Defect Lifecycle Stages

Stage	Description
Identification	Tester or user spots a defect during testing or usage
Logging	Defect is recorded in a tracking system with details
Triage	Team reviews and prioritizes the defect based on severity and impact
Assignment	Developers or team members take ownership of fixing the issue
Resolution	Fix is implemented and verified through retesting
Closure	Once validated, defect is marked as closed

Methods for Identifying Defects

- **Manual Testing**: Exploratory, scenario-based, or checklist-driven reviews
- **Automated Testing**: Scripts flag anomalies during unit or integration tests
- **Code Reviews**: Peer inspection finds logical or syntax flaws
- **Static Analysis Tools**: Scan for vulnerabilities, unused code, or poor practices
- **User Feedback**: Real-world use often reveals overlooked issues

Importance of Logging Defects

- Enables tracking and accountability
- Facilitates communication across teams
- Prioritizes workload based on severity
- Helps refine test cases for future releases
- Forms a historical record for long-term learning