Introduction to Issue Tracking System

劉建宏
台北科技大學資工系
10/27/2006
What is Issue Tracking System

- **What is Issue Tracking System (ITS)**
  - A software application that allows to record and follow the progress of identified problem or "issue" until the problem is resolved

- **Issue**
  - Can be anything such as customer question, review follow-up issues, change request, and report of an error or bug
  - Can be tracked by priority status, resolving state, owner, or some other customized criteria
  - Can be external or internal

- **Similar to bug tracker**
Why and When Need ITS?

- **Why need ITS**
  - Streamline project management
    - Track and audit the issues
    - Provide statistical metrics
  - Improve communication
  - Ensure accountability
  - Increase productivity
  - Enhance product quality and customer satisfaction

- **When need ITS**
  - Project/bug/issue/entry tracking and management
  - Collect project/bug/issue/entry metrics
  - Customer support
How Does It Work

- Provides the user with a way to report an issue
  - Detail the description of the issue
- Know who is responsible for resolving the issue
  - Assign the issue to whom is responsible
- Track progression towards its resolution
  - Record the any changes to the issue
  - May re-assign the issue to another user if necessary
- Provide statistical information about the issue
  - Collect and report the metrics of the issue
- Allow to customize the tracking procedure
  - Customize the work flow of the issue
Typical Scenario (from Wikipedia)

- A customer service technician receives a telephone call, email, or other communication from a customer about a problem. Some applications provide automatic error reporting from try/catch blocks.
- The technician verifies that the problem is real, and not just perceived. The technician will also ensure that enough information about the problem is obtained from the customer. This information generally includes the environment of the customer, when and how the issue occurs, and all other relevant circumstances.
- The technician creates the issue in the system, entering all relevant data, as provided by the customer.
- As work is done on that issue, the system is updated with new data by the technician. Any attempt at fixing the problem should be noted in the issue system.
- After the issue has been fully addressed, it is marked as resolved in the issue tracking system.
Who Needs to Use ITS

- Software developers
- Project managers
- Configuration managers
- Testing engineers
- Quality engineers
- Customers or end users
- IT help desks
- Others
Features of ITS

- Configurable workflow and form
- Issue management
- Project management
- Search (or filtering) mechanisms
- Statistical information about the issues
  - Graphical visualization
- Report generation
- Email notification and configuration
- Role, privilege, Account administration and Authentication
- Files attachment and Data import/export
- Integration with other tools (e.g., version control)
- Others
Attributes of Issue

- Type of the issue
  - Code error, design issue, change request
- Description of the issue
- Suggestion or attempt to resolving the issue
- Date of issue reported, assigned, resolved, and verified
- Who is involved in resolving the issues
  - Reporter, manager, developer (assigned to), verifier (tester)
- How to reproduce the issue? (optional)
- Other relevant information
Attributes of Issue

- Priority for resolving the issue
  - Critical, high, medium, low
- Severity of the issue
  - Used to describe the impact on users
  - Fatal, serious, minor
- Issue lifecycle (state)
  - Unconfirmed
  - New issue (Open)
  - Assigned (Started)
  - Reopened
  - Resolved
  - Verified
  - Closed
**Issue Life Cycle**

- Possible resolution of issue
  - Pending
  - Fixed
  - By design (Won’t fix)
  - Irreproducible
  - Duplicate
  - Later
  - Remind
  - WorksForMe
  - Withdraw
  - Duplicate
  - Disagree (Invalid)
  - Need more info

Source: netBean
# Issue State

<table>
<thead>
<tr>
<th>State</th>
<th>Set by reporter</th>
<th>Set by developer</th>
<th>QE</th>
<th>Relevant Status Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>X</td>
<td></td>
<td></td>
<td>status = <strong>NEW</strong> target_milestone = TBD (set automatically)</td>
</tr>
<tr>
<td>Evaluated</td>
<td></td>
<td>X</td>
<td></td>
<td>target_milestone != TBD</td>
</tr>
<tr>
<td>Incomplete</td>
<td></td>
<td></td>
<td>X</td>
<td>keyword = <strong>INCOMPLETE</strong></td>
</tr>
<tr>
<td>Assigned</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>assigned_to = <a href="mailto:somebody@netbeans.org">somebody@netbeans.org</a></td>
</tr>
<tr>
<td>Accepted</td>
<td></td>
<td>X</td>
<td></td>
<td>status = <strong>STARTED</strong></td>
</tr>
<tr>
<td>Rejected</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>status = <strong>RESOLVED</strong> and resolution = DUPLICATE/INVALID/WONTFIX/WORKSFORME</td>
</tr>
<tr>
<td>Fixed</td>
<td></td>
<td>X</td>
<td></td>
<td>status = <strong>RESOLVED</strong> and resolution = <strong>FIXED</strong> and target_milestone = x.y</td>
</tr>
<tr>
<td>Verified</td>
<td>X</td>
<td></td>
<td>X</td>
<td>status = <strong>VERIFIED</strong></td>
</tr>
<tr>
<td>Reopened</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>status = <strong>REOPENED</strong> and resolution = blank (set automatically)</td>
</tr>
<tr>
<td>Closed</td>
<td></td>
<td></td>
<td>X</td>
<td>status = <strong>CLOSED</strong></td>
</tr>
</tbody>
</table>

Source: netBean
Customize ITS & ITS Tools

- Customization of ITS
  - Establish the workflows for properly issue routing
    - Who should be notify
    - What actions need to be taken
  - Customize the type, scale of the priority, severity, lifecycle, resolution of the issue
  - Customize the role and privilege of users
  - Others (customize entry form, ...)

- ITS Tools
  - Commercial: clearQuest, Bugzero, Census, ...
  - Open source: mantis, bugzilla, GNATS, Scarab, Trac,...
Some Tips for using ITS

- Make **frequent** use of the "My issues" option in the Issue Tracker page. This list displays all project issues assigned to you.

- **Proceed** through your issues in order sorted by **milestone**, then by **priority**.

- Don't just go and fix issues that have not been given a milestone. It is extremely important for the project leader to carefully **consider the impact** the milestone's changes.

- Just be sure to **query the issue database first** to **avoid entering duplicate issues!**

- If you are assigned an issue and you do **not** have enough time to do the work, it is **your responsibility to communicate** (through the description field) your inability to complete the assignment.

- When an issue is completed (you've written the code, etc.), if applicable check it into CVS and mark the issue "fixed." **Include a comment on how you did it.** Fixed=Checked In!

- **Try not to batch up a bunch of issue fixes and check them in all at once.** This causes chaos on the project administrator's ability to track status and prohibits granular roll backs, etc.
Thank you for your attention 😊

Q&A
Mantis

- Mantis is a php/MySQL/web based bugtracking system
  - Allow individual or groups of developers to keep track of outstanding bugs in their product effectively

- Features and Benefits
  - Free
  - Easy installation
  - Web-based
  - Platform-independent
  - Multiple projects
  - Multiple languages
  - Emailing
  - Simple Search
  - Viewing filters
  - PHP

- http://www.mantisbt.org/