Open-Source Project Roles

Team4
18012360 김채영 20011811 전영현
20011820 천동희 21011823 서미경
21011826 백세진 21011830 장서현
21011840 이지은 22011867 염예은

Scenario 4 - Bug Backlog Crisis

The number of bugs reported by users has significantly increased, and your project is struggling to keep up. Some users are starting to lose confidence in the project's reliability. Your team needs to figure out a way to handle the bug backlog efficiently while keeping other important tasks moving forward.

Challenges:

- Managing a growing backlog of bugs.
- Allocating resources between fixing bugs and feature development.

<New Contributors>

1. Support fixing bugs: By solving easy bugs, new contributors adapt to the way team members work and gradually develop their skills. This takes the burden off other team members so that they can focus on more important bugs.

2. Functional development: New contributors can participate in the development of new functions to a certain extent. New contributors identify the roadmap of the project and take on important but non-urgent functional development so that they can find a balance between bugs and functional development.

3. Communication and cooperation: New contributors should actively communicate with experienced contributors and accept feedback. Through code review, new contributors can improve their code and develop understanding of the overall project.

4. Documentation and Support: While Experienced Contributors are solving bugs, new contributors can improve the text environment and review code. Also, for history tracking, new contributors can organize the solution process and document the guide so that other team members can see it at a glance.

5. Learn test code and improve quality: New contributors learn test code which is used in the project, and increase test coverage to prevent similar problems from recurring.

<Experienced Contributors>

1. Collaborates with the Project Leader to assign roles to contributors. Handles complex issues personally and delegates simpler tasks to New Contributors. Monitors the overall project status and provides continuous feedback to improve work efficiency.

2. Assists New Contributors in reproducing bugs locally. Reduces their burden by providing necessary tools, environment setup guides, and onboarding documentation to enable them to work independently.

3. Quickly diagnoses the core issues of bugs based on past experiences and guides New Contributors on how to fix them. Also reviews the code written by New Contributors and provides feedback, suggesting better solutions for improvement.

4. Manages bug fix code in the review queue and inspects the performance, consistency, and other aspects of the code that has passed automated testing.

5. Guides the creation of test cases to prevent similar bugs from reoccurring. Additionally, provides guidance on writing commit messages and documenting bug fixes to contribute to better project management.

<Committers>

Step:

1) Reproduce the bug locally based on collected detailed bug reports, by mimicking users' environment and actions. The Project Core Committer should prioritize addressing critical bugs affecting users the most to maintain software stability, ensuring such bugs are reproduced efficiently.

2) Add logging function and debug to capture more information about the bug when it occurs. The Core Committer should review and merge bug fix submissions quickly to manage a growing backlog while ensuring logs and debugging efforts are focused on solving high-priority issues.

3) Analyze the bug based on collected logs or debugging information, looking at patterns like memory or data size issues. Collaborating with the Project Lead or Manager, the Core Committer should help allocate team resources effectively, ensuring focus on critical bugs.

4) Fix the bug, which may involve correcting logic errors, adjusting memory allocation, or adding error handling mechanisms. The Core Committer can also encourage external contributors to assist with the bug backlog if it's an open-source project.

5) After fixing the bug, create a unit test simulating the same scenario to prevent the issue from reappearing. The Core Committer must ensure all fixes are thoroughly tested before merging, maintaining code quality and stability.

6) Commit the changes to the repository with an explanation of the bug and the fix. The Core Committer should enforce testing guidelines and encourage small, manageable pull requests to speed up the bug-fixing process.

<Project Leaders>

1. Resource Management

It maintains a balance between bug fixes and new feature development by allocating tasks according to the skills and experience of team members. If necessary, acquire external resources or additional personnel to reduce workload and support the smooth progress of the project.

2. Communication

Project leaders should facilitate daily stand-up meetings or weekly sessions to share the current bug status and promote collaboration within the team. It maintains transparency in the project by regularly communicating with team members about the project's progress, bug fixes, and feature development. Also, ensure that all team members clearly understand the project's guidelines and procedures through thorough documentation.

3. Process Improvement

Project leaders should utilize code reviews and test automation tools to ensure code quality and develop contribution guidelines that help team members work consistently according to defined standards. These guidelines should include the severity level, environment information, and steps to reproduce the bug. And they should analyze the root cause of recurring issues and establish long-term solutions to improve the stability and efficiency of the project.

4. Crisis Management

Project leaders should respond swiftly to unforeseen problems or bugs, and allocate necessary resources to resolve the issue. If critical bugs or functional defects are discovered, establish a strategy to address them quickly. Additionally, identify and manage potential risks in advance to prevent the project from being disrupted or significantly impacted.

5. Bug Prioritization

Project leaders should establish clear criteria for prioritizing bugs as a leader to prevent confusion about which bugs to address first. The criteria for categorizing and prioritizing bugs should include factors such as the severity of the bug, its impact on users, and the difficulty of resolution.

6. Quality Assurance Process Enhancement (QA Process Supervision)

If bugs continue to accumulate, it may indicate issues with code quality or the QA process. Implement a contribution guideline that mandates testing after bug fixes. And,the project leaders should establish and oversee procedures to ensure that key functionalities are operating correctly, thus coordinating efforts to prevent the recurrence of such issues.