

# TULEAP: THE BUG TRACKER (IJSETR)

Mansi Goel, Priyanka Jain

**Abstract**— this paper introduces Tracker via Tuleap which is fully free, open source tool to keep record of bugs, defects, risks, security tasks etc. We customized a Bug Tracker in Tuleap. Bug tracker should be used in the organizations. In this paper we introduces various advantages and importance of customization of Bug Tracker in the organizations. Bug tracker is used to find defects in the project and always depends upon the organization. Why to go for already existing trackers when Tuleap is providing a customization feature.

**Index Terms**— ALM (Application life cycle management), Artifacts, Bugs, Bug Tracker, CSV (Comma separated value).

## I. INTRODUCTION

Tuleap is a free open tool for Application Life Cycle Management. It can be used for Planning, Tracking, Tracing projects. It allows tracking change management and various artifacts like bugs, tasks, requirements, etc. We can create and customized any number of trackers in one project. Tracker service can artifact any item which is generic such as bug, security, tasks, risk etc.

Bug Tracker is an open source tool available in the market. It compromises with two words, “Bug” and “Tracker”. Bug is a terminology which is used by “tester” while Tracker is used to track those bugs. During development of the project using Software Development Life cycle there could be bug in any phase.

Before deployment of any project it becomes important to track the bugs otherwise it could be the reason of loss in the business or users can switch to another organization for their products due to unsatisfactory results. And tracking is also important security point of view. We can customize these trackers according to our organization policies or requirements. It provides a facility to modify each and everything according to our need. We can also create a replica for this kind of tracker. Defects tracking depend upon the project. We can use various templates which are predefined; can insert our own template as well or add fields according to our need. [1]

## II. WAY OF CUSTOMIZING A TRACKER

There are various ways available for customizing a tracker. But the best way is that we should use our own template to customize it according to organization requirement.

*Manuscript received Jan , 2014.*

*Mansi Goel, AIMNACT, Banasthali University, . Jaipur, India, 08860196427,*

*Priyanka Jain, AIMNACT, Banasthali University, Jaipur, India, 09871686405.*

### A. Create a new Tracker

To track a bug firstly registers a project into the Tracker. Registration of project is required to create a tracker into that particular project, there is an administrator for that tracker, and He is only the responsible person who would be having rights to access that tracker. [3]

After creating a Tracker administrator can modify following things, according to the organization requirement Final Stage:

### Create a new tracker

First, select a template you want to start from

#### From template gallery

Origin	Available templates
Default templates	Bugs
My projects	ChangeRequest
CCC poc for 1st migration	Empty
Guinea Pig	Patches
SEAS Project	Requirements
Other Projects	Risks
	SupportRequests
	Tasks
	UserStories

Then, fill the name, description and short name of this new tracker.

Name: \*

BUG\_Example

Description: \*

To track Bugs and defect log

Short name: \*

e.g. bug

Please avoid spaces and punctuation in short names

Create

Fig 1

After creating a Tracker administrator can modify following things, according to the organization requirement.



Fig 2

**B. General Settings**

This field is necessary field used to define the title, the description, etc. of this tracker. It can include various fields such as:

Name of Tracker	<input type="text"/>
Description	<input type="text"/>
Instructions	<input type="text"/>

Fig 3

There could be several other fields too such as “short name” for the tracker or Instantiate for a new project if any”. Name of the Project can be treated as a mandatory filed. And it should be mandatory otherwise it becomes difficult for us to recognize the created tracker.

**C. Manage Permissions**

**1.) Manage Tracker Permission**

This field is used to set the permissions for all the users, registered users, and project members or to set the permissions for the administrator too. We can make a list or to set the permissions there could be a dropdown we can write according to our requirements. For e.g.

All Users	<input type="text"/>
Registered Users	<input type="text"/>
Project Members	<input type="text"/>
Administrator	<input type="text"/>
Others	<input type="text"/>

Fig 4

We can add other fields too, suppose we set the permission for all user such as “access rights for all the artifacts” similarly we can set permission for others also.

**2.) Manage Fields Permission**

We have set permission for the tracker, now if permissions are set for all type of users then it would be somehow easy to set the permission for all the fields. Three kinds of permissions could be there such as: Read, Write and Update

All Users	<input type="checkbox"/>	Read Only
Project Members	<input type="checkbox"/>	Update also

Fig 5

Similarly in this way we can add some more fields and set permission according to the type of user.

**D. Manage Field Usage**

This field is the beauty of this Tracker, because this field is completely handled by the administrators of this Tracker. This is the filed that helps us to modify existing fields or to add new **Fields** can be

- |               |            |              |             |                  |
|---------------|------------|--------------|-------------|------------------|
| 1. String     | 2. Date    | 3. Check box | 4. MultiBox | 5. File Upload   |
| 6. Select Box | 7. Integer | 8. Open List | 9. Float    | 10. Multi Select |

Fig 6

We can include some **Dynamic fields** too.

- |                 |                 |                    |                      |
|-----------------|-----------------|--------------------|----------------------|
| 1. Submitted On | 2. Submitted By | 3. Cross reference | 4. Last Updated date |
|-----------------|-----------------|--------------------|----------------------|

Fig 7

So, using these kind of fields or dynamic filed we can prepare a complete set of fields according to the requirement of Organization.

**Product Details**

Product/Application(for Releases/ASAPs) *	Product Release Number(for Releases/ASAPs) *
Technology None	Impacted Phases None
Forum None	TP Id

**Defect Details**

Summary \*

Defect Description

Defect ID \*

Defect Item Creator

Severity \*

Category

Corrective Action

Item Type

Closure Date

Fig 8

*These kinds of fields could be added.*

**E. Semantics**

As trackers can be fully customized, you may want to define what the title of your artifacts is, or when you consider an artifact to be open or close. This information is used in the application to display artifact summary.

