# **Software Requirements Specification (SRS)**

**Project:** Software Change Management System (SCMS)

**Version:** 1.0

**Authors:**

Manchikanti Sharvani(PES1UG23AM165)

N Riddhi Samitha(PES1UG23AM175)

K S Niharika(PES1UG23AM132)

**Date:** 03-09-2025

**Status:** Draft / To be Approved

## **Revision History**

| **Version** | **Date** | **Author** | **Change Summary** | **Approval** |
| --- | --- | --- | --- | --- |
| 1.0 | 02-09-2025 | 1.Manchikanti Sharvani  2.N Riddhi Samitha  3.K S Niharika | Initial SRS with complete requirements and diagrams | Pending |

## **Approvals**

| **Role** | **Name** | **Signature/Email** | **Date** |
| --- | --- | --- | --- |
| Course Coordinator | Prof. Anand M S | anandms@pes.edu | 03-09-2025 |
| Project Guide |  |  |  |

## **Table of Contents**

1. Introduction
2. Overall description
3. External interfaces
4. System features (detailed)
5. Non-functional requirements (detailed)
6. Quality attributes & Acceptance tests
7. UML Use-Case Diagram
8. Requirements Traceability Matrix (RTM)

## **1. Introduction**

### **1.1 Purpose**

This document specifies the requirements for a Software Change Management System (SCMS). The SCMS manages the complete lifecycle of software changes from request submission through implementation, testing, and deployment, ensuring controlled and traceable software evolution.

### **1.2 Scope**

The SCMS covers change request lifecycle management, automated impact analysis, integration with development tools (Git, Jira, CI/CD pipelines), approval workflows, deployment coordination, rollback capabilities, and comprehensive audit trails. Excludes actual code development and testing execution.

### **1.3 Audience**

Software Developers, DevOps Engineers, Release Managers, QA Engineers, Project Managers, Compliance Officers, and System Administrators.

### **1.4 Definitions**

* **SCM**: Software Change Management
* **CR**: Change Request
* **CAB**: Change Advisory Board
* **CI/CD**: Continuous Integration/Continuous Deployment
* **RFC**: Request for Change
* **RBAC**: Role-Based Access Control
* **API**: Application Programming Interface
* **VCS**: Version Control System

## **2. Overall Description**

### **2.1 Product Perspective**

The SCMS is a centralized web-based platform that orchestrates software change management across development, staging, and production environments. It integrates with existing development toolchains including Git repositories, CI/CD pipelines, issue tracking systems, and deployment tools.

### **2.2 Major Product Functions**

* Submit and categorize change requests with technical impact assessment
* Automated dependency analysis and conflict detection
* Multi-stage approval workflows with CAB integration
* Integration with Git repositories for commit tracking
* CI/CD pipeline integration for automated testing and deployment
* Real-time change status monitoring, implementation tracking, and notifications
* Rollback planning, rollback execution with defined criteria, and emergency change procedures
* Comprehensive audit trails and compliance reporting
* Risk assessment and change impact prediction
* Release coordination and deployment scheduling
* Validation of test coverage through CI results mapped to change requests
* Coordination of CAB meetings (agenda, attendees, decisions linked to change requests)
* Auditor review of immutable logs filtered by CR, user, or date
* Configuration of CI/CD integration settings per project (endpoints, credentials, jobs, environments)
* Setup and management of Git webhooks for commit/merge event tracking
* Management of infrastructure configuration metadata (environments, change freeze windows)
* Customizable notification configuration (channels, recipients, rules) for deployments and rollbacks
* Validation of deployment health using automated probes and pipeline checks
* Generation of deployment reports (success/failure, duration, approvers, artifacts, environment)
* Triggering of pipeline builds and deployments directly from a change request or release page
* Synchronization of change request status with external issue tracking systems (e.g., Jira, Azure Boards)
* Real-time dashboards for monitoring deployments and system health

### **2.3 User Roles and Characteristics**

* **Developer**: Creates change requests, links code commits, and focuses on technical implementation.
* **Technical Lead**: Reviews technical feasibility, resolves conflicts, and approves low-risk changes.
* **Change Manager**: Oversees the change process, coordinates CAB meetings, and manages approval workflows.
* **Release Manager**: Plans and coordinates deployments, manages release schedules, and oversees release windows.
* **QA Engineer**: Validates test coverage, ensures quality assurance through CI/CD results, and approves testing strategies.
* **DevOps Engineer**: Handles deployment automation, sets up CI/CD integrations and Git webhooks, manages infrastructure configurations.
* **Auditor**: Reviews compliance, accesses immutable audit logs, and generates compliance-related reports.
* **System Administrator**: Manages environment configurations, monitors deployment health, enforces change freeze windows, and oversees rollback execution.

