Top of Form

1. **Establishment:   
   • Description:** In this step, a repository is created, access permissions are configured, and the SCM system is set up.  
   • As an illustration, use the git init command in a project directory to start a Git repository.   
     
   2**. Branching:**

• Synopsis: Using a branching mechanism, developers can focus on specific features or bug fixes without influencing the main codebase. Every branch denotes a distinct developmental path.   
• As an illustration, use git checkout -b feature-authentication to create a new branch with the name "feature-authentication".

1. **Development**

• Described as follows: Within their own branches, developers write, edit, and test code. Adding new features, resolving problems, and guaranteeing code quality are all part of this phase.

• As an illustration, consider writing code in the "feature-authentication" branch to add authentication features to a web application.

1. **Review:**

• Summary: Peer code review guarantees code quality, conformity to coding guidelines, and information exchange within the team.

• As an illustration, submit a pull request to GitHub for the "feature-authentication" branch so that other members of the team can see the updated code and offer comments.

1. **Integration:**

• Description: Code modifications are merged back into the primary codebase to include updated features or bug fixes after they have been examined and approved.

• As an illustration, use git merge feature-authentication to merge the "feature-authentication" branch into the main branch.

1. **Testing**:
   1. **Description**: Automated and manual testing is performed to validate the functionality, performance, and reliability of the integrated code.
   2. **Example**: Running unit tests, integration tests, and user acceptance tests to verify the authentication feature's behaviours.
2. **Deployment**:
   1. **Description**: The tested and validated code is deployed to production or staging environments for end-users to access.
   2. **Example**: Deploying the latest version of the application with the new authentication feature to a staging server for final testing before production release.
3. **Monitoring and Maintenance**:
   1. **Description**: Continuous monitoring of the deployed application ensures its availability, performance, and security. Maintenance involves fixing bugs, applying patches, and updating dependencies.
   2. **Example**: Using monitoring tools like New Relic or Prometheus to track application metrics and responding to alerts promptly to maintain system health.