SOPs, checklists, templates, diagrams & other documents creation:

color code: Completed InProgress NotStarted

PM SOPs, checklists, templates and guidelines

1. Standard Operating Procedures (SOPs)

Project Initiation SOP

1. Purpose

The purpose of this Standard Operating Procedure (SOP) is to define the process for initiating a project within the organization, ensuring that all necessary steps are followed to properly assess, approve, and plan a new project. This SOP will be used by project managers, sponsors, and stakeholders to standardize the initiation phase, ensuring alignment with organizational goals and strategic priorities.

2. Scope

This SOP applies to all new software development projects undertaken by MTN or China Mobile. It covers activities related to defining project goals, identifying stakeholders, gathering requirements, and preparing the initial project documentation. The SOP is intended for use by project managers, development teams, and stakeholders.

3. Responsibilities

- **Project Manager**: Responsible for coordinating the project initiation process, defining project goals, identifying stakeholders, and preparing the project charter.

- **Stakeholders**: Provide input on project requirements, expectations, and constraints.

- **Development Team Lead**: Assist in identifying technical requirements and resource needs.

4. Procedures

4.1 Project Request Submission

➢ **Document Project Concept:** Create a project concept document that includes the project objectives, expected outcomes, and alignment with organizational goals. Reference the Project Concept Document created in Templates

➢ **Feasibility Assessment:** Evaluate the feasibility of the project, considering alignment with business objectives, available resources, and any potential risks.

- **Approval to Undertake Project:** Obtain approval from stakeholders and sponsors to proceed with the project initiation phase.

4.2 Define Project Objectives and Scope

- **Identify Project Goals**: Clearly define the purpose and objectives of the project, including measurable success criteria.

- **Develop Project Scope Statement:** Document the project boundaries, including what is included and excluded. Reference the Project Scope Statement template created in Templates

4.3 Identify Stakeholders

- **List Stakeholders:** Identify all key stakeholders, including internal and external parties, such as business units, customers, and regulatory bodies.

- **Stakeholder Analysis:** Conduct a stakeholder analysis to determine stakeholder needs, influence, and level of involvement.

4.4 Develop Project Charter

- **Create Project Charter:** Prepare the project charter, detailing project objectives, scope, stakeholders, timeline, and resources required. Reference the Project Charter Template from Templates

- **Obtain Approval:** Circulate the project charter for review and approval by key stakeholders and project sponsors.

4.5 Allocate Resources

- **Resource Identification:** Identify the resources needed for the project, including personnel, tools, and technology.

- **Assign Roles and Responsibilities:** Assign roles and responsibilities to team members, ensuring that each role is clearly defined and communicated.

4.6 Risk Identification and Initial Assessment

- **Identify Potential Risks**: Conduct a preliminary risk assessment to identify potential risks that may impact the project.

- **Document Risks**: Use the Risk Management Plan Template to document identified risks, their potential impact, and initial mitigation strategies.

4.7 Initial Timeline and Budget Estimate

- **Develop High-Level Timeline:** Create a high-level project timeline, including major milestones and deliverables.
- **Estimate Budget:** Prepare an initial budget estimate based on project scope, resources, and potential risks. Include a contingency buffer to accommodate unforeseen changes.

4.8 Conduct Project Kickoff Meeting

- **Schedule Kickoff Meeting:** Schedule a project kickoff meeting with all stakeholders and team members.
- **Present Project Overview:** Present the project objectives, scope, timeline, and roles to all attendees, ensuring alignment and understanding.
- **Document Action Items:** Capture and document action items and follow-up tasks resulting from the kickoff meeting.

5. Deliverables

- **Project Concept Document**: A document outlining the project's concept, objectives, and expected outcomes.
- **Feasibility Assessment Report**: A report evaluating the project's feasibility and alignment with organizational goals.
- **Project Charter:** A formal document outlining the project objectives, scope, stakeholders, and key milestones.
- **Stakeholder Engagement & Analysis Report**: Use the Stakeholder Engagement checklist to create a document detailing stakeholder needs, influence, and engagement strategies.
- **Initial Risk Assessment and Management**: A document listing identified risks, their impact, and proposed mitigation strategies.
- **Kickoff Meeting Minutes**: Notes from the project kickoff meeting, including action items and responsibilities.