### **2.4 Operating Environment**

* Web-based application supporting Chrome 90+, Firefox 88+, Safari 14+
* Server: Linux/Windows Server with Docker containerization
* Database: PostgreSQL 13+ or MySQL 8+
* Integration APIs: REST/GraphQL with OAuth 2.0
* Cloud deployment ready (AWS, Azure, GCP)

### **2.5 Constraints**

* Must integrate with existing Git infrastructure
* Compliance with ITIL v4 change management processes
* ISO/IEC 27001 security requirements
* SOX compliance for financial sector deployments
* Limited to Git-based version control systems
* Requires network connectivity for real-time integrations

## **3. External Interface Requirements**

### **3.1 User Interfaces**

* **Primary Dashboard**: Real-time change status overview with filtering and search
* **Change Request Form**: Multi-step wizard with technical details, impact assessment
* **Approval Interface**: CAB dashboard with voting, comments, and risk visualization
* **Integration Status**: Real-time view of Git/CI/CD pipeline status
* **Reporting Interface**: Customizable reports with export capabilities
* **Mobile Interface**: Responsive design for approval notifications and status checks

### **3.2 Hardware Interfaces**

* Standard web browsers on desktop/laptop/mobile devices
* No specialized hardware requirements
* Network infrastructure for API integrations

### **3.3 Software Interfaces**

* **Git Integration**: GitHub/GitLab/Bitbucket APIs for commit tracking
* **CI/CD Integration**: Jenkins, GitLab CI, Azure DevOps pipeline APIs
* **Issue Tracking**: Jira, Azure Boards API integration
* **Notification Services**: SMTP, Slack, Microsoft Teams APIs
* **Identity Management**: LDAP, Active Directory, SAML 2.0

### **3.4 Communications**

* HTTPS with TLS 1.3 minimum for all external communications
* OAuth 2.0 for API authentication
* Webhook support for real-time integrations
* Message queuing for asynchronous processing (RabbitMQ/Apache Kafka)

## **4. System Features**

### **4.1 Change Request Submission**

| **Req ID** | **Requirement** | **Type** | **Priority** | **Source/Stakeholder** | **Acceptance Criteria** | **Comments/Dependencies** |
| --- | --- | --- | --- | --- | --- | --- |
| SCMS-F-001 | The system shall allow users to submit change requests with technical details, impact scope, and file attachments | Functional | High | Development Team | AC-SCMS-F-001: CR form accepts all required fields, validates technical impact assessment, allows multiple file uploads. Test: TC-CR-01 | Requires file storage and validation |
| SCMS-F-002 | The system shall automatically generate unique CR identifiers following format CR-YYYY-NNNN | Functional | High | Compliance | AC-SCMS-F-002: Each CR receives unique ID, no duplicates possible, format validated. Test: TC-CR-02 | Database sequence integrity required |
| SCMS-F-003 | The system shall categorize changes as Emergency, Standard, or Normal with different approval paths | Functional | High | ITIL Process | AC-SCMS-F-003: Category selection affects workflow routing, emergency changes bypass standard approval. Test: TC-CR-03 | Workflow engine dependency |

### **4.2 Impact Analysis and Dependencies**

| **Req ID** | **Requirement** | **Type** | **Priority** | **Source/Stakeholder** | **Acceptance Criteria** | **Comments/Dependencies** |
| --- | --- | --- | --- | --- | --- | --- |
| SCMS-F-004 | The system shall perform automated dependency analysis by scanning code repositories and architectural documentation | Functional | High | Architecture Team | AC-SCMS-F-004: System identifies affected modules, services, and dependencies automatically. Test: TC-IA-01 | Requires code parsing capabilities |
| SCMS-F-005 | The system shall detect potential conflicts with other pending change requests | Functional | High | Development Team | AC-SCMS-F-005: Conflict detection shows overlapping file changes, dependency conflicts. Test: TC-IA-02 | Real-time Git integration required |
| SCMS-F-006 | The system shall provide risk scoring based on change complexity, affected systems, and historical data | Functional | Medium | Risk Management | AC-SCMS-F-006: Risk score calculated using defined algorithm, displayed in approval interface. Test: TC-IA-03 | Historical data analysis required |
| SCMS-F-018 | The system shall allow tracking implementation status of each change request through defined states (In Progress, In Review, Verified, Ready for Deploy) | Functional | High | Change Manager | AC-SCMS-F-018: CR dashboard shows current state; state transitions are logged in audit trail. Test: TC-CR-04 | Workflow and audit trail integration |

