

Agile Quality Assurance Plan Template

Emmanuel BRUNO
Université de Toulon, LIS UMR CNRS 7020
emmanuel.bruno@univ-tln.fr

2026-10-14

Table des matières

1 Agile Quality Assurance Plan Template . .	1
1.1 Integration with Agile Ceremonies .	1
1.2 Testing Pyramid Strategy	1
1.3 Test Categories	2
1.3.a Functional Testing	2
1.3.b Non-Functional Testing	2
1.4 Agile Testing Quadrants	2
1.5 Automated Testing Infrastructure ...	2
1.6 Continuous Integration & Testing . .	3
1.7 Manual Testing Procedures	3
1.8 Quality Metrics in Agile	3
1.9 Defect Management	3
1.10 Defect Management in Agile	3
1.11 Release Readiness	3
1.12 Release Strategy	3
1.13 Continuous Improvement	3
1.14 Adaptation & Improvement	4

1 Agile Quality Assurance Plan Template

This document provides a template for creating a comprehensive quality assurance plan aligned with Agile methodologies and principles.

1.1 Integration with Agile Ceremonies

1. Sprint Planning

- [Define DoD quality criteria]
- [Define testing scope per story]

- [Define test effort estimation]
- [Define acceptance criteria standards]

2. Daily Stand-ups

- [Define testing status reporting]
- [Define blocking issues protocol]
- [Define test environment updates]

3. Sprint Review

- [Define demo quality gates]
- [Define stakeholder feedback capture]
- [Define quality metrics review]

4. Sprint Retrospective

- [Define quality improvement discussions]
- [Define testing process feedback]
- [Define action items tracking]

1.2 Testing Pyramid Strategy

Define your testing distribution across different levels:

1. Unit Tests ([Target]%)

- Focus: [Define scope]
- Tools: [List primary tools]
- Coverage Target: [Define percentage]
- Responsibility: [Assign team]

2. Integration Tests ([Target]%)

- Focus: [Define scope]
- Tools: [List primary tools]
- Coverage Target: [Define metrics]
- Responsibility: [Assign team]

3. UI Tests ([Target]%)

- Focus: [Define scope]

- Tools: [List primary tools]
- Coverage Target: [Define metrics]
- Responsibility: [Assign team]

1.3 Test Categories

1.3.a Functional Testing

1. **Feature Testing**
 - [Define BDD approach]
 - [Define story testing workflow]
 - [Define in-sprint testing strategy]
2. **Regression Testing**
 - [Define automation priorities]
 - [Define sprint regression scope]
 - [Define risk-based approach]
3. **Edge Case Testing**
 - [List critical scenarios]
 - [Define boundaries]
 - [Define special conditions]

1.3.b Non-Functional Testing

1. **Performance Testing**
 - [Define response time targets]
 - [Define throughput requirements]
 - [Define resource utilization limits]
 - [Define scalability metrics]
2. **Usability Testing**
 - [Define testing methods]
 - [Define user personas]
 - [Define accessibility standards]
 - [Define UX metrics]
3. **Reliability Testing**
 - [Define availability targets]
 - [Define fault tolerance requirements]
 - [Define recovery procedures]
 - [Define stability metrics]
4. **Security Testing**
 - [Define security standards]
 - [Define penetration testing scope]
 - [Define security compliance requirements]
 - [Define vulnerability assessment methods]

1.4 Agile Testing Quadrants

1. **Q1: Unit & Component (Technology-Facing, Supporting Team)**
 - [Define automated checks]
 - [Define technical debt criteria]
 - [Define component quality metrics]
2. **Q2: Functional Tests (Business-Facing, Supporting Team)**
 - [Define story testing approach]
 - [Define acceptance test automation]
 - [Define business rule validation]
3. **Q3: User Experience (Business-Facing, Critique Product)**
 - [Define usability testing in sprints]
 - [Define UX feedback loops]
 - [Define beta testing strategy]
4. **Q4: Non-Functional Testing (Technology-Facing, Critique Product)**
 - [Define performance requirements]
 - [Define security testing cadence]
 - [Define scalability verification]

1.5 Automated Testing Infrastructure

1. **CI/CD Pipeline Integration**
 - [Define automation triggers]
 - [Define test types]
 - [Define quality gates]
 - [Define reporting requirements]
2. **Testing Environments**
 - Development: [Define setup]
 - Testing: [Define setup]
 - Staging: [Define setup]
 - Production: [Define deployment strategy]
3. **Platform Coverage**
 - Priority 1: [Define primary platforms]
 - Priority 2: [Define secondary platforms]
 - Priority 3: [Define edge cases]
 - Version Support: [Define supported versions]

1.6 Continuous Integration & Testing

1. Pipeline Integration

- [Define automation triggers]
- [Define test types]
- [Define quality gates]
- [Define reporting requirements]

1.7 Manual Testing Procedures

1. Exploratory Testing

- [Define schedule]
- [Define methodology]
- [Define roles]
- [Define outcomes]

2. Acceptance Testing

- [Define stakeholders]
- [Define criteria]
- [Define process]

3. Beta Testing

- [Define approach]
- [Define feedback collection]
- [Define success metrics]

1.8 Quality Metrics in Agile

1. Sprint-Level Metrics

- [Define velocity impact measures]
- [Define technical debt tracking]
- [Define defect density per story]

2. Release-Level Metrics

- [Define quality trends]
- [Define customer satisfaction metrics]
- [Define technical metrics]

1.9 Defect Management

1. Priority Levels

- P0: [Define criteria and SLA]
- P1: [Define criteria and SLA]
- P2: [Define criteria and SLA]
- P3: [Define criteria and SLA]

2. Bug Life Cycle

- [Define stages]
- [Define responsibilities]
- [Define timeframes]

3. Reporting Requirements

- [Define template]
- [Define required information]
- [Define documentation standards]

1.10 Defect Management in Agile

1. Sprint Defect Handling

- [Define in-sprint fix criteria]
- [Define defect prioritization]
- [Define technical debt conversion]

1.11 Release Readiness

1. Release Criteria

- [Define quality gates]
- [Define metrics]
- [Define approvals]

2. Release Decision Process

- [Define stakeholders]
- [Define review process]
- [Define approval chain]

3. Post-Release

- [Define monitoring]
- [Define metrics]
- [Define response plan]

1.12 Release Strategy

1. Continuous Delivery

- [Define feature toggles]
- [Define progressive rollouts]
- [Define monitoring strategy]

2. Release Decision

- [Define sprint-based quality gates]
- [Define MVP quality criteria]
- [Define stakeholder sign-off]

1.13 Continuous Improvement

1. Key Metrics

- [Define metrics]
- [Define collection method]
- [Define analysis approach]

2. Process Review

- [Define review frequency]
- [Define evaluation criteria]

- [Define update process]

3. **Team Development**

- [Define training]
- [Define documentation]
- [Define knowledge sharing]

1.14 Adaptation & Improvement

1. **Inspect and Adapt**

- [Define quality retrospectives]
- [Define process experiments]
- [Define success metrics]

2. **Knowledge Sharing**

- [Define quality guilds]
- [Define community of practice]
- [Define learning backlog]