6. References

- Project Concept Document Form (see templates)
- Feasibility Assessment Report Template (see templates)
- Project Charter Template (see templates)
- Risk Management Plan Template (see templates)
- Project Scope Statement Template (see templates)
- Stakeholder Engagement Checklist (see checklists)

7. Revision History

- Version 1.0: Initial SOP created for project initiation.

Planning Phase SOP

1. Purpose

The purpose of this Standard Operating Procedure (SOP) is to define the steps involved in the planning phase of a project to ensure all necessary aspects are covered for successful execution. This SOP will guide project managers, team members, and stakeholders in developing a comprehensive project plan that aligns with organizational goals and expectations.

2. Scope

This SOP applies to all software development projects undertaken by MTN or China Mobile. It covers activities such as defining project scope, creating schedules, resource planning, budgeting, risk assessment, and establishing project baselines. This SOP is to be followed by project managers, stakeholders, and development teams during the planning phase.

3. Responsibilities

- **Project Manager**: Responsible for developing the project management plan, managing resources, and coordinating planning activities.
- **Team Members**: Provide input on task breakdown, resource requirements, and timelines.
- **Stakeholders**: Provide feedback and approvals for the project plan, budget, and schedule.

4. Procedures

4.1 Develop Project Scope Statement

1. **Define Project Boundaries**: Document what is included and excluded in the project, referencing the Project Scope Statement Template from Step 1.A.
2. **Confirm Scope with Stakeholders**: Obtain approval from stakeholders to ensure alignment with expectations.

4.2 Create Work Breakdown Structure (WBS) + Gantt chart

1. **Break Down Project into Tasks**: Develop a detailed WBS, breaking the project into smaller tasks and sub-tasks.
2. **Assign Task Owners**: Assign responsible team members for each task, using the WBS Template from Step 1.A.

4.3 Schedule Development

1. **Define Milestones**: Identify major project milestones and timelines, ensuring alignment with the overall project goals.
2. **Create Gantt Chart**: Develop a Gantt chart to illustrate task durations and dependencies, and reference the Implementation Schedule from Step 1.C.

4.4 Resource Planning

1. **Identify Resource Requirements**: Determine the resources required for each task, including personnel, equipment, and materials.
2. **Allocate Resources**: Assign resources to tasks, ensuring availability and considering workload balancing.

4.5 Budget Planning

1. **Estimate Costs**: Estimate costs for each task, including labor, equipment, materials, and contingency. Reference the Cost Estimation Template from Step 1.A.
2. **Develop Budget**: Develop a consolidated budget, ensuring alignment with project objectives and available funds.

4.6 Risk Management Planning

1. **Identify Risks**: Conduct a comprehensive risk assessment to identify potential risks, including technical, operational, and external risks.
2. **Develop Risk Management Plan**: Use the Risk Management Plan Template from Step 1.A to document risks, mitigation strategies, and contingency plans.

4.7 Communication Planning

1. **Define Communication Requirements**: Identify key stakeholders, communication frequency, and methods for sharing project information.

2. **Create Communication Plan**: Develop a communication plan that outlines who, how, and when information will be shared.

4.8 Quality Management Planning

1. **Define Quality Standards**: Establish quality standards for project deliverables, ensuring they align with organizational expectations.

2. **Develop Quality Assurance Plan**: Create a quality assurance plan that includes review processes, testing protocols, and acceptance criteria.

4.9 Develop Baselines

1. **Establish Scope Baseline**: Define the scope baseline, including the scope statement, WBS, and WBS dictionary.

2. **Establish Schedule and Cost Baselines**: Develop baselines for schedule and budget, which will be used to track project performance.

5. Deliverables

- **Project Scope Statement**: A document defining the boundaries of the project, including inclusions and exclusions.

- **Work Breakdown Structure (WBS) +Gantt chart**: A detailed breakdown of project tasks and sub-tasks.

- **Project Schedule**: A Gantt chart illustrating task durations, milestones, and dependencies.

- **Resource Plan**: A document outlining the resources required for the project, including personnel, tools, and equipment.

- **Project Budget**: A comprehensive budget detailing costs for each task, including contingency.