### **4.3 Approval Workflow Management**

| **Req ID** | **Requirement** | **Type** | **Priority** | **Source/Stakeholder** | **Acceptance Criteria** | **Comments/Dependencies** |
| --- | --- | --- | --- | --- | --- | --- |
| SCMS-F-007 | The system shall route change requests through configurable approval workflows based on risk level and change type | Functional | High | Change Advisory Board | AC-SCMS-F-007: Workflow routes correctly based on rules, parallel/sequential approvals supported. Test: TC-WF-01 | Workflow engine required |
| SCMS-F-008 | The system shall allow approvers to approve, reject, or request modifications with mandatory comments | Functional | High | Management | AC-SCMS-F-008: All approval actions require comments, decisions are recorded with timestamps. Test: TC-WF-02 | Audit trail integration |
| SCMS-F-009 | The system shall support emergency change procedures with post-implementation approval | Functional | High | Operations | AC-SCMS-F-009: Emergency changes can be implemented immediately with required post-approval process. Test: TC-WF-03 | Emergency workflow paths |
| SCMS-F-021\* | The system shall coordinate CAB meetings with agenda, attendees, and decisions linked to CRs | Functional | Medium | Change Manager | AC-SCMS-F-021: CAB events stored; decisions auto-attach to CRs; notifications sent. Test: TC-WF-04 | Calendar/notification integration |

### **4.4 Version Control Integration**

| **Req ID** | **Requirement** | **Type** | **Priority** | **Source/Stakeholder** | **Acceptance Criteria** | **Comments/Dependencies** |
| --- | --- | --- | --- | --- | --- | --- |
| SCMS-F-010 | The system shall integrate with Git repositories to link commits with change requests using commit messages or tags | Functional | High | Development Team | AC-SCMS-F-010: Commits automatically linked to CRs via CR ID in commit message. Test: TC-GIT-01 | Git webhook integration |
| SCMS-F-011 | The system shall automatically update CR status when associated commits are merged to target branches | Functional | High | DevOps Team | AC-SCMS-F-011: CR status changes to "Implemented" when commits merged to specified branches. Test: TC-GIT-02 | Branch monitoring required |
| SCMS-F-012 | The system shall maintain traceability between change requests, commits, and deployed versions | Functional | High | Compliance | AC-SCMS-F-012: Complete audit trail from CR to deployment version maintained. Test: TC-GIT-03 | Deployment pipeline integration |
| SCMS-F-024\* | The system shall setup and manage Git webhooks to receive commit/merge events | Functional | High | DevOps Engineer | AC-SCMS-F-024: Webhook registration and secret verification succeed. Test: TC-GIT-04 | Git hosting platform |

### **4.5 Deployment and Release Management**

| **Req ID** | **Requirement** | **Type** | **Priority** | **Source/Stakeholder** | **Acceptance Criteria** | **Comments/Dependencies** |
| --- | --- | --- | --- | --- | --- | --- |
| SCMS-F-013 | The system shall integrate with CI/CD pipelines to track deployment status and success/failure | Functional | High | DevOps Team | AC-SCMS-F-013: Pipeline status reflected in CR, deployment failures trigger notifications. Test: TC-DEP-01 | CI/CD webhook integration |
| SCMS-F-014 | The system shall support rollback procedures with automated rollback triggers based on defined criteria | Functional | High | Operations | AC-SCMS-F-014: Rollback procedures defined per CR, automatic triggers configurable. Test: TC-DEP-02 | Deployment tooling integration |
| SCMS-F-015 | The system shall coordinate release windows with calendar integration and resource scheduling | Functional | Medium | Release Management | AC-SCMS-F-015: Release calendar shows conflicts, resource availability checked. Test: TC-DEP-03 | Calendar API integration |
| SCMS-F-023 | The system shall configure CI/CD integrations per project (endpoints, credentials, jobs, environments) | Functional | High | DevOps Engineer | AC-SCMS-F-023: Setup screen validates credentials and endpoints. Test: TC-DEP-05 | Pipeline configuration |
| SCMS-F-025 | The system shall manage infrastructure configuration metadata (environments, freeze windows) | Functional | Medium | System Administrator | AC-SCMS-F-025: Environment registry editable with RBAC; freeze windows enforced. Test: TC-ADM-01 | Infra configuration DB |
| SCMS-F-026 | The system shall configure notification channels (Email, Slack, Teams) for deployments and rollbacks | Functional | Medium | Release Manager | AC-SCMS-F-026: Notifications sent on trigger events; test message function works. Test: TC-NOTIF-01 | Notification API |
| SCMS-F-027 | The system shall validate deployment health using pipeline/job checks and service probes | Functional | High | System Administrator | AC-SCMS-F-027: Health status displayed; failures escalate per rules. Test: TC-DEP-06 | Health monitoring tools |
| SCMS-F-028 | The system shall generate deployment reports (success/failure, duration, approvers, artifacts) | Functional | Medium | Release Manager | AC-SCMS-F-028: Reports available per release; exportable. Test: TC-DEP-07 | Report generator |
| SCMS-F-029 | The system shall trigger pipeline builds and deployments directly from a CR or release page | Functional | High | DevOps Engineer | AC-SCMS-F-029: Authorized users trigger jobs; job results linked back to CR. Test: TC-DEP-08 | CI/CD permissions |

