Business Requirement (BR)

BR01: The bug, in general, might be defined as a deviation from requirements or the abnormal behavior of the software. Bugs Tracker is reporting and monitoring the progress of bugs/defects from discovery through to resolution. This process should focus on preventing the defects, catching the defects as early as possible, and minimizing their impact on the project.

BR02: BugTracker will be the tool that Rock Soft uses to monitor the status of defects. It will maintain bug history and store details from bug origin to bug resolution.

Requirement Statement (RS)

RS01: Monitor emails to see the status of defects.

RS02: Maintain the bug history and store details from bug origin to bug resolution.

RS03: The tracker should have different levels of error

RS04: Error tracker is generally associated with a manager

RS05: Bugs reporting guidelines

Stakeholder

RS01: The tracker should monitor emails to see the status of defects.

F01: Register bug

- Give a status to the bug by the developer (i.e: unconfirmed, new, assigned, resolved etc).
- Least significant will be dropped or ignored

RS02: The tracker should maintain the bug history and store details from bug origin to bug resolution.

F02: Apply bug priority standard (i.e: immediate, urgent, high, normal, low)

RS03: The tracker should be error-prone.

F03: Give a bug one of severity standard (i.e: blocker, critical, major, normal, minor etc)

RS04: Error tracker is generally associated with a manager

F04: Assign roles and responsibilities

- Several of the roles might be performed by the same individual and

- Whenever necessary - several people could fulfill one role.

F05: Policy for bugs management

- Any registered Bug Tracker user can report a bug to the bug-tracking system.
- The bugs must be filed correctly and with as much information as possible. If the bug is reported in an improper way, the testing and development teams might not be able to reproduce it and /or fix it, and the bug might be closed promptly.
- If the reported issue is further evaluated as an enhancement in the product, it will be considered as an enhancement and the reporter will be asked to file it as a change request.
- All bugs must follow the process. If a bug is not submitted in accordance with this process, it won't be considered.
- The content of the bugs repository will be visible to all BUG Tracker users.
- The bugs might be rejected, because they are duplicates or are instead considered to be a feature request.

RS05: Bugs reporting guidelines

F06: Short but clear and explanatory description of an issue

F07: Environment. The specifics of the environment in which the issue occurred are necessary to allow the problem to be reproduced (i. e:hardware configuration; operating system; dependent software etc)

F08: Steps to reproduce the problem: a concise and minimal set of steps which can be followed to reproduce the problem.

F09: Expected results: This should describe the expected behavior or expected results which will readily explain why the problem is being reported when compared to the actual results.

F10: Actual results: This should describe what actually happened, complete with any error messages, stack traces, screen shots, log files that show the outcome. It may be the case that the behavior is correct but has been misunderstood when compared to the expected results, in this case the resolution may be that clearer documentation is required

Functional Requirements List (FR)

F01: Register bug

I01: Bug ID

I02: Unfinished requirements (bugController)

I03: Requirements that are not detailed enough

I04: Requirements with unclear or multiple meanings

I05: Logic errors in the design documents

I06: Coding errors

I07: Not enough sufficient testing

I08: Misunderstanding of user needs

I09: Lack of documentation

- Give a status to the bug by the developer. Least significant will be dropped or ignored

I10: Unconfirmed: The bug is added, and it's not yet validated.

I11: New: The bug is validated, and it must be processed.

I12: Assigned: The bug is not yet resolved, but is assigned to the developer.

I13: Resolved: The bug is resolved in a way described below.

1. Fixed. The bug is checked and tested.
2. Wontfix. The developer described a bug which will never be fixed.
3. Duplicate. The bug is repeated more than once. -
4. Worksforme. All attempts at reproducing this bug were futile, and reading the code produces no clues as to why the described behavior would occur.
5. Invalid The bug is in some way not valid.

I14: Verified: The bug is duly fixed.

I15: Reopened: The bug is detected again.

I16: Closed: The bug is fixed and confirmed its absence

F02: Apply bug priority standard:

I15: Immediate: The bug blocks development or testing work and should be fixed ASAP, or is a security issue in a released version of the software.

I16: Urgent: The bug blocks usability of a large portion of the product, and should be fixed before the next planned release.

I17: High: Seriously broken, but not as high impact. Should be fixed before the next major release.

I18: Normal: Either a fairly straightforward workaround exists or the functionality is not very important and/or not frequently used.

I19: Low: The bug is not all that important.

F03: Give a bug level of severity standard:

I20: Blocker – The bug is so important that it prevents development or testing on the affected product.