- **Risk Management Plan**: A document listing identified risks, their impact, mitigation strategies, and contingency plans.

- **Communication Plan**: A plan defining stakeholder communication needs, methods, and frequencies.
- **Quality Assurance Plan**: A plan detailing quality standards, testing protocols, and acceptance criteria.

6. References

- Project Scope Statement Template (see templates)
- Work Breakdown Structure (WBS) Template (see templates)
- Cost Estimation Template (see templates)
- Risk Management Plan Template (templates)
- Communication Plan Template (see templates)
- Quality Assurance Plan Template (see templates)
- Implementation Schedule (see Step 1.C)

7. Revision History

Version 1.0: Initial SOP created for planning phase.

Execution Phase SOP

1. Purpose

The purpose of this Standard Operating Procedure (SOP) is to outline the steps involved in executing a software development project to ensure that all tasks are carried out effectively, resources are properly utilized, and objectives are met. This SOP aims to guide project managers, team members, and stakeholders throughout the execution phase, ensuring alignment with project plans and successful delivery.

2. Scope

This SOP applies to all software development projects executed by MTN or China Mobile. It covers activities related to executing planned tasks, managing resources, maintaining quality, managing stakeholder communication, and ensuring progress toward project goals. This SOP is intended for project managers, development teams, and key stakeholders.

3. Responsibilities

- **Project Manager**: Responsible for coordinating execution activities, managing resources, tracking progress, and communicating with stakeholders.
- **Development Team**: Execute assigned tasks, adhere to quality standards, and provide updates on progress.
- **Stakeholders**: Provide input, feedback, and approvals during the execution phase as required.

4. Procedures

4.1 Task Assignment and Resource Allocation

- ➢ **Assign Tasks to Team Members**: Allocate tasks as per the Work Breakdown Structure (WBS) and ensure all team members understand their responsibilities.
- ➢ **Monitor Resource Utilization**: Track the allocation and utilization of resources to ensure efficiency and adjust as needed.

4.2 Monitor Project Progress

- ➢ **Track Task Completion**: Use project management tools to track the status of tasks and ensure they are completed according to the schedule.
- ➢ **Conduct Regular Stand-Up Meetings**: Schedule daily or weekly stand-up meetings to discuss progress, identify blockers, and ensure team alignment.

4.3 Quality Control

- ➢ **Conduct Code Reviews**: Implement peer reviews for code quality, ensuring adherence to coding standards and best practices.
- ➢ **Perform Testing**: Execute various testing levels, including unit, integration, and user acceptance testing (UAT), to ensure software meets quality requirements.

4.4 Risk Management

- **Monitor Identified Risks**: Continuously monitor identified risks using the Risk Management Plan and take necessary actions to mitigate them.

- **Identify New Risks**: Be vigilant for any new risks that arise during the execution phase, document them, and implement mitigation strategies.

4.5 Communication and Stakeholder Management

- **Update Stakeholders**: Provide regular updates to stakeholders on project progress, issues, and any deviations from the plan.

- **Manage Stakeholder Expectations**: Address stakeholder concerns promptly and ensure their expectations are managed throughout the execution.

4.6 DevSecOps Practices Integration

- **Automate Security Testing**: Integrate automated security testing into the CI/CD pipeline to identify vulnerabilities early in the development process.

- **Apply Security Patches**: Continuously monitor for security vulnerabilities and apply patches to maintain system integrity.

4.7 Cloud-Native Development Activities

- **Deploy to Cloud Environment**: Use cloud-native tools and IaC (Infrastructure as Code) for deploying software to the cloud environment, ensuring scalability and reliability.

- **Monitor Cloud Resources**: Monitor cloud resource utilization to ensure efficient use and adjust based on performance metrics.

4.8 AI/ML Integration

- **Train and Deploy Models**: If applicable, train AI/ML models using the prepared datasets and integrate them into the software application.

- **Monitor Model Performance**: Continuously monitor the performance of AI/ML models and adjust training or parameters as needed.

5. Deliverables

- **Task Assignments and Progress Reports**: Documentation of task assignments, status updates, and any changes to the schedule.