### **4.6 Reporting and Analytics**

| **Req ID** | **Requirement** | **Type** | **Priority** | **Source/Stakeholder** | **Acceptance Criteria** | **Comments/Dependencies** |
| --- | --- | --- | --- | --- | --- | --- |
| SCMS-F-016 | The system shall generate real-time dashboards showing change request metrics, approval bottlenecks, and deployment success rates | Functional | Medium | Management | AC-SCMS-F-016: Dashboard updates in real-time, shows key metrics and trends. Test: TC-REP-01 | Analytics engine required |
| SCMS-F-017 | The system shall export compliance reports in PDF/Excel formats with customizable date ranges and filters | Functional | Medium | Compliance | AC-SCMS-F-017: Reports generated with specified formats, filters applied correctly. Test: TC-REP-02 | Report generation engine |
| SCMS-F-022 | The system shall allow auditors to review immutable audit logs filtered by CR, user, and date | Functional | High | Auditor | AC-SCMS-F-022: Read-only log viewer; export to CSV/PDF. Test: TC-AUD-01 | Immutable audit log storage |
| SCMS-F-019 | The system shall validate test coverage for each change request by integrating CI results and mapping them to the CR | Functional | High | QA Engineer | AC-SCMS-F-019: Coverage percentage and pass/fail status displayed per CR; blocking rules configurable. Test: TC-CR-05 | Requires CI/CD integration and test result parsing |

## **5. Non-Functional Requirements**

| **Req ID** | **Requirement** | **Category** | **Priority** | **Acceptance Criteria/Measurement** |
| --- | --- | --- | --- | --- |
| SCMS-NF-001 | System shall maintain 99.5% uptime per month excluding scheduled maintenance windows | Availability | High | Monthly uptime reports show ≥99.5%. Test: TC-Perf-01 |
| SCMS-NF-002 | System shall respond within 2 seconds for 95% of change request submission and approval actions under normal load | Performance | High | Load testing shows 95th percentile ≤2s response time. Test: TC-Perf-02 |
| SCMS-NF-003 | System shall support 1000 concurrent users with degraded performance acceptable up to 1500 users | Scalability | High | Concurrent user testing validates performance thresholds. Test: TC-Perf-03 |
| SCMS-NF-004 | System shall encrypt all sensitive data at rest using AES-256 and in transit using TLS 1.3 | Security | High | Security audit confirms encryption standards compliance. Test: TC-Sec-01 |
| SCMS-NF-005 | System shall provide responsive web interface compatible with desktop and mobile devices meeting WCAG 2.1 AA standards | Usability | Medium | Accessibility audit and cross-device testing pass. Test: TC-UX-01 |
| SCMS-NF-006 | System shall archive all change management logs and records for at least 7 years to support compliance audits | Compliance | Medium | Archived data retrievable on request; storage verified. Test: TC-Comp-01 |
| SCMS-NF-007 | System shall ensure Recovery Time Objective (RTO) of 1 hour and Recovery Point Objective (RPO) of 15 minutes | Reliability | High | DR drill validates ≤1h recovery and ≤15m data loss. Test: TC-DR-01 |
| SCMS-NF-008 | System shall support zero-downtime upgrades using blue-green or rolling deployment strategies | Maintainability | Medium | Deployment test shows uninterrupted service during upgrades. Test: TC-DEP-09 |
| SCMS-NF-009 | System shall allow export of audit logs in CSV and PDF formats for regulatory review | Auditability | Medium | Logs exported successfully in CSV/PDF with filters. Test: TC-AUD-02 |

## **5.1 Security**

### **5.1.1 Security Objectives**

1. **Ensure confidentiality, integrity, and availability** of change request data, source code references, and audit trails to prevent unauthorized access and data tampering.
2. **Implement defense-in-depth security controls** including multi-factor authentication, role-based access control, and comprehensive audit logging to prevent unauthorized system access and maintain compliance with security standards.

