Leapmile - Software Development Life Cycle (SDLC)

1. Development:
Software development begins in a controlled environment where developers write and unit-test code based on requirements. Coding standards and best practices are strictly followed to ensure quality from the start. Version control systems are used to manage code changes and enable collaboration.

2. Testing (QA/QC):
After development, code moves to the testing phase, where comprehensive Quality Assurance (QA) and Quality Control (QC) activities take place. This includes functional testing, integration testing, performance testing, and security assessments. Automated and manual tests verify that the software meets all specifications and is free from critical defects.

3. Staging:
Once QA/QC is completed successfully, the code is deployed to a staging environment that closely mirrors production. This environment is used for User Acceptance Testing (UAT) and final validation. Peer reviews and detailed code walkthroughs are conducted by the development manager before approval. Any issues discovered here are fed back to development for resolution.

4. Release to Production:
After successful UAT and management approval, the software is released to the production environment. Deployment follows documented release procedures to minimize downtime and ensure rollback options if needed.

5. Updates and Patches:
Ongoing maintenance includes applying updates, bug fixes, and patches through controlled release cycles. Each change goes through the same rigorous testing and approval process to maintain system stability.

Change Control Processes

Throughout the SDLC, change control is strictly enforced to manage scope and maintain quality:

- **Change Requests:** All modifications are initiated via change requests detailing the reason, impact, and proposed solution.
- **Impact Analysis:** Changes undergo impact and risk assessments involving developers, QA, and project managers.
- **Approval Workflow:** Changes require multi-level approval before implementation, ensuring alignment with project goals and stakeholder expectations.
- **Version Control:** All changes are tracked in version control systems, maintaining clear audit trails.
- **Documentation:** Updates to code, configurations, and test cases are documented to reflect changes accurately.

- **Regression Testing:** Before deployment, regression tests confirm that new changes do not adversely affect existing functionality.

This structured approach ensures that software development is disciplined, transparent, and focused on delivering reliable, high-quality solutions.