- **Code Review Reports**: Records of code review sessions, including feedback and necessary corrections.
- **Test Reports**: Results of unit, integration, and user acceptance testing, documenting any issues identified and resolutions.
- **Risk Log Updates**: An updated risk log that includes newly identified risks, mitigation strategies, and actions taken.
- **Stakeholder Update Reports**: Regular reports shared with stakeholders to keep them informed about progress, issues, and any changes.
- **Cloud Deployment Documentation**: Records of deployment to the cloud environment, including configurations and resource allocation details.

6. References

- Work Breakdown Structure (WBS) Template (see Step 1.A)
- Risk Management Plan Template (see Step 1.A)
- Quality Assurance Plan (see Step 1.B)
- Communication Plan Template (see Step 1.A)
- DevSecOps Security Requirements Template (see Step 1.A)
- Cloud Resource Planning Template (see Step 1.A)

7. Revision History

Version 1.0: Initial SOP created for the execution phase.

Monitoring and Controlling SOP

1. Purpose

The purpose of this Standard Operating Procedure (SOP) is to establish the processes required to monitor project progress, control deviations, and ensure that project objectives are met efficiently. This SOP aims to help project managers, team members, and stakeholders identify issues early, implement corrective actions, and maintain alignment with the project plan.

2. Scope

This SOP applies to all software development projects managed by MTN or China Mobile. It covers activities related to monitoring project performance, managing changes, controlling scope, schedule, cost, quality, and addressing risks. This SOP is intended for use by project managers, stakeholders, and development teams throughout the monitoring and controlling phase.

3. Responsibilities

- **Project Manager**: Responsible for tracking project progress, implementing corrective actions, managing changes, and ensuring alignment with project baselines.
- **Development Team**: Provide regular updates on task progress, identify issues, and assist in resolving problems.
- **Stakeholders**: Review progress reports, provide feedback, and approve any required changes.

4. Procedures

4.1 Performance Monitoring

- **Track Progress Against Baselines**: Use the project schedule, cost, and scope baselines to measure actual performance. Compare planned vs. actual progress and identify variances.
- **Use Performance Metrics**: Track key performance indicators (KPIs) such as schedule variance (SV), cost variance (CV), and earned value (EV) to measure project health.

4.2 Issue and Change Management

- **Identify Issues**: Continuously monitor project activities to identify issues that may impact project performance.
- **Log and Address Issues**: Maintain an issue log and prioritize issues for resolution. Implement corrective actions as needed.
- **Change Request Process**: Use the Change Request Form to document any requested changes. Evaluate the impact on scope, schedule, and budget before obtaining stakeholder approval.

4.3 Risk Monitoring and Control

- **Monitor Identified Risks**: Regularly monitor risks using the Risk Management Plan. Assess the effectiveness of mitigation strategies.
- **Identify New Risks**: Be vigilant for emerging risks, document them, and update the risk response plan accordingly.

4.4 Quality Control

- **Conduct Quality Checks**: Perform quality inspections and audits to verify that deliverables meet predefined standards.
- **Identify Defects**: Log any defects found during testing or quality checks. Assign corrective actions to relevant team members.

4.5 Communication and Reporting

- **Status Reporting**: Generate regular status reports to communicate project health, risks, and issues to stakeholders.
- **Stakeholder Meetings**: Conduct periodic review meetings with stakeholders to discuss project status, risks, and required changes.

4.6 DevSecOps Monitoring

- **Security Monitoring**: Continuously monitor the software for security vulnerabilities using automated DevSecOps tools.
- **Apply Remediation Measures**: When vulnerabilities are detected, apply remediation measures, and update the security risk log.

4.7 Cloud Resource Monitoring

- **Monitor Cloud Usage**: Track cloud resource usage to ensure efficiency and cost-effectiveness.
- **Optimize Resource Allocation**: Adjust cloud resources based on performance metrics to ensure optimal utilization.

4.8 AI/ML Model Performance Monitoring

- **Track Model Accuracy**: Monitor the performance of AI/ML models to ensure they meet desired accuracy and reliability standards.
- **Retrain Models if Necessary**: If model performance degrades, initiate retraining using updated datasets to maintain effectiveness.