### **5.1.2 Security Requirements**

| **Req ID** | **Requirement** | **Type** | **Priority** | **Acceptance Criteria/Test Case Ref** |
| --- | --- | --- | --- | --- |
| SCMS-SR-001 | TLS 1.3 mandatory for all network communications with certificate validation | Security | High | AC-SCMS-SR-001: All connections use TLS 1.3, certificate validation enforced. Test: TC-SEC-01 |
| SCMS-SR-002 | Multi-factor authentication required for Change Managers, Release Managers, and Auditors | Security | High | AC-SCMS-SR-002: MFA enforced for specified roles, multiple auth methods supported. Test: TC-SEC-02 |
| SCMS-SR-003 | Account lockout after 5 failed login attempts with progressive delay and admin unlock capability | Security | High | AC-SCMS-SR-003: Account locks after 5 failures, progressive delays implemented. Test: TC-SEC-03 |
| SCMS-SR-004 | Comprehensive audit logging of all critical change management actions with immutable storage | Security | High | AC-SCMS-SR-004: All actions logged with user/timestamp, logs tamper-evident. Test: TC-SEC-04 |
| SCMS-SR-005 | Regular security vulnerability scanning with automated dependency checking for integrated systems | Security | Medium | AC-SCMS-SR-005: Weekly vulnerability scans, dependency checks integrated in CI/CD. Test: TC-SEC-05 |
| SCMS-SR-006 | System shall authenticate all webhook and CI/CD pipeline integrations using secure tokens or keys | Security | High | AC-SCMS-SR-006: Penetration testing validates secure authentication. Test: TC-SEC-06 |
| SCMS-SR-007 | System shall maintain immutable, tamper-proof audit logs accessible only to authorized auditors | Security | High | AC-SCMS-SR-007: Audit review verifies immutability and RBAC enforcement. Test: TC-SEC-07 |

## **6. Quality Attributes & Acceptance Tests**

**Exit Criteria for Acceptance:**

* All high-priority functional requirements implemented and verified through automated and manual testing
* No critical non-functional requirement failures
* Security requirements validated through penetration testing
* Integration testing with Git, CI/CD, and notification systems completed
* Performance benchmarks met under specified load conditions
* Complete Requirements Traceability Matrix with all test cases passed

**Acceptance Test Suites:**

* **Change Request Management**: Submission, categorization, lifecycle management
* **Integration Testing**: Git, CI/CD, Jira, notification systems
* **Approval Workflow**: Multi-stage approvals, emergency procedures, CAB integration
* **Deployment Coordination**: Pipeline integration, rollback procedures
* **Performance Testing**: Load testing, concurrent user scenarios
* **Security Testing**: Authentication, authorization, data protection
* **Compliance Testing**: Audit trail validation, regulatory compliance

