Defect Management

- **What is priority?**

  Priority refers to how soon a bug should be fixed. It is set by the Project Manager or Client based on business needs.

- **High priority:** Needs immediate fix.
- **Medium priority:** Fix after high-priority issues.
- **Low priority:** Can be fixed later or in future releases.

- **What is severity?**

  Severity indicates the impact of a bug on the system's functionality. It is usually set by the tester or developer**.**

- **Critical severity:** System crash or data loss.
- **High severity:** Major functionality broken.
- **Medium severity:** Minor functionality affected.
- **Low severity:** Cosmetic issue or type.

- **Bug categories are…**

Functional Bugs: Issues in features or logic.

Performance Bugs: System is slow or lagging.

UI/UX Bugs: Visual defects or misalignments.

Security Bugs: Vulnerabilities like XSS, SQL injection.

Compatibility Bugs: Doesn’t work on some browsers/devices.

Integration Bugs: Data or process mismatch between modules.

Regression Bugs: Old bugs reappear after changes.

- **Advantage of Bugzilla**.

  - ✓ Open-source and free.
  - ✓ Customizable bug-tracking workflows.
  - ✓ Email notifications and permissions.
  - ✓ Supports attachments, comments, and advanced queries.
  - ✓ Useful reports, charts, and timelines.

- **Difference between Priority and Severity:**

Aspect	Severity	Priority
Meaning	Impact of the bug on the system functionality	Urgency of fixing the bug
Decided by	QA/Testers	Product Managers/Developers
Focuses on	Technical impact	Business urgency
Type	Objective (based on how bad the bug is)	Subjective (based on business needs)
Example	App crashes = High severity	Homepage typo before launch = High priority
Fix Timing	Might be high severity but fixed later	High priority bugs are fixed first

- **What are the different Methodologies in Agile Development Model?**

Agile is a flexible and iterative approach to software development. Here are the most common methodologies under Agile:

1. **Scrum**
   - Most widely used.
   - Work is done in fixed-length iterations called Sprints (usually 2–4 weeks).
   - Roles: Product Owner, Scrum Master, and Development Team.
   - Includes daily Stand-up meetings, Sprint Planning, Review, and Retrospective.
2. **Kanban**

- Visual tool using a Kanban board with columns (To Do, In Progress, Done).
- No fixed time frames like Scrum.
- Focuses on **continuous delivery** and limiting work-in-progress (WIP).

3. **Extreme Programming (XP)**
   - Focus on technical excellence and developer practices.
   - Uses pair programming, test-driven development (TDD), continuous integration, and frequent releases.
4. **Lean**
   - Derived from Lean manufacturing.
   - Aims to reduce waste, improve efficiency, and deliver faster.
   - Emphasizes value delivery, minimum viable product (MVP), and continuous improvement.
5. **Crystal**
   - Lightweight and adaptive.
   - Tailored to team size and criticality of the project.
   - Encourages collaboration and frequent delivery.
6. **DSDM (Dynamic Systems Development Method)**
   - Time-boxed and focused on business needs.
   - All changes must be reversible.
   - Ensures active user involvement throughout the project.

- **Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?**

Feature	Authentication	Authorization
Purpose	Confirms who the user is	Confirms what the user can access
Happens When?	First – during login	After authentication
Example	Entering username and password	Accessing admin-only features
Tools	Login forms, OAuth, biometrics	Role-based access control, permissions

- **Common Problems Faced in Web Testing**

- ✓ **Cross-browser compatibility** – Site works in Chrome but not in Firefox/Edge.
- ✓ **Responsive design issues** – Poor layout on mobiles or tablets.
- ✓ **Broken links or images** – 404 errors or missing content.
- ✓ **Slow load times** – Affects user experience and SEO.
- ✓ **Security issues** – Vulnerable to attacks like SQL injection, XSS, CSRF.
- ✓ **Form validation problems** – Weak input checks can allow invalid or harmful data.
- ✓ **Session management bugs** – Session timeout issues or improper logout behavior.
- ✓ **UI/UX inconsistencies** – Misaligned buttons, overlapping elements.
- ✓ **JavaScript errors** – Scripts not executing properly or causing crashes.
- ✓ **Browser caching issues** – Old versions of files being used.