5. Deliverables

- **Performance Reports**: Reports detailing project status, including schedule, cost, and scope variances.
- **Issue Log**: A log of issues identified during the monitoring phase, including actions taken and resolution status.
- **Change Request Log**: A record of all change requests, including the impact assessment and approval status.
- **Quality Inspection Reports**: Documentation of quality checks performed, including any defects and corrective actions.
- **Risk Log Updates**: An updated risk log reflecting ongoing risk monitoring and any new risks identified.
- **Security Monitoring Reports**: Reports generated by DevSecOps tools detailing vulnerabilities and remediation actions taken.

6. References

- Project Schedule and Baselines (see Step 1.B)
- Risk Management Plan Template (see Step 1.A)
- Change Request Form (see Step 1.A)
- Quality Assurance Plan (see Step 1.B)
- Cloud Resource Monitoring Template (see Step 1.C)
- DevSecOps Security Requirements Template (see Step 1.A)

7. Revision History

Version 1.0: Initial SOP created for monitoring and controlling phase.

Project Closure SOP

1. Purpose

The purpose of this Standard Operating Procedure (SOP) is to establish the steps required for the formal closure of a project, ensuring that all deliverables are completed, stakeholders are satisfied, and lessons learned are documented. This SOP provides a framework for finalizing a project, transferring ownership, and formally closing the project within the organization.

2. Scope

This SOP applies to all software development projects completed by MTN or China Mobile. It covers activities such as deliverable handover, stakeholder acceptance, final documentation, performance review, and project archival. This SOP is intended for use by project managers, development teams, and stakeholders during the project closure phase.

3. Responsibilities

- **Project Manager**: Responsible for coordinating closure activities, ensuring all deliverables are completed, collecting feedback, and archiving project documentation.
- **Development Team**: Assist in final testing, documentation, and providing support during the handover process.
- **Stakeholders**: Review final deliverables, provide acceptance, and participate in post-project reviews.

4. Procedures

4.1 Final Deliverable Handover

- ➢ **Complete Final Testing**: Ensure all deliverables are fully tested, including final user acceptance testing (UAT), to confirm they meet quality standards.
- ➢ **Handover to Stakeholders**: Transfer completed deliverables to stakeholders or clients, including any user guides, technical documentation, and training materials.

4.2 Obtain Stakeholder Acceptance

- ➢ **Acceptance Sign-Off**: Obtain formal acceptance from stakeholders, confirming that all deliverables meet requirements and expectations.
- ➢ **Document Acceptance**: Use the Project Closure Report Template to document stakeholder acceptance and approval.

4.3 Performance Evaluation and Lessons Learned

- **Conduct Project Performance Review**: Evaluate project performance against the original scope, schedule, and budget. Identify any deviations and their reasons.
- **Document Lessons Learned**: Conduct a lessons learned meeting with the project team and key stakeholders. Document insights, successes, challenges, and recommendations for future projects.

4.4 Administrative Closure

- **Close Contracts**: Ensure all contracts related to the project, including vendor agreements and service contracts, are formally closed.
- **Financial Closure**: Finalize all project-related financials, including payments, invoicing, and budget reconciliation.

4.5 Project Archival

- **Archive Documentation**: Gather and archive all project documents, including the project charter, scope statement, risk register, change logs, and final reports.
- **Update Knowledge Base**: Update the organization's knowledge base with relevant documentation, including lessons learned and best practices.

4.6 Celebrate Success

- **Acknowledge Team Efforts**: Recognize and celebrate the contributions of the project team, either through formal recognition or an informal event.
- **Communicate Success**: Communicate the successful completion of the project to all stakeholders and recognize key contributors.

5. Deliverables

- **Final Testing Report**: A document summarizing the results of final testing, including any issues identified and resolved.
- **Project Closure Report**: A formal report documenting stakeholder acceptance, final deliverables, and project outcomes.
- **Lessons Learned Document**: A document summarizing insights gained during the project, including successes, challenges, and recommendations.
- **Archived Project Documentation**: All project-related documents archived for future reference, including contracts, reports, and change logs.

- **Performance Evaluation Report**: A report summarizing the evaluation of project performance against initial goals and metrics.

