Systemy raportujące błędy (Bugzilla)

Rafał Zieliński

Spis treści:

- Obsługa błędów
- System raportujący błędy
- Ticket tracking
- Bugzilla

Obsługa błędów

- O tym jak ważne jest odnajdywanie i właściwa obsługa błędów w oprogramowaniu przekonali się Francuzi podczas startu rakiety *Ariane 5*, która to spadła po 40 sekundach lotu. Przyczyną było zastosowanie tego samego oprogramowania co przy starszej *Ariane 4*. Spowodowało to przy większej rakiecie przekroczenie zastosowanego typu *short int*, co skutkowało zakończeniem programu i uruchamianiem zastępczych komputerów. Poniesione straty były rzędu 500 milionów dolarów

System raportujący błędy

- W celu śledzenia i zgłaszania napotkanych błędów podczas pracy z danym programem stosuje się tak zwane *Bugtrackery,* czyli systemy do śledzenia błędów. Stosowanie ich pomaga zapewnić wyższą jakość oprogramowania i znajdowanie przez programistów błędów. Regularne używanie *Bugtrackerów* jest jedną z zasad dobrego zespołu programistycznego. Firmy rozwijające oprogramowanie szeroko stosują systemy raportujące błędy. Wśród nich możemy znaleźć Mozille, Facebooka czy dystrybucje Linuxa.

- Ich zastosowanie jest szczególnie popularne w przypadku oprogramowania *open source,* gdyż z faktu wolnego dostępu do kodu aplikacji każdy może sam zaglądać w kod i składać precyzyjne raporty o błędach.
- *Bugtrackery* mogą być traktowane jako rodzaj *Ticket trackingu.*
- Systemy śledzenia błędów zazwyczaj są zintegrowane z inną aplikacją zarządzania projektem.

- *Bugtrackery* dzielimy ze względu na pracujące w poszczególnych architekturach:
  - Architektura *klient-serwer*:
    - Jest to najpopularniejszy rodzaj ze względu na wygodę i bezpieczeństwo przechowywanych danych. W tym przypadku klient zgłasza do serwera żądanie obsługi, a serwer zapewnia mu usługi. Zatem klient żąda dostępu do danej usługi lub zasobu, a serwer udostępnia zasoby czy usługi.
  - Aplikacje *hostowane*:
    - Drugie co do popularności rozwiązanie. W ogólności polega na oddaniu przez osobę hostującą pewnej przestrzeni dyskowej, transferu danych i określonych obsługiwanych usług.

- Architektura *rozproszona*:
  - Najrzadziej spotykane rozwiązanie. W ogólności polega na połączeniu siecią komputerową pewnej ilości komputerów w spójną i logiczną strukturę z zastosowaniem aplikacji do współdzielenia zasobów.
    Aby móc mówić o architekturze rozproszonej musi spełnić pewne założenia:
    - Skalowalność
    - Dzielenie zasobów
    - Tolerancja na awarie
    - Otwartość
    - Współbieżność
    - Przezroczystość

Najistotniejszym elementem jest przezroczystość, czyli cecha utwierdzająca użytkownika w tym, że system jest spójną całością bez zauważalnych podziałów na elementy składowe.

- Komponenty *Bugtrackerów*:
  - Elementem wspólnym na którym bazują *Bugtracery* jest baza danych. To w niej przechowywane są wszelkie dane o błędach. Zwykle taka baza zawiera następujące encje:
    - ID błędu
    - Czas zgłoszenia
    - Priorytet błędu
    - Skutek jaki powoduje
    - Sposób odtworzenia błędu
    - Dane o osobie raportującej
    - Programiści, którzy mogą naprawić błąd

- Najczęściej spotykane *Bugtrackery* to:
  - Architektura *Klient-serwer*:
    - Bugzilla
    - Trac
    - Redmine
    - FogBugz
    - Debbugs
    - Mantis
    - BugTracker.NET
    - Flyspray
    - JIRA

- Aplikacje *hostowane*:
  - CodePlex
  - bugtracker.pl
  - Google Code
  - SourceForge
  - Bitbucket
  - Assembla
  - GNU Savannah
  - Launchpad

- Architektura *rozproszona*:
  - Fossil

Ticker tracking

- Czyli w dosłownym tłumaczeniu „śledzenie biletów”. Stanowią trzon większości systemów obsługi klienta w różnych firmach. W ogólności można powiedzieć, że jest to sposób zarządzania system, który opowiada na masowe zapytania. Zwykle odbywa się to za pośrednictwem drogi mailowej.

