane przez opiekunów danej Bugzilli, (przykładowy odsyłacz (*“Before reporting bugs, please ensure that you read the Gentoo Bug Reporting Guide.”*)),
- 3 Aby ułatwić pracę innym warto możliwie dokładnie zlokalizować błąd (programy strace, ggdb itp.),
- 4 Czasem wymagane jest, aby zamieścić dodatkowe informacje (np. w Gentoo zawsze trzeba umieszczać wynik polecenia *“emerge -info”*)

## Zadanie 0

Natrafiliśmy na następujący błąd:

- 1 Korzystamy z dystrybucji Gentoo.
- 2 Zainstalowaliśmy nowe jądro 2.6.25-r1.
- 3 Podczas reinstalacji sterowników karty graficznej ATI natrafiliśmy na błąd kompilacji.

Sprawdźmy na stronie `bugs.gentoo.org`, czy ktoś przed nami nie zgłaszał już tego błędu.

# Zadanie 1

Do wykonania poniższych zadań potrzebne są konta założone na danych stronach. Jeżeli nie posiadamy tam konta, można użyć konta:

login: – login –

pass: – hasło –

- 1 Zadanie - Chcemy do Thunderbirda dodać nową funkcjonalność.

<https://bugzilla.mozilla.org/>

- 2 “It would be useful if a user could define a date variable. Mails older than this variable wouldn't be downloaded”

# Podsumowanie

- W prezentacji wykorzystano:
  - Wikipedię
  - Archiwalne materiały z ZPP
  - Strony producentów
  - [www.bugzilla.org](http://www.bugzilla.org) (w tym zdjęcia)
  - Stronę  
`http://www.gentoo.org/doc/pl/bugzilla-howto.xml`
  - Stronę `http://istyar.net/el/wprow\_bugzilla/html/wprow\_bugzilla-5-1.html`
  - Dokument `http://aragorn.pb.bialystok.pl/~tomluk/docs/io2/bugzilla\_tutorial.pdf`

# Podsumowanie

Dziękujemy za uwagę.