6. References

- **Final Testing Report**: A document summarizing the results of final testing, including any issues identified and resolved.
- **Project Closure Report**: A formal report documenting stakeholder acceptance, final deliverables, and project outcomes.
- **Lessons Learned Document**: A document summarizing insights gained during the project, including successes, challenges, and recommendations.
- **Archived Project Documentation**: All project-related documents archived for future reference, including contracts, reports, and change logs.
- **Performance Evaluation Report**: A report summarizing the evaluation of project performance against initial goals and metrics.

6. References

- Project Closure Report Template (see Step 1.A)
- Lessons Learned Document Template (see Step 1.A)
- Project Charter (see Step 1.A)
- Risk Register (see Step 1.A)
- Change Request Form (see Step 1.A)
- User Acceptance Testing (UAT) Guidelines (see Step 1.B)

7. Revision History

Version 1.0: Initial SOP created for project closure.

DevSecOps SOP

1. Purpose

The purpose of this Standard Operating Procedure (SOP) is to integrate security into every phase of the software development and operations lifecycle, ensuring that security is prioritized alongside development and deployment activities. This SOP aims to guide project managers, developers, security teams, and operations teams in implementing DevSecOps practices effectively, minimizing risks, and enhancing software security.

2. Scope

This SOP applies to all software development projects undertaken by MTN or China Mobile. It covers activities from secure coding practices, vulnerability assessments, to security monitoring in production. This SOP is intended for use by development teams, security teams, operations teams, and project managers.

3. Responsibilities

- **Project Manager**: Ensure that security practices are integrated into the development lifecycle and that all team members understand their security responsibilities.
- **Development Team**: Implement secure coding practices, conduct code reviews, and address security vulnerabilities.
- **Security Team**: Conduct vulnerability assessments, penetration testing, and provide recommendations for risk mitigation.
- **Operations Team**: Ensure security monitoring, incident detection, and response capabilities are in place in the production environment.

4. Procedures

4.1 Secure Coding Practices

- **Follow Secure Coding Standards**: Adhere to secure coding guidelines, such as OWASP Top 10, to mitigate common vulnerabilities like SQL injection and cross-site scripting (XSS).
- **Code Review**: Implement peer code reviews focused on identifying and addressing security vulnerabilities early in the development process.

4.2 Automated Security Testing

- **Integrate Static Application Security Testing (SAST)**: Use SAST tools to analyze source code and identify vulnerabilities during the development phase.
- **Dynamic Application Security Testing (DAST)**: Conduct DAST to identify vulnerabilities in a running application, simulating real-world attack scenarios.

4.3 Vulnerability Management

- **Regular Vulnerability Scans**: Schedule regular vulnerability scans using tools like Nessus or Qualys to detect potential security issues.

- **Patch Management**: Apply patches promptly to address known vulnerabilities. Track and prioritize patches based on their severity.

4.4 Continuous Integration and Continuous Deployment (CI/CD) Security

- **Integrate Security into CI/CD Pipeline**: Implement security checks as part of the CI/CD process, including automated testing for vulnerabilities and code quality.

- **Container Security**: Ensure container images used in the CI/CD pipeline are scanned for vulnerabilities and comply with security standards.

4.5 Cloud Security Practices

- **Implement Identity and Access Management (IAM)**: Apply the principle of least privilege to cloud resources, ensuring that permissions are limited to what is necessary.

- **Infrastructure as Code (IaC) Security**: Use tools like Terraform and AWS CloudFormation to deploy infrastructure securely, ensuring that configurations meet best practices.

4.6 Incident Detection and Response

- **Security Monitoring**: Implement real-time monitoring using tools like Splunk or ELK Stack to detect suspicious activities in the production environment.

- **Incident Response Plan**: Develop and maintain an incident response plan to handle security incidents effectively. Conduct regular incident response drills to prepare the team.

4.7 Compliance and Audit

- **Security Compliance Checks**: Ensure the software complies with relevant regulations (e.g., GDPR, PCI-DSS) and conduct regular audits to verify compliance.

- **Audit Trail**: Maintain a detailed audit trail of all security activities, including code reviews, vulnerability assessments, and incident responses.

5. Deliverables

- **Secure Coding Guidelines**: A document detailing secure coding practices, including references to OWASP Top 10.

