



# Fall 2020 Mid-Term Assignment

## *Software Verification and validation*

**Submitted To:** Mr. Zain Shaukat (Lecturure)

**Submitted By:** Muhammad Zain Ul Abideen

ID# 13740

BSSE (6th Semester)

# Testing Tool: Bugzilla

## Part (a): Pros and Cons of Bugzilla

### Pros of Bugzilla:

The Pros of Bugzilla are as follow:

1. The Bugzilla has an optimized database structure which provides us increased performance and scalability.
2. Bugzilla is a much secured which provides us confidentiality.
3. Bugzilla has integrated Email capability.
4. Bugzilla is widely used bug tracker because it is open source.
5. Bugzilla is easy in usage because it is having an easy interface a non-technical person can easily understand it.
6. Bugzilla has a good documentation part, which can filter out all bugs as per its status. In addition, the report is very beautiful and very easily readable.
7. Bugzilla has an advance search option, which is very useful to filter the logged defects.
8. It is a great tracking tool for managing multi-project level defect tracking

### Cons of Bugzilla:

The Cons of Bugzilla are as follow:

1. The Bugzilla isn't liked by all the testers because its interface is much simple and sophisticated it needs to be a bit improved.
2. Bugzilla has a lot of defects in its code which needs to be fixed.
3. Bugzilla has the biggest con as it can't be a part of Agile methodology as no spirit association can happen in Bugzilla.
4. Bugzilla has a limited number of plugins which is not a good experience.
5. Bugzilla needs improvement for reporting and filtering area.
6. Bugzilla has also performance issues.

## **Part (b) Functionality of Bugzilla.**

# **Functionality:**

## **For Users**

- Advanced Search Capabilities
- Email Notifications Controlled By User Preferences
- Bug Lists in Multiple Formats (Atom, iCal, etc.)
- Scheduled Reports (Daily, Weekly, Hourly, etc.) by Email
- Reports and Charts
- Automatic Duplicate Bug Detection
- File/Modify Bugs By Email
- Time Tracking
- Request System
- Private Attachments and Comments
- Automatic Username Completion or Drop-Down User Lists
- Patch Viewer
- "Watch" Other Users
- Move Bugs Between Installs
- Save and Share Searches

## **For Administrators**

- Excellent Security
- Extension Mechanism for Highly Customizable Installations
- Custom Fields
- Custom Workflow
- Full Unicode Support
- Localization
- mod\_perl Support for Excellent Performance
- Web services (XML-RPC) Interface
- Control Bug Visibility/Editing with Groups
- Impersonate Users
- Multiple Authentication Methods
- Support for Multiple Database Engines
- Sanity Check

## **Part (c) Supporting Languages of Bugzilla:**

# **Supporting Languages**

Bugzilla support the following languages:

- Perl with Catalyst
  - DBIx Class for ORM.
- Perl with CGI Application
  - Easy short-term goal, stepping stone towards Frameworks like Catalyst, Maypole, Jifty, etc
- Ruby with Rails
  - Also investigate Hobo.
  - Want to look at Globalize for i18n.
- Python with Pylons
  - Using SQLAlchemy
  - Mako for templating, or use the upcoming Python TT port.
  - Examine ToscaWidgets and FormEncode
  - Possibility of using AuthKit
- PHP with CakePHP
  - The "serious PHP framework" space is boiling down to one of two these days

## **Part (d) Supporting Tests of Bugzilla:**

# **Supporting Test:**

## **1. Functionality Testing:**

This is used to check if your product is as per the specifications you intended for it as well as the functional requirements you charted out for it in your developmental documentation. Web based Testing Activities includes:

Test all links in your webpages are working correctly and make sure there are no broken links. Links to be checked will include -

- Outgoing links
- Internal links
- Anchor Links
- MailTo Links

## 2. Usability testing:

Usability Testing has now become a vital part of any web based project. It can be carried out by testers like you or a small focus group similar to the target audience of the web application.

**Test the site Navigation:**

- Menus, buttons or Links to different pages on your site should be easily visible and consistent on all webpages.

## 3.Interface Testing:

Three areas to be tested here are - Application, Web and Database Server

- **Application:** Test requests are sent correctly to the Database and output at the client side is displayed correctly. Errors if any must be caught by the application and must be only shown to the administrator and not the end user.
- **Web Server:** Test Web server is handling all application requests without any service denial.
- **Database Server:** Make sure queries sent to the database give expected results.

## 4. Database Testing:

Database is one critical component of your web application and stress must be laid to test it thoroughly. Testing activities will include-

- Test if any errors are shown while executing queries
- Data Integrity is maintained while creating, updating or deleting data in database.
- Check response time of queries and fine tune them if necessary.
- Test data retrieved from your database is shown accurately in your web application

## 5. Compatibility testing.

Compatibility tests ensures that your web application displays correctly across different devices. This would include-

**Browser Compatibility Test:** Same website in different browsers will display differently. You need to test if your web application is being displayed correctly across browsers, JavaScript, AJAX and authentication is working fine. You may also check for mbile Browser Compatibility.

The rendering of web elements like buttons, text fields etc. changes with change in **Operating System**. Make sure your website works fine for various combination of Operating systems such as Windows, Linux, Mac and Browsers such as Firefox, Internet Explorer, Safari etc.

## 6. Performance Testing:

This will ensure your site works under all loads. Software Testing activities will include but not limited to -

- Website application response times at different connection speeds
- Load test your web application to determine its behavior under normal and peak loads
- Stress test your web site to determine its break point when pushed to beyond normal loads at peak time.
- Test if a crash occurs due to peak load, how does the site recover from such an event
- Make sure optimization techniques like gzip compression, browser and server side cache enabled to reduce load times

## 7. Security testing:

security testing is vital for e-commerce website that store sensitive customer information like credit cards. Testing Activities will include-

- Test unauthorized access to secure pages should not be permitted
- Restricted files should not be downloadable without appropriate access
- Check sessions are automatically killed after prolonged user inactivity
- On use of SSL certificates, website should re-direct to encrypted SSL pages.

### Part (e) Finding Bug in code:

```
1 #include<stdlib.h>
2 void match_or_mismatch()
3 {
4     char buf[10];
5     double* flt_ptr = new double;
6     int* numarray = new int[20];
7     free(buf);
8     free( flt_ptr);
9     delete numarray;
10 }
```

### Bug:

Freeing mismatched the memory from “flt\_ptr”. Memory referenced by “flt\_ptr” allocated through function ‘new’ at line 5 illegally freed by passing argument 1 to function “free” at line 8.