info@sdh-it.de | www.sdh.global

20537, Anckelmannspaltz 1, Hamburg, Germany

DevOps Lifecycle Framework

The DevOps management process follows the industry-standard “7 C’s of DevOps” lifecycle, which forms the foundation of modern software delivery practices task_1.

Prepared by SDH.global DevOps Services Team

---

1. Development

The initial phase where planning and coding occur in small, manageable cycles.

☐ **Planning & Requirements Analysis**
☐ Document business requirements and technical specifications
☐ Break down large projects into smaller, manageable tasks
☐ Prioritize development tasks based on business value
☐ **Code Management Implementation**
☐ Set up version control repositories (e.g., Git)
☐ Establish branching strategy (e.g., Git Flow, GitHub Flow)
☐ Define code review procedures and standards
☐ Implement code quality checks

Tools & Technologies:

- Jira for project management and task tracking
- Git for version control
- IDE plugins for code quality analysis

2. Integration

The phase where code changes are automatically integrated and tested.

☐ **Integration Server Setup**
☐ Install and configure CI server (e.g., Jenkins, CircleCI)
☐ Configure build pipelines
☐ Set up automated unit testing
☐ **Code Quality Validation**
☐ Implement static code analysis tools
☐ Set up security scanning tools
☐ Define quality gates and thresholds

Tools & Technologies:

- Jenkins or CircleCI for continuous integration
- SonarQube for code quality analysis
- JUnit and similar frameworks for automated testing

3. Testing

The phase focused on automated testing to validate functionality and quality.

☐ **Test Automation Implementation**
☐ Develop automated unit tests
☐ Create integration test suites
☐ Build end-to-end test scenarios
☐ Implement performance test frameworks
☐ **Test Environment Management**
☐ Set up isolated test environments
☐ Implement data management for testing
☐ Configure test reporting and dashboards

Tools & Technologies:
- Selenium for UI testing
- JMeter for performance testing
- Cucumber for behavior-driven testing

4. Deployment

The phase where validated code is automatically deployed to various environments.

☐ **Deployment Pipeline Configuration**
☐ Set up deployment automation scripts
☐ Configure environment-specific deployment parameters
☐ Implement deployment verification checks
☐ **Container Orchestration Setup**
☐ Configure container images and registries
☐ Set up Kubernetes clusters
☐ Define resource allocation and scaling policies

Tools & Technologies:
- Docker for containerization
- Kubernetes for orchestration
- Terraform for infrastructure provisioning

info@sdh-it.de | www.sdh.global
20537, Anckelmannspaltz 1, Hamburg, Germany

5. Monitoring

The phase where application and infrastructure performance are tracked in real-time.

☐ **Monitoring Infrastructure Setup**
☐ Deploy monitoring agents and collectors
☐ Configure system health checks
☐ Set up performance metric collection
☐ Implement log aggregation
☐ **Alert Configuration**
☐ Define alert thresholds
☐ Configure notification channels
☐ Create escalation procedures

Tools & Technologies:

- Prometheus for metrics collection
- Grafana for visualization
- ELK Stack for log management

6. Feedback

The phase where user feedback and system data are collected to drive improvements.

☐ **Feedback Collection Mechanisms**
☐ Implement user feedback collection tools
☐ Set up A/B testing frameworks
☐ Configure usage analytics
☐ **Feedback Analysis Process**
☐ Define data analysis procedures
☐ Schedule regular feedback review meetings
☐ Create improvement tracking system

Tools & Technologies:

- Datadog for performance insights
- Google Analytics for user behavior
- Pendo for feature usage tracking

7. Continuous Operations
The final phase ensuring systems remain operational and resilient.

☐ **Operational Automation**
☐ Implement infrastructure as code

☐ Configure automated backups
☐ Set up disaster recovery procedures
☐ **Infrastructure Management**
☐ Define scaling policies
☐ Implement configuration management
☐ Set up system health dashboards

Tools & Technologies:

- Terraform for infrastructure provisioning
- Chef or Ansible for configuration management
- Backup and disaster recovery tools

Appendix A: Tool Selection Guide

Tool Category	Common Tools	Selection Criteria
Version Control	Git, SVN, Mercurial	Distributed vs. centralized, integration capabilities, branching model
CI/CD	Jenkins, CircleCI, GitHub Actions, GitLab CI	Pipeline flexibility, integration options, scaling capabilities
Infrastructure as Code	Terraform, CloudFormation, Ansible	Cloud provider support, state management, community support
Containerization	Docker, Podman, containerd	Performance, security features, orchestration compatibility
Orchestration	Kubernetes, Docker Swarm, Nomad	Scaling capabilities, self-healing features, community support

Monitoring	Prometheus, Grafana, Datadog, New Relic	Metric types, visualization capabilities, alerting features
Logging	ELK Stack, Graylog, Splunk	Indexing capabilities, search performance, retention options
Security	Snyk, SonarQube, OWASP ZAP	Language support, integration options, remediation guidance

Appendix B: DevOps Maturity Assessment

- [ ] **Initial Assessment**
  - [ ] Conduct team surveys
  - [ ] Collect process metrics
  - [ ] Document current tools
  - [ ] Map current workflows
  - [ ] Identify bottlenecks
- [ ] **Maturity Scoring**
  - [ ] Score across key dimensions
  - [ ] Identify maturity level
  - [ ] Create maturity radar chart
  - [ ] Set improvement targets
  - [ ] Document findings

Document Version: 1.0
Last Updated: [Current Date]
© SDH.global DevOps Services

Note: This checklist is customizable based on specific organizational needs and existing DevOps maturity. Items can be added, removed, or modified to fit your particular context.

info@sdh-it.de | www.sdh.global

20537, Anckelmannspaltz 1, Hamburg, Germany