- **Code Review Reports**: Records of code reviews conducted, including identified vulnerabilities and corrective actions.
- **Vulnerability Assessment Reports**: Reports detailing the results of vulnerability scans, penetration tests, and recommended actions.
- **CI/CD Pipeline Security Checklist**: A checklist used to verify the integration of security practices in the CI/CD pipeline.
- **Incident Response Plan**: A documented plan for responding to security incidents, including roles, responsibilities, and procedures.
- **Compliance Audit Report**: A report detailing the results of security compliance audits and any corrective measures taken.

6. References

- OWASP Top 10 Guidelines (see Step 1.A)
- Code Review Checklist (see Step 1.B)
- CI/CD Pipeline Template (see Step 1.B)
- Infrastructure as Code (IaC) Implementation Playbook (see Step 1.C)
- Incident Response Playbook (see Step 1.C)
- Cloud Security Best Practices (see Step 1.B)

7. Revision History

Version 1.0: Initial SOP created for DevSecOps integration.

Cloud-Native Implementation SOP

1. Purpose

2. Scope

3. Responsibilities

4. Procedures

5. Deliverables

6. References

7. Revision History

AI/ML Integration SOP

1. Purpose

2. Scope

3. Responsibilities

4. Procedures

5. Deliverables

6. References

7. Revision History

2. Checklists

Project Kickoff Checklist

1. Pre-Meeting Preparation

- ✓

2. Meeting Logistics

- ✓

3. During the Kickoff Meeting

- ✓

4. Post-Meeting Actions

- ✓

5. References

- Project Charter (see Step 1.A)
- Project Timeline (see Step 1.B)
- Communication Plan (see Step 1.B)

Stakeholder Engagement Checklist

1. Identify Stakeholders

- ✓

2. Stakeholder Analysis

- ✓

3. Develop Engagement Strategy

- ✓

4. Stakeholder Communication

- ✓

5. Feedback Collection and Analysis

- ✓

6. Monitor Stakeholder Engagement

- ✓

7. Post-Project Evaluation

- ✓

8. References

- **Stakeholder Engagement Plan Template** (see Step 1.B)
- **Communication Plan** (see Step 1.A)

Risk Assessment Checklist

1. Identify Risks

✓

2. Categorize Risks

✓

3. Risk Analysis

✓

4. Develop Risk Mitigation Strategies

✓

5. Establish Monitoring Mechanisms

✓

6. Communication

✓

7. Post-Project Evaluation

✓

8. References

- **Risk Register Template** (see Step 1.B)
- **Risk Probability and Impact Matrix Template** (see Step 1.B)
- **Contingency Plan Guidelines** (see Step 1.C)

Quality Assurance Checklist

1. Define Quality Standards

✓

2. Develop Test Plan

✓

3. Conduct Code Reviews

✓

4. Perform Testing

✓

5. Defect Tracking and Resolution

✓

6. Quality Metrics Tracking

✓

7. Document Quality Assurance Activities

✓

8. Post-Release Evaluation

✓

9. References

- **Test Plan Template** (see Step 1.A)
- **Defect Tracking Tool Guidelines** (see Step 1.B)
- **Quality Summary Report Template** (see Step 1.B)
- **User Acceptance Testing (UAT) Guidelines** (see Step 1.B)

Project Closure Checklist

1. Final Deliverable Completion

✓

2. Financial Closure

✓

3. Contract Closure

✓

4. Documentation and Archiving

✓

5. Post-Project Evaluation

- ✓

6. Team Recognition and Release

- ✓

7. Stakeholder Communication

- ✓

8. Post-Implementation Support

- ✓

9. References

- **Project Closure Report Template** (see Step 1.A)
- **Post-Project Evaluation Template** (see Step 1.B)
- **Lessons Learned Document Template** (see Step 1.A)
- **Team Recognition Guidelines** (see Step 1.C)

DevSecOps Implementation Checklist

1. Planning and Preparation

- ✓

2. Secure Development Practices

- ✓

3. CI/CD Pipeline Integration

- ✓

4. Container Security

- ✓

5. Identity and Access Management (IAM)

- ✓

6. Monitoring and Logging

- ✓

7. Vulnerability Management

✓

8. Compliance and Auditing

✓

9. Incident Response and Recovery

✓

10. Post-Implementation Review

✓

11. References

- **OWASP Top 10 Security Guidelines** (see Step 1.B)
- **CI/CD Security Integration Template** (see Step 1.B)
- **Incident Response Plan Template** (see Step 1.C)
- **Container Security Guidelines** (see Step 1.B)
- **IAM Best Practices** (see Step 1.C)