## **7. System Models and Diagrams**

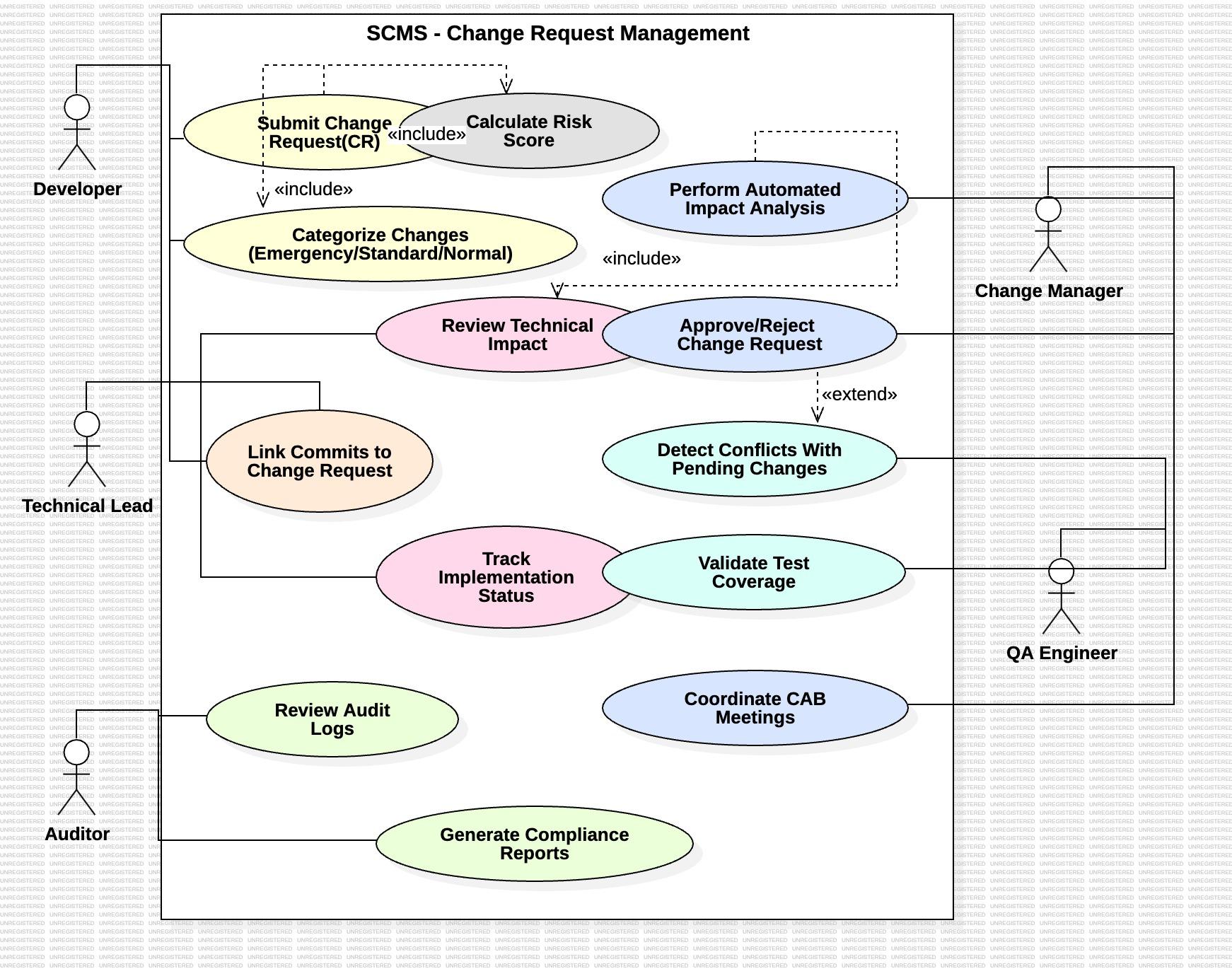
### **7.1 UML Use-Case Diagrams**

**Diagram 1: Change Request Lifecycle Management**

Actors: Developer, Technical Lead, Change Manager, QA Engineer

Use Cases:

* Submit Change Request
* Review Technical Impact
* Perform Impact Analysis
* Approve/Reject Change
* Link to Source Code
* Track Implementation Status