- Schemat działania:
  - Do każdego nadchodzącego pytania przypisany jest *bilet*, który zawiera aktualny postęp i archiwum korespondencji.
  - Stworzenie bufora typu LIFO z *biletów*
  - Operator systemu pobiera kolejno *bilety*
  - *Bilet*, który został pobrany jest przypisany do operatora, który od tej pory jest za niego odpowiedzialny
  - W przypadku pomyślnego zakończenia działań związanych z biletem zostaje przeniesiony do listy *„załatwionych”*. Natomiast jeśli operator nie jest w stanie sobie z nim poradzić przejmuje go inny operator. Może również trafić do listy nigdy *niezakończonych* lub *odrzuconych*.

Bugzilla

- Jest to popularne, darmowe i otwarte oprogramowanie w architekturze klient-serwer do tworzenia serwisów internetowych służące do **raportowania błędów.** Aplikacja ta została napisana przez programistów z *Fundacji Mozilla* w języku Perl. Nad polską wersją programu pracuje zespół *Aviary.pl*. Może być również stosowane do aplikacji komercyjnych.

- Bardzo ważnym elementem tego systemu jest to, że zwykle każdy użytkownik może raportować błędy. Następnie dany raport dopinany jest do odpowiedniego programisty, który może zająć się tym problemem. Zgłoszeniom można nadawać różne statusy. Ponadto *Bugzilla* daje możliwość dodawania załączników i umieszczania własnych komentarzy.

- Podczas raportowania błędów w *Bugzilli* mamy możliwość zgłaszania własnych pomysłów i sugestii związanych z rozwojem programu. Niekiedy takie rozwiązanie może okazać się równie istotne jak samo raportowanie błędów. Minusem tego rozwiązania jest możliwość otrzymywania bezużytecznych zgłoszeń, lecz na szczęście takim można przydzielać odpowiedni status.

- Kto korzysta z *Bugzilli*?

Ok. 1300 różnego rodzaju korporacji, organizacji i projektów, a wśród nich można znaleźć Facebook, NASA, Mozilla, Ecplipse, Open Office, Apache, dystrybucje Linux i wiele innych.

Pełną listę można znaleźć pod adresem:

http://www.bugzilla.org/installation-list/

- Czego potrzeba do zainstalowania *Bugzilli* w wersji 2.18:
  - Bugzilla z pakietu bzr (Bazaar)
  - MySQL (5.5.x)
  - ActiveState Perl (5.12.4 lub wyższa) wraz z modułami:
    - AppConfig (1.66)
    - TimeDate (1.19)
    - DBI
    - DBDmysql (4.011)
    - Template-Toolkit (2.22)
    - MailTools (2.04)
    - GD
    - Chart (2.4.1)
    - GDGraph (1.44)
    - PatchReader (0.9.5)
    - Net-LDAP-Express
  - Apache (2.x)

- Skąd pobrać wszystkie komponenty?
  - Bugzilla z Bazaar: http://wiki.bazaar.canonical.com/WindowsDownloads
  - Bugzilla z TarBall: http://www.bugzilla.org/download/#stable
  - MySQL: http://www.mysql.com/downloads/mysql/
  - Perl: http://www.activestate.com/activeperl/downloads
  - Apache: http://httpd.apache.org/download.cgi

- Jak zainstalować i skonfigurować Bugzille pod Windowsem?

  Są dwie możliwości:

  1) Pierwsza z wykorzystaniem gotowej paczki, w której zebrane są wszystkie komponenty potrzebne do instalacji i konfiguracji.
  2) Druga natomiast polega na ręcznej instalacji i konfiguracji każdego elementu.