Cloud Readiness Checklist

3. Templates (includes Forms)

Project Concept Document form

Feasibility Assessment Report Template

Project Charter Template

Project Scope Statement Template

Work Breakdown Structure (WBS) + Gantt chart Template

Risk Management Plan Template

Communication Plan Template

Change Request Form

Project Status Report Template

Project Closure Report Template

Lessons Learned Document Template

DevSecOps Security Requirements Template

Cloud Resource Planning Template

AI/ML Integration Template

4. Guidelines for Use

General Project Management Guidelines

Cloud-Native Development Guidelines

DevSecOps Best Practices

AI/ML in Project Management

Stakeholder Communication Guidelines

Process SOPs, checklists, templates and guidelines

1. Standard Operating Procedures (SOPs)

Software Development Lifecycle (SDLC) SOP

Requirements Gathering SOP

DevSecOps SOP

Cloud-Native Development SOP

AI/ML Model Integration SOP

Testing and Quality Assurance SOP

Change Management SOP

Incident Management SOP

Continuous Integration/Continuous Deployment (CI/CD) SOP

2. Checklists

Requirements Validation Checklist

Code Review Checklist

DevSecOps Security Checklist

Cloud Readiness Checklist for Development

AI/ML Model Deployment Checklist

Testing Checklist

Deployment Readiness Checklist

Incident Resolution Checklist

Change Request Checklist

3. Templates (includes Forms)

Requirements Document Template

Software Design Document (SDD) Template

DevSecOps Pipeline Template

Cloud Deployment Template

AI/ML Model Evaluation Template

Test Plan Template

Change Request Form

Incident Report Template

Deployment Plan Template

4. Guidelines

Code Quality Guidelines

Cloud-Native Development Guidelines

AI/ML Integration Guidelines

CI/CD Best Practices Guidelines

User Acceptance Testing (UAT) Guidelines

Additional Documentation and flow description (flow diagrams)

1. Additional Documentation

Continuous Improvement Plan

DevSecOps Risk Mitigation Guide

AI/ML Integration Strategy Document

Cloud-Native Adoption Roadmap

Automated Testing Guidelines

Infrastructure as Code (IaC) Implementation Playbook

Ethical AI Guidelines

DevOps Team Collaboration SOP

Incident Response Playbook

Technical Debt Management Plan

Microservices Communication Guidelines

Cloud Cost Optimization Guidelines

Software Release Strategy Plan

Security Compliance Checklist for CI/CD

2. Diagrams

High-Level SDLC Process Flow Diagram

DevSecOps Workflow Diagram

Cloud-Native Architecture Diagram

AI/ML Integration Flow Diagram

CI/CD Pipeline Diagram

Incident Management Flowchart

Change Management Flow Diagram

Automated Testing Architecture Diagram

Microservices Interaction Diagram

Infrastructure as Code (IaC) Flow Diagram

Stakeholder Communication Flowchart

Risk Management Flow Diagram

Deployment Flow Diagram

Continuous Improvement Loop Diagram

Data Flow Diagram for AI/ML Processing

Implementation within the business and follow up review.

1. Implementation

Training Plan for Development Team

Stakeholder Engagement Strategy

Change Adoption Plan

Tool Integration Guide

Implementation Schedule

Environment Setup Checklist

Pilot Testing Plan

User Training Materials

Deployment Guidelines

Onboarding Guide for New Team Members

Compliance and Security Validation Plan

2. Follow-up Review

Post-Implementation Review (PIR) Report

Process Performance Evaluation

Stakeholder Satisfaction Survey

Continuous Improvement Plan

Compliance Audit Report

Retrospective Meetings and Documentation

Risk Assessment Review

Training Effectiveness Evaluation

Performance Benchmarking Report

Feedback Loop Mechanism

Support and Troubleshooting Guidelines

Final Handover Report