Software development lifecycle step-by-step guidance

Below are phase-by-phase actions, mandatory deliverables and practical checklists to guide implementation.

- Initiation
- Requirements and acquisition
- Architecture and design
- Development
- Testing
- Deployment
- Operations and Maintenance
- Upgrade / Decommission

Initiation

Goal: Establish security and privacy expectations and identify risks before design work begins.

Actions:

1. Appoint project sponsor, system owner and security lead.
2. Perform initial Threat and Privacy Risk Assessment (documented).
3. Define security and privacy objectives of CIA, non-repudiation and legal requirements.
4. Draft a Security and Privacy Plan with milestones, roles and budget for security activities.
5. Require security awareness briefing for project stakeholders.

Requirements and acquisition

Goal: Ensure requirements include explicit privacy and security criteria.

1. Define functional, privacy and security requirements. Include purpose limitation and data minimization requirements.
2. Conduct Privacy Impact Assessment (PIA) and update risk register.
3. Translate risks into measurable security requirements like encryption, RBAC, logging and retention.
4. For procurement: include security clauses, acceptance criteria and tender security evaluation.
5. For third parties: require evidence of prior security audits and contractual data protection obligations.

Architecture and design

Goal: Design an architecture that enforces privacy and security by construction.

1. Produce security architecture diagrams showing trust boundaries, data flows and classification.
2. Apply Data Flow Mapping and Data Classification (sensitive vs non-sensitive).
3. Embed privacy controls: data minimization, consent capture points, and user-facing
4. Specify encryption, key management, segmentation, and secure default configurations.
5. Plan for logging, monitoring, and auditability (what to log, how long, who has access).
6. Document fallback modes and failure behaviors to avoid privacy leaks or insecure defaults.

Development

Goal: Implement secure, privacy-aware code and configurations.

1. Adopt secure coding standards (OWASP, CERT) and include them in the definition of done.
2. Use automated static analysis (SAST), dependency scanning and secret detection in CI/CD pipelines.
3. Enforce strong access controls for development environments and use separate secrets management.
4. Perform regular code reviews focused on security and privacy by identifying hard-coded secrets and data exposures.
5. Implement privacy-enhancing techniques (pseudonymization, tokenization) where feasible.
6. Maintain secure build and deployment scripts; avoid embedding credentials in code.

Testing

Goal: Verify security and privacy controls work as intended.

1. Create a security test plan covering unit, integration, system, and acceptance tests.
2. Include privacy test cases validating consent, data minimization, and access controls.
3. Conduct vulnerability scanning and dynamic application security testing (DAST).
4. Arrange independent penetration testing for critical systems and production environments.
5. Perform usability testing to ensure privacy settings and notices are clear and actionable.
6. Run regression tests after patches and new features to prevent reintroducing vulnerabilities.

Deployment

Goal: Deploy securely with correct configurations, access controls and monitoring in place.

1. Apply secure configuration baselines and hardening to servers, databases and network devices.
2. Enforce RBAC and configure least privilege for all accounts; set up MFA for admin accounts.
3. Enable and protect audit logging; ensure log storage and retention meet policy requirements.
4. Conduct a production penetration test and address critical findings before go-live.
5. Publish privacy notices and provide user controls for consent and data management.
6. Establish monitoring and alerting (IDS/IPS, SIEM) and define on-call incident responders.

Operations and Maintenance

Goal: Sustain security and privacy posture throughout operations.

1. Maintain a schedule for vulnerability scanning, patch management, and configuration reviews.
2. Conduct periodic privacy and security control reviews and update PIAs as needed.
3. Ensure change management enforces security reviews and testing before changes are applied.
4. Continue training for administrators and users; run phishing and awareness programs.
5. Keep data retention schedules and securely sanitize or delete data when no longer required.
6. Keep an incident response plan current and conduct tabletop exercises regularly.

Upgrade / Decommission

Goal: Safely retire or replace systems while preserving required records and preventing data leakage.

1. Plan archival or migration of records according to legal retention requirements.
2. Sanitize media and verify secure deletion of sensitive data using approved methods.
3. Revoke access, disable accounts and remove credentials tied to decommissioned systems.
4. Update documentation to reflect where data was moved and how it can be accessed or destroyed.
5. Notify stakeholders and offer users guidance to export or delete their data where applicable.