Issue Tracking

PV260 Software Quality

Matěj Karolyi, Michal Abaffy

7. 4. 2020 / 14. 4. 2020

Terms and Conditions

This PDF file is available only for students of the PV260 Software Quality course for learning purposes only.

Please do not modify or redistribute this PDF file.

Issue Tracking – Context

1. Project management
2. Software Development Life Cycle (SDLC)
3. Bigger projects (enhancements, new features, new products)
4. Smaller changes (smaller defects, bugs)

Issue Tracking – Motivation

1. What to do?
2. Who should do the next step and what the next step should be?
3. What to do with what priority?
4. How do you know what was done?
5. How do you know two people are not working on the same thing?
6. How do you know who is working on what?
7. ...

Roles related to Product

1. **Technical Support**: communication with customers; trying to solve their problems; if unable to solve, pass the problem to QA/DEV.
2. **Quality Assurance**: investigation of problems reported by TS; finding and reporting problems; checking the code changes by DEV; end-to-end automated tests.
3. **Development**: developing production code (bugs and features); unit testing; documentation writing.
4. **Product Management**: providing specification for difficult problems solution and for features; telling what features to do.
5. Possibly other roles (Scrum Master, Project Management, Release, ...)

Issue Statuses and workflows

Exercise 1

In 10 minutes, try to come up with as many as possible issue statuses. For every role from the previous slide think of at least 1 status. Think also about switching of different issue statuses among different roles - for every ordered pair "role1 to role2" think of a situation where role1 wants to send issue to role2.

Examples:

- TS to QA: Status: QA-Take a Look. Situation: Technical Support is unable to solve customers problem, wants help from QA/DEV.
- (QA/DEV/PM) to TS: Status: TS-Verify workaround. Situation: QA or DEV or PM finds a workaround for customer problem and wants TS to communicate that workaround with a customer.

Fill issue statuses and situations into Socrative.

Issue Statuses and workflows

Solution

(not complete and may differ from product to product, company to company.)

1. TS to QA: QA-Take a Look (new issue), QA-Investigate (issue where QA have already taken a look)
2. TS to PM: (Features, not defects) : PM-Take a Look
3. QA/PM/DEV to TS: TS-Could Not Reproduce; TS-Not a Bug; TS-Fact of Life; TS-Investigate; TS-Duplicate; TS-Convert to Enhancement; TS-Verify workaround...
4. QA to DEV: DEV-Fix it (well described bug); DEV - Work in Progress (fix for the bug is not good, need to change it); ...
5. DEV to QA: similar statuses as for (QA/PM/DEV to TS); QA-Verify fix (the code change is done and needs verification)

Issue Statuses and workflows

Solution (continuation)

6. TS/QA/DEV to PM: PM-Investigate (for difficult problems where the solution is not clear)
7. PM to QA/DEV: QA-Investigate; DEV-Fix it
8. TS/QA/DEV/PM to DOC: DOC-document it
9. DOC to QA: QA-Verify fix; QA-Investigate
10. QA to QA: from Take a Look to Investigate; from any to QA-Postponed; ...
11. DEV to DEV: from Fix it to Work In Progress; from any to DEV-Postponed; ...
12. TS to CLOSED: Not a Bug; Fact of Life; Duplicate; Could Not Reproduce; ...
13. QA to CLOSED: Verified fix; Not a Bug; Fact of Life; Duplicate; ...
14. Release: Closed-Verified fix to Closed-Released into production; ...

Issue Statuses and situations

Lessons learned:

1. Different roles in big companies and how those roles cooperate on serving the customers and product.
2. Introduction to issue tracking systems and how they serve easier/quicker/safer/tracked+documented/... issue resolution.
3. Got a feeling how important (and complex) a definition of issue processes can be and that the Issue Tracking Systems can easily help to model those processes.

Issue Tracking Systems

Exercise 2

In 15 minutes, give your specification for components (from user, not developer perspective) of a good issue tracking system and try to come up with as many as you can attributes of an issue record.

Fill the list of suggested components into Socrative.

Issue Tracking Systems

Solution

(Not complete and may differ from product to product, company to company. Not all issue attributes are important for all products and some product might want more attributes in issue tracking system).

Components:

1. **Issue Search** (you want to easily find issues based on several possible criteria - assigned to some person; created in such and such date interval; on QA-Take a Look status; etc... And based on combinations of such criteria. You also want to be able to define what attributes to show in the results and the ordering of the results).
2. **Issue Record** (including form with all the attributes in View/Edit mode)

Issue Tracking Systems

Solution (continuing - Components)

3. **Notifications** (you might want to get a notification - like email - when new issue is created; or when some attribute gets some value (for example when the issue gets biggest severity/priority; or you might want to able to get email On Any Change).
4. **Users/Roles + privileges + administration** (different roles should not be allowed to do the same things in Issue Tracking Systems and there is a need that someone can create/edit/delete users and set their privileges and the privileges of roles)
5. **Integration with other systems and tools** (Git; Code Review System; Continuous Integration; Customer Relationship Management and possibly others)

Issue Tracking Systems

Solution (continuing)

Attributes:

1. Issue ID
2. Abstract + Detailed Description/Repro Steps
3. Status
4. Severity
5. Priority
6. Customers reporting the problem
7. Assigned to
8. Comments
9. History (what fields were changed how and when and who changed them)
10. Automation (not possible; not wanted; to be done; done)
11. Root Cause

Issue Tracking Systems

Solution (continuing - Attributes)

12. Special Escalation
13. Component Area / Tags / Search Keywords
14. Related issues (including different relationship like Related; Follow up for; Followed up by; Duplicate of; Injected by; ...)
15. Business Use Case
16. Change Lists (integration with Version Control System)
17. Code Review + its status (integration with Code Review System)
18. Needs documentation
19. Workaround
20. Notification attributes (users can set their preference on getting notifications on issue change; when doing change, users can add other people, teams to receive notification).
21. TS/QA/DEV/PM/DOC Time Spent

Issue Tracking Systems

Solution (continuing - Attributes)

22. Date Created
23. Broken In (branch + some CL/label in the branch)
24. Fixed In (branch + its CL/label)
25. Next-Release-Stopper (must this issue be fixed before some date and/or release?)
26. Created/Scrubbed/Fixed/Reviewed/Verified By (with the value of the appropriate TS/QA/DEV)
27. Expected Date of Fix (and/or Investigation)
28. Type (Defect, Enhancement/Feature, Regression test, ...)
29. Product Team
30. ...

Issue Verification

Things to do during issue verification (but depends on the processes in your company):

1. Check that code review was done and the change is integrated to all correct branches.
2. Check that automated tests do not start to fail (Continuous Integration)
3. Write new automated tests if desired.
4. (If needed) manual testing that the fix did not break something.
5. (If needed) testing that the defect is really fixed.

Issue Verification

What to do if there is a problem discovered during verification:

Depending on several things (mainly on the processes in your company), either new issue should be filled or the existing one should be returned back to DEV.

Some problems that might be discovered:

- The fix is only partial, does not fix the whole problem
- The fix fixed the original problem, but creates a new one (the new one might be much more serious than the previous one. In that case a backout is often needed)
- Something from the processes is not done well (like missing Code Review; change not being integrated to all branches where supposed to, ...)

Links

- List and comparison of existing issue tracking systems: https://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems
- Public issues for JIRA in JIRA: https://jira.atlassian.com/projects/JRA/issues