S/W Quality

Unit V

- Quality concepts
- Quality Attributes
- Deployment Overview and Planning
- Deployment Rollback

Software Quality Concepts

- Software quality refers to how well software meets functional and non-functional requirements.

Key Concepts:

- **Purpose:** Software should fulfill the intended use as defined in the Software Requirements Specification (SRS).
- **Conformance to Requirements:** Quality is measured by how closely the software adheres to documented requirements.
- **Defect Management:** Focuses on identifying, categorizing, and minimizing defects during development.
- **Quality Assurance vs. Quality Control:**
  - *QA* ensures processes are in place to prevent defects.
  - *QC* involves testing and inspection to detect defects

Quality Attributes

- measurable characteristics that determine software’s performance and user satisfaction.

Common Attributes:

- **Reliability:** Ability to perform under specified conditions for a defined period.
- **Usability:** Ease of use and user interface friendliness.
- **Efficiency:** Optimal use of system resources.
- **Maintainability:** Ease of making changes or fixing bugs.
- **Portability:** Ability to run across different environments.
- **Security:** Protection against unauthorized access and data breaches.

Quality Attributes: An example

Attribute	Description	Real-World Example
Reliability	Consistently performs its intended function without failure.	**WhatsApp** rarely crashes and reliably delivers messages even with poor signal.
Usability	Easy to learn and use; intuitive interface.	**Google Maps** offers clear navigation, voice guidance, and simple search tools.
Efficiency	Uses minimal resources while delivering optimal performance.	**Subway Surfers** runs smoothly even on low-end devices with limited memory.
Maintainability	Easy to update, fix bugs, or add features.	**WordPress** plugins and themes can be updated independently without breaking the site.
Portability	Can run across different platforms or environments.	**Spotify** works seamlessly on Android, iOS, Windows, macOS, and web browsers.
Security	Protects data and prevents unauthorized access.	**Paytm** uses two-factor authentication and encryption to secure financial data.

Deployment Overview and Planning

- Deployment is the process of delivering software to users or production environments.

Phases of Deployment:

- **Preparation:** Final testing, documentation, and packaging.
- **Environment Setup:** Configuring servers, databases, and networks.
- **Release Planning:** Scheduling, versioning, and stakeholder communication.
- **Monitoring:** Post-deployment checks and performance tracking.

Best Practices:

- Use **CI/CD pipelines** for automated deployment.
- Include **rollback strategies** in the deployment plan.
- Communicate clearly with users about changes and downtime.

Continuous Integration and Continuous Delivery/Deployment

- it's a modern software development practice that automates and streamlines the process of building, testing, and releasing code.

CI/CD Pipeline Stages

- **Source Control**: Code is pushed to a version control system (e.g., Git).
- **Build**: Code is compiled and dependencies are resolved.
- **Test**: Automated unit, integration, and UI tests are run.
- **Release**: Code is packaged and tagged for deployment.
- **Deploy**: Code is deployed to staging or production environments.
- **Monitor**: Logs and metrics are tracked to ensure stability.

Advantages of CI/CD

- Faster release cycles
- Reduced bugs and regressions
- Improved collaboration
- Easier rollback and recovery
- Higher confidence in code quality

CI/CD

Concept	Description	Real-World Analogy
CI (Continuous Integration)	Developers frequently merge code changes into a shared repository. Automated builds and tests run to catch issues early.	Like a team writing a book together and constantly checking for grammar errors as they add new pages.
CD (Continuous Delivery)	Code is automatically prepared for release to production. It’s tested, packaged, and ready to deploy at any time.	Like having a finished book proofread and printed, ready to ship whenever needed.
CD (Continuous Deployment)	Every change that passes automated tests is deployed directly to production without manual approval.	Like automatically publishing each new chapter online as soon as it’s written and reviewed.

Deployment Rollback

- Rollback is reverting to a previous stable version when a deployment fails or causes issues.

Rollback Strategies:

- **Version Control:** Maintain previous builds for quick reversion.
- **Blue-Green Deployment:** Run two environments (live and standby) to switch instantly.
- **Feature Flags:** Disable problematic features without full rollback.
- **Database Snapshots:** Restore data to pre-deployment state if needed.

When to Rollback:

- Critical bugs or crashes
- Security vulnerabilities
- Performance degradation