**F. Manage Workflow**

To control the cycle of any artifact, management of workflow is important. Cycle of artifact will depend upon the organization requirements

We can manage the status from the workflow,

FROM	TO STATE	TO STATE	TO STATE	TO STATE	TO STATE	TO STATE	TO STATE
	New	Assigned	Work In Progress	Pending	Resolved		
(New Artifact)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
New	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assigned	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Work In Progress	<input type="checkbox"/>	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pending	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Resolved	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	<input type="checkbox"/>	<input type="checkbox"/>

Fig 9

Taking an example of Bug Tracker when tester submits a new artifact, then we can manage our transitions such as: “status of any new artifact should be only in ‘New’ and administrator can see that in ‘Assign’ Mode only.

Similarly there are various transitions ‘FROM’ to ‘TO STATE’ that can be customized according to organizations workflow as mentioned in the above figure

**G. Email Notification Settings**

Tracker uses predefined rules to keep people too aware with the life of artifact.

For example in a Bug Tracker if tester submits a new artifact then all the team members will receive a notification mail such as all would be aware about the status or about the artifact. One benefit through email-notification is that we can track artifacts directly through emails.

**H. CSV Import**

Import is required when:

*Migration from one defect tracking system to Tuleap tracker system or migration from one Tuleap to another tracker.*

Project Administrators need to import the artifacts into the customized Tuleap Trackers using the well-known CSV (Comma separated value) Formats which generally supported by all the office production suites.

**I. Export**

To export complete structure of any artifact if required this feed is important. Sometimes we need to move worked tracker to any drive or machine then Export feature helps us to move from one machine to another.

To export an Excel sheet to CSV format, simply follow the steps below:

- Select File -> Save as

- In the dialog window choose CSV as the save type

**J. Delete Artifacts**

We can delete any tracker from the environment directly.

**III. HOW TO USE CUSTOMIZED TRACKER**

After customizing a tracker according to our requirement or need of the organization, we can use it easily to track the bugs or defects. Now defect tracking will depend upon the organization to organization.

It can show various options such as:

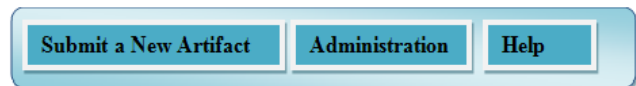


Fig 10

**A. Submit a New artifact**

**Product Details**

Product/Application(for Releases/ASAPs) \*  Product Release Number(for Releases/ASAPs) \*

Technology  Impacted Phases

Forum  TP Id

---

**Defect Details**

Summary \*

Defect Description

Defect ID \*  Defect Item Creator

Severity \*  Category

Figure 11

When tester find any kind of bug in the application he submits a new artifact which is reported to then developer. The artifact could consist of some mandatory fields that need to be filled. The above artifact figure is customized according to our organization’s requirement.

**B. Administration**

Administrator can modify the artifacts settings or fields according to need which are shown above in Fig 2.He can reset all the fields or can add/delete a field if required.

**C. Help**

During submission or creating a new tracker administrator or user can take help from this option. This would help if user stuck in between anywhere.

Moreover these various options there could be a query form, through which administrator or handler can check or track all the submitted artifacts.

#### IV. BENEFITS OF TRACKER

##### 1.) *Plan*

- Project member can plan for deliverables, assign tasks as well as it can handle the status of the project.
- Configurations of the project can be customized or we can generate graphs (Pi chart, bar chart etc.) to review the clear view of defects.

##### 2.) *Track Items*

- It can manage all types of artifacts such as bug, risks, tasks or requirements.
- It can track all the changes and can link artifacts to other projects.

##### 3.) *Code, Review, Build*

- It can manage source code with open source tools like GIT, SVN etc.
- It can integrate development environment with productivity.

##### 4.) *Generate Graphs and Reports*

- Admin team can generate various types of graphs such as Pie, Gantt, flow charts for the submitted artifacts status.
- Feature of generating graph helps to get a clear view of bug or defect.
- Report generation is also possible which will help in future to resolve the bug or to see the history of all the defects.[2]

#### V. CONCLUSION

Tuleap is a tracker which is an open source, then why we should not go for our own customized tracker when it is freely available in the market. It is user friendly and easy to use; anybody can customize it according to company requirements.

As we customized a Tuleap for Bug tracker, it helps in the organization for bugs and defect reporting. It follows the proper cycle to track the defect. As before deployment defects need to be tracked. It is a good way to track the bug in a consolidated manner. [1]

#### REFERENCES

- [1] <https://tuleap-documentation.readthedocs.org/en/latest/user-guide/tracker.html>
- [2] <http://www.enalean.com/?q=en/solutions-tuleap-open-alm>
- [3] [https://tuleap\\_net/wiki/?group\\_id=101&pagename=DeveloperGuid](https://tuleap_net/wiki/?group_id=101&pagename=DeveloperGuid)
- [4] [https://tuleap.net/plugins/tracker/?group\\_id=10](https://tuleap.net/plugins/tracker/?group_id=10)

##### **Mansi Goel**

Pursuing M.Tech (IT), From Banasthali Vidyapith & working as a Project Intern at ST Microelectronics Pvt. Ltd., Greater Noida.

##### **Priyanka Jain**

Pursuing M.Tech (CS), From Banasthali Vidyapith & working as a Project Intern at ST Microelectronics Pvt. Ltd., Greater Noida