1) Paczkę, w której znajduje się:
   - Bugzilla
   - MySQL
   - Perl
   - Apache

   można pobrać stąd.

   Do zainstalowania potrzeba ok. 260 MB wolnej przestrzeni.

   Podczas instalacji potrzebny będzie SMTP poczty administratora konta.

   Należy zwrócić szczególną uwagę podczas wypełniania pól w kreatorze konfiguracji serwera.

2) Instalacja i konfiguracja ręczna jest mozolna i skomplikowana. Zdecydowanie lepszym i wygodniejszym rozwiązaniem jest zastosowanie poprzedniego sposobu, czyli użycie gotowej paczki.

   Jeżeli jednak ktoś lubi wyzwania może podjąć się tego sposobu korzystając z tutoriala: https://wiki.mozilla.org/Bugzilla:Win32Install.

- Po zainstalowaniu w katalogu Bugzilli znajdziemy kilka ważnych plików:
  - BugzillaCmdPrompt.cmd – otwiera linie komend skonfigurowaną tak by łatwo można było instalować dodatkowe moduły Perla i wykonywać skrypty Bugzilli
  - BackupBugzilla.cmd – tworzy kopie zapasową bazy danych Bugzilli
  - Checksetup.cmd – podaje wersje zainstalowanych komponentów wraz z wymaganymi modułami Perla

- Dodając nowe zgłoszenie należy wypełnić parę istotnych pól, których znaczenie warto poznać:
  - Severity – ważność zgłoszenia:
    - Blocker – zatrzymuje rozwój i/lub testy
    - Critical – wycieki pamięci, utraty danych, crashe
    - Major – duża utrata funkcjonalności
    - Normal – standardowy błąd, lekka utrata funkcjonalności
    - Minor – pomniejsza utrata funkcjonalności i „proste” błędy
    - Trivial – drobne błędy jak literówki itd..
    - Enhancement – ulepszenia i porady

- Platform – platforma, z której zostało wysłano zgłoszenie:
  - All – zdarza się na każdej platformie
  - Macintosh
  - PC
- Operating System– OS użytkownika zgłaszającego:
  - All – zdarza się na każdym systemie
  - Windows
  - Mac OS
  - Linux

- Gdy edytujemy raport mamy do dyspozycji wiele nowych pól.
  - CC List - dodanie użytkowników, którzy mają otrzymywać maile w momencie dokonania zmiany.
  - Orig. Est. – oryginalny szacowany czas
  - Current Est. – aktualnie szacowany czas
  - Hours Worked – przepracowane godziny
  - %Complete – ile procent jest wykonane
  - Gain – liczba godzin wyprzedzenia Orig. Est.
  - Deadline – termin zakończenia pracy nad błędem

- Depends on – numery błędów, które trzeba naprawić aby ten można było poprawić
- Blocks – numery błędów zatrzymanych przez ten błąd
- URL – URL związane z tym błędem

- Należy również pamiętać o tym, że przed dodaniem nowego raportu powinno się sprawdzić, czy nie występuje już taki temat klikając na *Most Frequently Reported Bugs* w zakładce *New.*
- Bugzilla oferuje wiele opcji związanych z przeglądaniem błędów. Gdy wejdziemy w zakładkę *Browse* i wybierzemy niepusty projekt zobaczymy parę użytecznych opcji:
  - Long Format – nieedytowalna lista podsumowań każdego błędu
  - XML – lista błędów w formacie XML
  - CVS – eksport listy błędów do arkusza danych

- Feed – subskrypcja kanału
- iCalendar – eksportuje listę błędów do iCalendara
- Change Columns – edycja wyświetlanych kolumn
- Change Several Bugs at Once – edycja wszystkich raportów na raz, gdy jest ich więcej niż jeden
- Remember search – zapisane wyniku wyszukiwania do stopki

- W zakładce *Search* mamy dwie pod zakładki, *Simple Search*, które jak sama nazwa wskazuje służy do prostego i szybkiego wyszukiwania. Jak i *Advances Search*, w którym znajdziemy szereg opcji dzięki którym sprecyzujemy nasze zapytanie.

