SDLC

Presented By : SecureTest Partners

STP Training Academy | 2024

WHAT IS SDLC

SDLC stands for Software Development Life Cycle. It is a systematic process or set of phases that guide the development of software applications or systems. SDLC aims to produce high-quality software that meets or exceeds customer expectations, is delivered on time and within budget, and is easy to maintain and adapt.

WHY IT IS IMPORTANT

- Structured Approach
- Cost and Time Management
- Efficient Collaboration
- Continuous Improvement
- Quality Assurance
- Clear Communication
- Customer Satisfaction
- Risk Management
- Adaptability and Flexibility
- Documentation

KEY STAGES OF SDLC

- Planning
- Implementation
- Analysis
- Testing
- Maintenance
- Design
- Deployment

SDLC

Phase 1 - Planning

Define Project Scope
Set Project Goals and Objectives
Identify Stakeholders
Resource Allocation
Risk Assessment

Phase 2 - Analysis

Understand Business Needs
Define System Requirements
Identify Constraints
Gathering Requirements

Phase 3 - Design

High-Level Design (Architectural Design)
Low-Level Design (Detailed Design)
Design Documents

Phase 4 - Implementation

Coding and Development
Collaboration
Code Documentation

SDLC

Phase 5 - Testing

User Acceptance Testing (UAT)

Test Planning and Execution

Defect Reports

Phase 6 - Deployment

Strategies for Deployment

Rollback Plans

Real-time Monitoring

User Support

Feedback Collection

Phase 7 - Maintenance

- Importance of ongoing maintenance
- Bug fixing and updates
- Enhancements and optimizations

BEST PRACTICES IN SDLC

1. Foster a collaborative culture among team members.Use communication tools to enhance real-time collaboration.Regular meetings and updates to ensure everyone is on the same page.Encourage open communication to address challenges.

2. Implement a robust version control system (e.g., Git).Ensure consistent branching and merging strategies.Regularly commit code changes to the repository.Use descriptive commit messages for clear documentation.

3. Set up automated CI/CD pipelines to streamline the development process.Regularly integrate code changes to identify and address issues early.Automate testing to ensure code quality and reliability.Enable continuous delivery for efficient and reliable software releases.

TOOLS FOR SDLC

Version Control Tools

- Git
- SVN (Subversion)
- Mercurial

Testing and Deployment Tools

- Jenkins
- Selenium
- Redmine
- Testrail

Project Management Tools

- Jira
- Trello
- Asana

SDLC APPROACHES

- **Traditional Approaches**
  - Waterfall Model
  - V-Model
  - Spiral Model

- **Agile Approaches**
  - Scrum
  - Kanban
  - Extreme Programming (XP)
  - Lean Software Development

STP Training Academy | 2024

THANK YOU