**1.Khi nào chúng ta dừng thử nghiệm?**

**Test Coverage**:

- All planned test cases have been executed, including functional, non-functional, regression, and acceptance tests.

- High-priority test cases are passed, and any critical issues have been addressed.

**Defect Rate**:

- The rate of finding new defects has decreased to an acceptable level.

- No high-priority or critical defects remain unresolved.

**Deadlines and Budget**:

- Testing has reached the scheduled deadline, and further testing may not be feasible within the project timeline or budget.

**Quality Goals**:

- The software meets the predefined quality standards and requirements.

- Stakeholders are satisfied with the current quality level.

**Risk Assessment**:

- Risks have been mitigated to an acceptable level.

- Residual risks are documented and agreed upon by stakeholders.

**Performance Benchmarks**:

- The system meets performance, scalability, and reliability benchmarks.

- Non-functional requirements are satisfied.

**User Acceptance Testing (UAT)**:

- End-users or clients have performed UAT and approved the system for release.

**Regulatory Compliance**:

- The software complies with all relevant legal and regulatory standards.

**Test Environment Stability**:

-The testing environment is stable, and results are consistent and reliable.

**2. What information do you mention in a bug?**

* Summary .
* Component
* Description:
* Precondition.
* Step to reproduce.
* Actual result.
* Expected result.
* Environment
* Fix version
* Priority
* Labels
* Attachment
* Linked issues
* Issue
* Assignee
* Epic Link
* Sprint

**3. What differ ‘Priority’ from ‘severity’ Tùng anh**

of bugs? Who set these fields?

**Priority** vs. **Severity**:

* **Severity**: Indicates the impact of the bug on the system (e.g., Critical, Major, Minor). Set by the tester.
* **Priority**: Indicates the urgency to fix the bug (e.g., High, Medium, Low). Set by the project manager or product owner.

For example, a critical bug causing a system crash has high severity but might be low priority if it affects a rarely used feature.

**4. State of bug and bug life cycle? Tùng Anh**

### **Bug Life Cycle**

1. **New**: Bug is reported.
2. **Assigned**: Bug is assigned to a developer.
3. **Open**: Developer starts working on the bug.
4. **Fixed**: Bug is resolved by the developer.
5. **Test**: Bug fix is verified by QA.
6. **Verified**: Bug is confirmed as fixed.
7. **Closed**: Bug is closed after successful verification.
8. **Reopened**: Bug is reopened if not fixed properly.

Additional states can include **Deferred**, **Rejected**, and **Duplicate**.

**5. What tool do you store bugs?(Hường)**

I often use Google Sheet and Jira tool to store bugs:

* Jira tool: Jira allows users to discuss and add comments directly within each bug ticket to update information, discuss specific issues, and track the history of changes. It also enables tracking and updating the status of bugs from when they are reported, through processing, to resolution or closure.
* Google Sheet : In addition, I use Google Sheets to support bug tracking by sharing spreadsheets with team members to add bug details. Later, this data is inputted into Jira for more detailed management.

**6. What do you do if the issue is rejected? (Focus on the quality, not fight for self. issue not bug cause it is not confirmed as bug from both sides) (Hường)**

Firstly, I will inquire about the specific reason why the issue was rejected and base my actions accordingly. If the issue does not significantly impact the project but has the potential to enhance its quality, I will discuss with the team to consider improvements. However, if the rejection is due to insufficient information, I will provide clear explanations to clarify the issue. Because every problem raised aims to contribute positively to project improvement and represents an opportunity for learning and deeper understanding of the project.