- Po wejściu w zakładkę *Reports* mamy możliwość wybrania spośród dwóch reprezentacji zestawienia błędów:
  - Raporty w formie bazodanowej.
  - Wykresy, których przebieg zmienia się w dziedzinie czasu. W tym przypadku dane można reprezentować w jednym, dwóch lub trzech wymiarach. Standardowo są w formie tabelarycznej (*Table*) Ponadto po doinstalowania pakietu graficznego jest możliwość reprezentacji w formie wykresu kołowego (*Pie*), słupkowego (*Bar*) lub liniowego *(Line*). Dane można również eksportować do arkusza kalkulacyjnego (*CSV*).

- Bugzilla daje nam do dyspozycji szereg ustawień, gdy wejdziemy w zakładkę *Preferences*:
  - General Preferences:
    - Bugzilla's general appearance (skin) – zmiana skórki
    - Quote the associated comment when you click on its reply link – zachowanie się podczas cytowania
    - Timezone used to display dates and times
    - Language used in email – wybór języka mailów
    - After changing a bug – wyświetlana strona po zmianach
    - Enable tags for bugs – tagowanie błędów
    - Zoom textareas large when in use (requires JavaScript) – automatyczne rozszerzenie obszarów tekstowych
    - Field separator character for CSV files – separator w eksportowane liście błędów CSV

- Automatically add me to the CC list of bugs I change – automatyczne dodawanie do listy CC
- When viewing a bug, show comments in this order – kolejność wyświetlania komentarzy
- Show a quip at the top of each bug list –

- Email Preferences:
  - Reporter – gdy jesteś osobą zgłaszającą błąd
  - Assignee – gdy jesteś osobą wyznaczoną jako opiekun błędu
  - CCed – jesteś na liście CC
  - Users Warching – wysyłanie kopi wszystkich bugmaili podanych kont użytkowników

- Save Searches – można zobaczyć i włączyć wszystkie zapisane wyszukiwania jakie stworzył właściciel konta lub członkowie grupy, do której należy. Zapisane wyszukiwania mogą być dodane do stopki strony.

- Name and Password – zmiana danych personalnych, hasła i adresu e-mail.

- Permissions – informacje na temat posiadanych oprawnień.

- Jeżeli posiadamy konto administratora mamy również do dyspozycji zakładkę *Administration,* w której znaleźć można wiele zaawansowanych ustawień:
  - Parameters – ustawienia parametrów instalacji. Można tu na przykład określić URL dostępu do tej instalacji, uwierzytelnianie użytkowników, mail transfer, ustawienia wykresów i wiele innych.
  - Default Preferences – domyślne preferencje użytkowników, które zostaną użyte jako standardowe dla wszystkich. Każdy później będzie mógł edytować swoje ustawienia.
  - Sanity Check – uruchamia testy wykonujace diagnozę pod kontem poszukiwania błędów w bazie danych.

- Users – edycja i dodawanie kont użytkowników, wraz z przydziałem do grup.
- Products – edycja i dodawanie produktów wraz z ograniczeniami grupowymi i ustaleniem dostępu do błędów.
- Flags – edycja i dodawanie flag czyli zazwyczaj czteroargumentowe pola, będącymi oznaczeniami identyfikującymi czy błąd lub załącznik ma przyznany czy odrzucony status.
- Custom Fields – dodawanie nowych niestandardowych pól, które to można następnie używać przy tworzeniu błędów i wyszukiwać po ich zawartości. Zbyt duża liczba pól odbije się kosztem zwiększenia złożoności interfejsu.

- Field Values – dodawanie i usuwanie wartości pól.
- Bug Status Workflow – początkowe statusy tworzonych błędów i dozwolone przejścia statusu błędu podczas edycji istniejących błędów.
- Gropus – dodawanie i usuwanie grup użytkowników wraz z możliwością określania praw.
- Keywords – dodawanie i usuwanie słów kluczowych, które pomagają przy znajdywaniu zapytań.
- Whining – tworzenie zapytań, które będą rozsyłane mailem w określonym czasie do użytkowników.