SOFTWARE AND SYSTEMS DEVELOPMENT STANDARDS

JANUARY 2026

POLICY NUMBER: TMG-SEC-041
APPROVAL AUTHORITY: BOARD OF DIRECTORS

THE MAZI GROUP

1. PURPOSE

This Software and Systems Development Standards policy establishes guidelines for developing secure, high-quality software at The Mazi Group. Consistent development practices ensure our technology products—including Glam Switch and Upvendo platforms—are reliable, secure, and maintainable.

2. SCOPE

This policy applies to:

- All software development activities
- All development teams across brands
- Internal and customer-facing applications
- Mobile, web, and backend systems
- Third-party development partners

3. DEVELOPMENT METHODOLOGY

3.1 Agile Practices

- Iterative development approach
- Regular sprint cycles
- Continuous feedback and improvement
- Cross-functional teams
- Regular retrospectives

3.2 DevOps Principles

- Collaboration between development and operations
- Automation of build, test, and deployment
- Continuous integration and delivery
- Infrastructure as code
- Monitoring and observability

4. SECURE DEVELOPMENT LIFECYCLE

4.1 Security by Design

- Consider security from project inception
- Threat modeling for new features
- Security requirements in user stories
- Privacy by design principles

4.2 Secure Coding Practices

- Input validation and sanitization
- Output encoding
- Parameterized queries (prevent SQL injection)
- Authentication and session management
- Access control implementation
- Error handling without information disclosure
- Secure cryptographic practices

4.3 OWASP Guidelines

- Address OWASP Top 10 vulnerabilities
- Follow OWASP secure coding guidelines
- Use OWASP testing guide for security testing

5. CODE QUALITY

5.1 Coding Standards

- Consistent coding style per language
- Meaningful naming conventions
- Code documentation and comments
- Modular, maintainable code
- DRY (Don't Repeat Yourself) principle

5.2 Code Review

- All code changes require peer review
- Review for functionality, security, and quality
- Use pull request workflow
- Document review feedback
- Address all comments before merge

5.3 Static Analysis

- Automated code analysis tools
- Security scanning (SAST)
- Code quality metrics
- Address findings before deployment

6. VERSION CONTROL

- All code in version control (Git)
- Branching strategy (GitFlow or similar)
- Meaningful commit messages
- Protected main/production branches
- No secrets in version control

7. TESTING

7.1 Testing Requirements

- Unit tests for all code
- Integration testing
- End-to-end testing
- Performance testing
- Security testing
- Accessibility testing

7.2 Test Coverage

- Minimum code coverage targets
- Critical paths fully tested
- Regression test suites
- Automated test execution

7.3 Security Testing

- Dynamic application security testing (DAST)
- Penetration testing for major releases
- Vulnerability scanning
- API security testing

8. CONTINUOUS INTEGRATION/CONTINUOUS DEPLOYMENT

8.1 CI Pipeline

- Automated builds on commit
- Automated test execution
- Static analysis in pipeline
- Build artifact management

8.2 CD Pipeline

- Automated deployment to environments
- Environment promotion workflow
- Deployment approvals for production
- Rollback capabilities

8.3 Environment Management

- Separate development, staging, production
- Production-like staging environment
- Environment configuration management
- No production data in non-production

9. RELEASE MANAGEMENT

9.1 Release Process

- Defined release criteria
- Release approval process
- Release notes and documentation
- Communication to stakeholders

9.2 Change Management

- Change requests for production changes
- Impact assessment
- Approval workflow
- Change documentation

9.3 Deployment Windows

- Scheduled deployment windows
- Emergency change procedures
- Minimize customer impact

10. DOCUMENTATION

10.1 Technical Documentation

- Architecture documentation
- API documentation
- Database schemas
- Integration specifications
- Runbooks for operations

10.2 User Documentation

- User guides
- Help content
- Release notes

- Training materials

11. THIRD-PARTY COMPONENTS

11.1 Open Source

- Approved open source licenses
- License compliance tracking
- Security vulnerability monitoring
- Regular updates for security patches

11.2 Software Composition Analysis

- Inventory of third-party components
- Vulnerability scanning
- License scanning
- Remediation of vulnerabilities

12. API DEVELOPMENT

- RESTful API design principles
- API versioning strategy
- Authentication and authorization
- Rate limiting
- API documentation (OpenAPI/Swagger)
- Input validation

13. MOBILE DEVELOPMENT

- Platform-specific guidelines (iOS, Android)
- Secure data storage on device
- Certificate pinning
- App store compliance
- Privacy permissions handling

14. DATA HANDLING

- Data classification in development
- No real customer data in non-production
- Data masking/anonymization for testing
- Encryption of sensitive data
- Secure data deletion

15. INCIDENT RESPONSE

- Security vulnerability response process
- Bug severity classification
- Hotfix procedures
- Post-incident review

16. ROLES AND RESPONSIBILITIES

16.1 Developers

- Write secure, quality code
- Follow coding standards
- Write tests
- Participate in code reviews

16.2 Tech Leads

- Ensure standards compliance
- Conduct code reviews
- Technical decision-making
- Mentor developers

16.3 Security Team

- Define security requirements
- Conduct security reviews
- Manage security testing

- Respond to vulnerabilities

17. TRAINING

- Secure coding training for developers
- Technology-specific training
- Security awareness
- Ongoing skill development

18. RELATED POLICIES

This policy should be read in conjunction with:

- TMG-SEC-017: Cybersecurity and IT Security Policy
- TMG-SEC-038: AI Ethics and Responsible AI Use Policy
- TMG-SEC-040: Cloud Computing and SaaS Governance Policy

19. POLICY REVIEW

This policy shall be:

- Reviewed annually by Engineering leadership
- Updated for technology changes
- Approved by CTO
- Communicated to development teams

20. DEFINITIONS

- **CI/CD:** Continuous Integration/Continuous Deployment.
- **SAST:** Static Application Security Testing.
- **DAST:** Dynamic Application Security Testing.
- **OWASP:** Open Web Application Security Project.
- **API:** Application Programming Interface.

For questions about this policy, contact:
Technology Department

technology@mazigroup.us