I21: Critical – The bug causes the product's crash or a function does not work at all.

I22: Major – The bug that affects the product's feature to be operational.

I23: Normal – The bug impacts the product to work improperly.

I24: Minor – The bug causes the product to not work optimal, but it is possible to workaround such a bug.

I25 Trivial – The bug irritates users, something is clearly not right, but it does not affect usability.

I26 Enhancement – A proposal of new functionality that improves the product in the future.

F04: Assign roles and responsibilities

I27: Reporter - Person(s) who reports the bug.

I28: Manager - Person who acknowledges the bug and assigns it to a developer.

I29: Tester Person(s) - who tests the bugs and verifies the solution.

I30: Developer Person(s) - who resolves the bugs.

F05: Policy for bugs management

I31: guide to managing and resolving errors

F06: Short but clear and explanatory description of an issue
F07: Environment. This may include details such as:
I31: hardware configuration;
I32: operating system;
I33: dependent software;
I34: browser version;
I35: product version;
I36: the component which failed.

F08: Steps to reproduce the problem

F09: Expected results
I37: Description
I38: Resolution
I39: Required documentation

F10: Actual results:
I40: Error messages
I41: Stack traces
I42: Screen shots
I43: Log files that show the outcome

Non-Functional Requirements List (NR)

Qualities
Reliability
Q01: Creates a more efficient system

- Organized system notification of bug creation and status change to team members increases responsiveness and awareness of priority & productivity.

Q02: Prevent the defects

- If it is not possible or practical, the defect should be found as quickly as possible in order to minimize the impact of the defect.

Q03: Risk driven

- The priorities, and resources should be based on the extent to which risk can possibly be reduced.

Q04: Automatization

- The process of reporting and analyzing the bug-related information should be automated as much as possible.

Q05: Solving application’s bug

- Unless bugs are tracked, they can't be fixed, leading to failed projects.

Q06: Visibility of development process

- Improve customer satisfaction by increasing communication and allowing them to monitor the progress of development.

Q07: Traceability

- Maintain audit trails of bugs and their resolutions to ensure all changes are accounted for.

Q08: Release planning

- Manage the bugs and enhancements that are to be resolved for the next product release.

Q09: Resource scheduling

- Manage the bugs that are assigned to each team member.

Q10: Prioritization

- Assign priorities to the bugs to ensure critical errors are addressed before minor issues, such as the wording of an error message.

Q11: Control of a project

- Monitor the status and progress of bugs, to follow the improvement in stability of a product or to ensure early detection of failing projects.

Q12: Information consolidation

- Capture all bugs in one place to promote the sharing of information project wide.

Constraints

RS01: Monitor emails to see the status of defects.

C01: All users must have a valid email address to track defects.

C02: Individual System Technical Difficulties (i.e. loss of internet connection, lack of system authorization, etc.)

C03 : Improper bug logging process.

Basic rule in bug reporting is bug should be properly described & it should be self explainable. If the bug is not properly reported means missing a few required fields like Incorrect Defect Title, Missing Expected Result, Error from log file or Steps to reproduce etc. which increase the defect life cycle process.

C04: Use of different bug tracking template

If all team members are using different Issue Tracking templates to report the issues which may lead to discrepancy at a later stages. Due to this reason standard bug tracking or logging templates should be used, which could help a lot in the report generation.

RS02: Maintain the bug history and store details from bug origin to bug resolution.

C05: Confusion over descriptions, lack of information, tools that are overly cumbersome and require mandatory fields for which the user doesn't have the answers, and difficulty in reporting.

C06: Improper Defect Triage Process

All issues reported by the tester must have common understanding over all peoples. Without this meeting, which allows us to discuss all open defects & take actions on them, it is very difficult to understand the reason for blocking test execution.

RS03: The tracker should have different levels of error

C07: If every low-priority defect is tracked, the defect report will include many bugs that will never be fixed because fixing them doesn't provide a high enough priority standard.

C08: Improper set up of Severity & Priority

The defect severity & priority may vary from defect to defect; also the fixing time of defect will not be the same all the time. The fixing of the defect is totally based on the priority of the defect.

RS04: Error tracker is generally associated with a manager

C09: If code is thrown "over the wall" from development to test and the bugs are thrown back to development, when the only communication about defects is done through a tool, there are bound to be misunderstandings. This results in the classic line from developers, "it works on my machine," along with closure of a "user error".

RS05: Bugs reporting guidelines