

Quality in Agile Software Development

Emmanuel BRUNO

Université de Toulon, LIS UMR CNRS 7020

emmanuel.bruno@univ-tln.fr

2026-10-03

Table des matières

1	Introduction to Quality in Agile	1
2	Core Quality Concepts	1
2.1	Definition of Quality	1
2.2	Quality Dimensions	1
3	Quality Throughout the Agile Lifecycle ..	2
3.1	Planning Phase	2
3.2	Development Phase	2
3.3	Testing Phase	2
3.4	Release Phase	2
4	Quality Practices in Agile	2
4.1	Continuous Integration/Continuous Delivery (CI/CD)	2
4.2	Test Automation	2
4.3	Code Quality	2
4.4	Documentation	2
5	Quality Metrics and KPIs	2
5.1	Performance Metrics	2
5.2	Code Quality Metrics	2
5.3	Process Metrics	2
6	Quality Tools and Technologies	3
6.1	Testing Tools	3
6.2	Code Analysis Tools	3
6.3	Monitoring Tools	3
7	Quality Governance	3
7.1	Standards and Guidelines	3
7.2	Review Process	3
7.3	Quality Roles	3
8	Best Practices	3
8.1	Development Practices	3
8.2	Testing Practices	3

8.3	Review Practices	3
9	Common Quality Challenges	3
9.1	Technical Challenges	3
9.2	Process Challenges	3
9.3	Cultural Challenges	3
10	Quality Improvement	4
10.1	Continuous Improvement	4
10.2	Knowledge Management	4
10.3	Innovation	4

1 Introduction to Quality in Agile

Quality in agile software development is integrated throughout the development lifecycle rather than being a separate phase. This reference guide provides an overview of quality practices, methodologies, and best practices in agile contexts.

2 Core Quality Concepts

2.1 Definition of Quality

- Meeting user expectations and requirements
- Delivering value consistently
- Technical excellence
- Sustainable development practices
- Continuous improvement

2.2 Quality Dimensions

- Functional correctness
- Performance efficiency

- Compatibility
- Usability
- Reliability
- Security
- Maintainability
- Portability

3 Quality Throughout the Agile Lifecycle

3.1 Planning Phase

- Quality objectives definition
- Risk assessment
- Definition of Done (DoD)
- Acceptance criteria establishment

3.2 Development Phase

- Test-Driven Development (TDD)
- Pair programming
- Code reviews
- Continuous Integration

3.3 Testing Phase

- Automated testing
- Exploratory testing
- Performance testing
- Security testing

3.4 Release Phase

- Deployment automation
- Monitoring
- Feedback collection
- Post-deployment validation

4 Quality Practices in Agile

4.1 Continuous Integration/Continuous Delivery (CI/CD)

- Automated builds
- Test automation
- Deployment automation
- Quality gates

4.2 Test Automation

- Unit testing
- Integration testing
- End-to-end testing
- Performance testing
- Security testing

4.3 Code Quality

- Static code analysis
- Code reviews
- Technical debt management
- Refactoring

4.4 Documentation

- Living documentation
- API documentation
- User guides
- Technical documentation

5 Quality Metrics and KPIs

5.1 Performance Metrics

- Response time
- Throughput
- Resource utilization
- Scalability

5.2 Code Quality Metrics

- Code coverage
- Cyclomatic complexity
- Technical debt
- Code smells

5.3 Process Metrics

- Lead time
- Cycle time
- Deployment frequency
- Change failure rate

6 Quality Tools and Technologies

6.1 Testing Tools

- Unit testing frameworks
- Integration testing tools
- Load testing tools
- Security scanning tools

6.2 Code Analysis Tools

- Static analyzers
- Code coverage tools
- Dependency scanners
- Security analyzers

6.3 Monitoring Tools

- Application monitoring
- Infrastructure monitoring
- User monitoring
- Error tracking

7 Quality Governance

7.1 Standards and Guidelines

- Coding standards
- Architecture guidelines
- Security standards
- Performance requirements

7.2 Review Process

- Code review guidelines
- Architecture review
- Security review
- Performance review

7.3 Quality Roles

- Quality champions
- Code reviewers
- Test engineers
- Security experts

8 Best Practices

8.1 Development Practices

- Clean code principles
- SOLID principles
- Design patterns
- Security by design

8.2 Testing Practices

- Test pyramid
- Shift-left testing
- Risk-based testing
- Continuous testing

8.3 Review Practices

- Peer reviews
- Cross-functional reviews
- Regular audits
- Retrospectives

9 Common Quality Challenges

9.1 Technical Challenges

- Legacy code
- Technical debt
- Tool integration
- Environment consistency

9.2 Process Challenges

- Time constraints
- Resource limitations
- Knowledge gaps
- Communication issues

9.3 Cultural Challenges

- Quality ownership
- Resistance to change
- Collaboration barriers
- Skills development

10 Quality Improvement

10.1 Continuous Improvement

- Regular assessments
- Feedback loops
- Process refinement
- Tool optimization

10.2 Knowledge Management

- Training programs
- Documentation
- Knowledge sharing
- Best practices

10.3 Innovation

- New tools adoption
- Process automation
- Quality frameworks
- Emerging practices