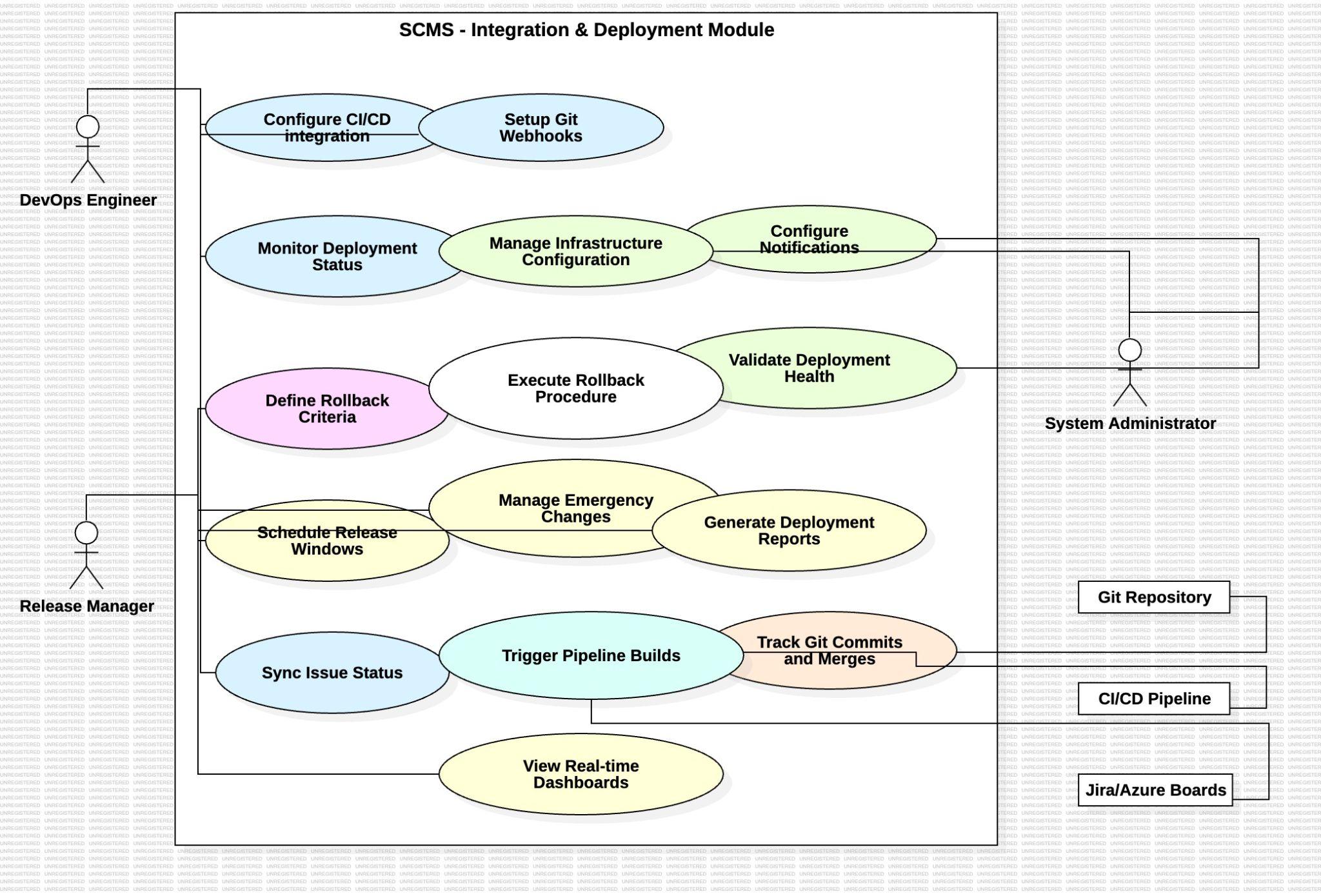


**Diagram 2: Integration and Deployment Workflow**

Actors: DevOps Engineer, Release Manager, System Administrator

Use Cases:

* Configure CI/CD Integration
* Monitor Deployment Status
* Execute Rollback Procedure
* Generate Compliance Reports
* Manage Emergency Changes
* Schedule Release Windows



