

Software Maintenance Plan for Smart Checkout Nexus

Alex M. Frear

College of Science, Engineering, and Technology, Grand Canyon University

Course Number: CST-326 Written and Verbal Communication for Software Development

Professor Stephanie Macuga

04/28/2025

Table of Contents

<i>Post-Delivery Observations from Case Studies.....</i>	<i>3</i>
<i>Maintenance Plan.....</i>	<i>4</i>
Software Versioning and Delivery.....	4
IT Infrastructure Setup.....	4
Frequency of Updates.....	4
Prioritization of Updates.....	5
Bug Reporting and Enhancement Suggestions.....	5
Version Control and Configuration Management	5
<i>Training Plan</i>	<i>6</i>
Training Client IT Administrators.....	6
Training End Users	6
Training Cloud Administrators (if applicable)	6
Training for Bug Reporting	6
<i>Deployment Diagram.....</i>	<i>7</i>
Figure 1. UML Deployment Diagram for Software Application	7

Post-Delivery Observations from Case Studies

Many projects, such as Harvard Pilgrim Health Care (HPHC) and Valpak, faced significant post-delivery challenges. Common complaints involved coordination issues during updates, unexpected downtimes, and stakeholder miscommunications.

However, projects that delivered early wins and maintained continuous communication—like Valpak’s quarterly planning and cross-team dependency management—received praise for exceeding expectations.

Lesson Learned: Structured release management and a proactive maintenance plan are vital to sustained success.

Maintenance Plan

Software Versioning and Delivery

- **Versioning Strategy:** Semantic Versioning (MAJOR.MINOR.PATCH), e.g., 1.2.3
 - Major: Breaking changes
 - Minor: New features
 - Patch: Bug fixes
- **Delivery Mechanism:**
 - Updates distributed via centralized servers or cloud platforms (AWS, Azure).
 - Auto-updates for web applications; email notifications for client-side software.
- **Deployment Strategy:**
 - Blue-Green Deployment for minimal downtime.
 - Immediate hotfix patches applied as needed.

IT Infrastructure Setup

- **Network Requirements:** Secure VPN or HTTPS connections.
- **Required Software:**
 - Database: PostgreSQL 15+
 - Web Servers: NGINX or Apache
 - Browsers: Chrome, Firefox (latest versions), Edge, Safari
- **Recommended Deployment Architecture:**
 - [Client Device] → [Load Balancer] → [Web Server] → [Application Server] → [Database Server]

Frequency of Updates

Type	Frequency	Priority
Bug Fixes	Every 2 weeks	High
Feature Enhancements	Every 2–3 months	Medium
Major Releases	Every 6–12 months	Low

Prioritization of Updates

Priority	Description	Example
High	Security patches, data corruption fixes	Immediate hotfix
Medium	New features, moderate bugs	Next scheduled release
Low	Cosmetic changes, UI improvements	Bundled into later updates

Bug Reporting and Enhancement Suggestions

- **Reporting Methods:**
 - In-app feedback tool with file upload.
 - Email support and integrated ticketing system (e.g., Jira Service Desk).
- **Review Process:**
 - Triage tickets within 24–48 hours.
 - Critical bugs prioritized immediately.
 - Feature suggestions reviewed quarterly.

Version Control and Configuration Management

- **Git Branching Model:**
 - main: Production-ready
 - develop: Active development
 - feature/: New features
 - hotfix/: Urgent fixes
 - release/: Release preparation
- **Tags:** Each release tagged (e.g., v1.2.0).
- **Forks:** Used for client-specific customizations when necessary.

Training Plan

Training Client IT Administrators

- **Purpose:** Installation, update management, and backup procedures.
- **Format:**
 - 2-hour live virtual or onsite training.
 - Recorded tutorials and printable quick-start guides.

Training End Users

- **Purpose:** Basic software usage, navigation, and feature updates.
- **Format:**
 - 1-hour webinars.
 - 5–10 minute microlearning video modules.
 - FAQs and quick start PDF guides.

Training Cloud Administrators (if applicable)

- **Purpose:** Cloud environment monitoring and scaling procedures.
- **Format:**
 - 90-minute virtual workshops.
 - Documentation provided in shared cloud storage.

Training for Bug Reporting

- **Purpose:** Accurate bug reporting and escalation protocols.
- **Format:**
 - 30-minute instructional videos.
 - Easy-to-follow step-by-step reporting guides.

Deployment Diagram

Figure 1. UML Deployment Diagram for Software Application

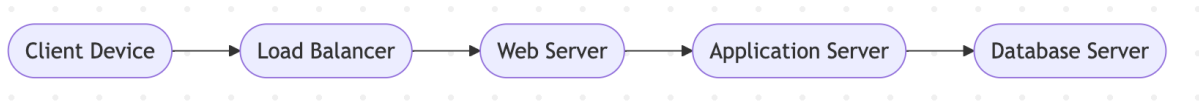


Figure 1 UML Deployment Diagram for Software Application.