## **8. Requirements Traceability Matrix (RTM)**

| **Req ID** | **Requirement Short** | **Section Ref** | **Module** | **Test Case(s)** | **Status** | **Comments** |
| --- | --- | --- | --- | --- | --- | --- |
| SCMS-F-001 | Submit CR with attachments | 4.1 | CR Management Module | TC-CR-01 | N | File upload validation required |
| SCMS-F-002 | Generate unique CR ID | 4.1 | CR Management Module | TC-CR-02 | N | Database sequence implementation |
| SCMS-F-003 | Categorize changes by type | 4.1 | Workflow Module | TC-CR-03 | N | Workflow engine dependency |
| SCMS-F-004 | Automated dependency analysis | 4.2 | Analysis Module | TC-IA-01 | N | Code parsing integration |
| SCMS-F-005 | Conflict detection | 4.2 | Analysis Module | TC-IA-02 | N | Git integration required |
| SCMS-F-006 | Risk scoring algorithm | 4.2 | Analysis Module | TC-IA-03 | N | Historical data analysis |
| SCMS-F-007 | Configurable approval workflows | 4.3 | Workflow Module | TC-WF-01 | N | Workflow engine configuration |
| SCMS-F-008 | Approval actions with comments | 4.3 | Approval Module | TC-WF-02 | N | Audit trail integration |
| SCMS-F-009 | Emergency change procedures | 4.3 | Workflow Module | TC-WF-03 | N | Emergency workflow paths |
| SCMS-F-010 | Git commit linking | 4.4 | Integration Module | TC-GIT-01 | N | Git webhook integration |
| SCMS-F-011 | Auto-update from Git merges | 4.4 | Integration Module | TC-GIT-02 | N | Branch monitoring service |
| SCMS-F-012 | Change-to-deployment traceability | 4.4 | Integration Module | TC-GIT-03 | N | Deployment pipeline hooks |
| SCMS-F-013 | CI/CD pipeline integration | 4.5 | Deployment Module | TC-DEP-01 | N | Pipeline API integration |
| SCMS-F-014 | Automated rollback procedures | 4.5 | Deployment Module | TC-DEP-02 | N | Deployment tooling required |
| SCMS-F-015 | Release window coordination | 4.5 | Scheduling Module | TC-DEP-03 | N | Calendar API integration |
| SCMS-F-016 | Real-time dashboards | 4.6 | Reporting Module | TC-REP-01 | N | Analytics engine required |
| SCMS-F-017 | Export compliance reports | 4.6 | Reporting Module | TC-REP-02 | N | Report generation engine |
| SCMS-NF-001 | 99.5% monthly uptime | 5 | Infrastructure | TC-Perf-01 | N | Monitoring integration |
| SCMS-NF-002 | 2-second response time | 5 | Performance Module | TC-Perf-02 | N | Load testing required |
| SCMS-NF-003 | 1000 concurrent users | 5 | Scalability Module | TC-Perf-03 | N | Load balancing required |
| SCMS-NF-004 | Data encryption standards | 5 | Security Module | TC-Sec-01 | N | Encryption library integration |
| SCMS-NF-005 | Cross-device compatibility | 5 | UI Module | TC-UX-01 | N | Responsive design testing |
| SCMS-SR-001 | TLS 1.3 enforcement | 5.1.2 | Security Module | TC-SEC-01 | N | SSL configuration |
| SCMS-SR-002 | MFA for privileged users | 5.1.2 | Authentication Module | TC-SEC-02 | N | MFA provider integration |
| SCMS-SR-003 | Account lockout policy | 5.1.2 | Authentication Module | TC-SEC-03 | N | Login attempt tracking |
| SCMS-SR-004 | Comprehensive audit logging | 5.1.2 | Audit Module | TC-SEC-04 | N | Immutable log storage |
| SCMS-SR-005 | Vulnerability scanning | 5.1.2 | Security Module | TC-SEC-05 | N | Security scanning tools |
| SCMS-F-018 | Track CR implementation status | 4.2 | CR Management Module | TC-CR-04 | N | Workflow and audit trail required |
| SCMS-F-019 | Validate CI test coverage per CR | 4.6 | QA Module | TC-CR-05 | N | Requires CI/CD integration |
| SCMS-F-021 | Coordinate CAB meetings | 4.3 | Workflow Module | TC-WF-04 | N | Calendar/notification integration |
| SCMS-F-023 | Configure CI/CD integration settings | 4.5 | Deployment Module | TC-DEP-05 | N | Pipeline configuration |
| SCMS-F-024 | Manage Git webhooks for commit/merge events | 4.4 | Integration Module | TC-GIT-04 | N | Git hosting platform |
| SCMS-F-025 | Manage infra configuration & freeze windows | 4.5 | Admin Module | TC-ADM-01 | N | Infra config DB |
| SCMS-F-026 | Configure notification channels | 4.5 | Notification Module | TC-NOTIF-01 | N | Notification API |
| SCMS-F-027 | Validate deployment health | 4.5 | System Monitoring | TC-DEP-06 | N | Health monitoring tools |
| SCMS-F-028 | Generate deployment reports | 4.5 | Reporting Module | TC-DEP-07 | N | Report generator |
| SCMS-F-029 | Trigger pipeline builds from CR/release page | 4.5 | Deployment Module | TC-DEP-08 | N | CI/CD permissions |
| SCMS-NF-006 | System shall archive change logs ≥7 years | 5 | Compliance Module | TC-Comp-01 | N | Long-term storage |
| SCMS-NF-007 | RTO ≤ 1 hour, RPO ≤ 15 minutes | 5 | Disaster Recovery | TC-DR-01 | N | Backup/replication testing |
| SCMS-NF-008 | System updates with zero downtime via blue-green deployment | 5 | Deployment Module | TC-DEP-09 | N | Blue/green strategy |
| SCMS-NF-009 | Audit logs exportable in CSV/PDF formats | 5 | Audit Module | TC-AUD-02 | N | Export tool required |
| SCMS-SR-006 | Secure token authentication for webhooks/CI | 5.1.2 | Integration Module | TC-SEC-06 | N | Pen-test validation |
| SCMS-SR-007 | Immutable audit logs with RBAC access | 5.1.2 | Audit Module | TC-SEC-07 | N | RBAC enforcement required |