Quality Metrics

Quality Metrics

Generated by adpa-enterprise-framework-automation v3.2.0
Category: quality-assurance
Generated: 2025-07-14T21:03:27.219Z
Description: Quality metrics and measurement criteria

Quality Metrics Framework

Project: ADPA - Advanced Document Processing & Automation Framework
Version: 3.2.0
Prepared by: Quality Assurance & Metrics Analyst
Date: July 2025

1. Quality Metrics Overview

Purpose and Objectives

- **Purpose:** To continuously measure, monitor, and improve the quality of the ADPA framework across its full lifecycle (development, testing, deployment, and production).
- **Objectives:**
  - Ensure compliance with enterprise, regulatory, and industry standards (BABOK v3, PMBOK 7, DMBOK 2.0, GDPR, SOX, etc.).
  - Deliver robust, performant, and secure automation for document and project management.
  - Maximize customer satisfaction and minimize production issues.

Metrics Framework and Methodology

- **Approach:**
  - Metrics are defined for process, product, defect, and customer quality.
  - Data is collected using automated tools (CI/CD, static analysis, test coverage tools, monitoring dashboards) and manual audits.
  - Thresholds are established for each metric, aligned with project and stakeholder goals.

- KPIs are reviewed in regular quality meetings, with continuous improvement actions documented and tracked.

Quality Goals and Targets

- **Process Compliance:** ≥ 95%
- **Test Coverage:** ≥ 90% of critical code paths
- **Defect Density:** ≤ 0.4/1000 LOC pre-release
- **User Acceptance Rate:** ≥ 98% for major features
- **System Availability:** ≥ 99.9% uptime in production

Stakeholder Expectations and Success Criteria

- **Regulatory Compliance:** Full adherence to listed standards
- **Enterprise-Readiness:** Demonstrated by successful integrations (Adobe, SharePoint, Confluence)
- **Performance:** Sub-second API response for 95% of calls
- **Security:** Zero critical vulnerabilities in production

2. Process Quality Metrics

Development Process Metrics

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
Code Review Effectiveness	% PRs with >1 reviewer; % PRs with actionable comments	≥ 95% PRs reviewed, ≥ 80% actionable	GitHub/GitLab, Code Review Audit	Weekly
Defect Injection Rate	# defects reported per development phase	< 2/feature/phase	Jira, GitHub Issues	Sprint
Process Compliance	% adherence to documented workflows (coding standards, branching, commit messages, security checks)	≥ 95%	CI/CD Audit, Manual Check	Monthly
Development Velocity	# story points / sprint	Trend improving or stable	Jira, Azure DevOps	Sprint

Testing Process Metrics

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
Test Execution Effectiveness	% of planned tests executed per cycle	≥ 98%	Jest, CI/CD Reports	Sprint
Test Coverage	% code coverage (statements, branches, functions)	≥ 90% critical, ≥ 80% overall	Jest, coverage tools	Sprint
Defect Detection Efficiency	% defects found in testing vs. production	≥ 95% pre-prod	Jira, TestRail, CI/CD	Release
Test Automation Coverage	% of regression test suite automated	≥ 90%	Jest, Automation Logs	Sprint

3. Product Quality Metrics

Functional Quality

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
Requirements Coverage	% of requirements with corresponding test cases	100%	Requirements Traceability Matrix	Release
Feature Completeness	% of committed features delivered per release	≥ 98%	Jira, Release Notes	Release
User Story Acceptance Rate	% of stories accepted by PO/BA	≥ 98%	Jira, UAT Reports	Sprint
Business Rule Compliance	% of business rules implemented and validated	100%	Test Cases, PMBOK/BABOK Mapping	Release

Technical Quality

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
Code Quality	# ESLint/Prettier/TypeScript warnings/errors	0 errors, < 5 warnings	ESLint, Prettier, tsc	CI/CD
Performance	API p95 response time under load	≤ 800ms (95th percentile)	k6, JMeter, NewRelic	Release
Security Vulnerabilities	# critical/high vulnerabilities (prod)	0 critical/high	Snyk, npm audit, OWASP ZAP	Release
Reliability/Availability	% uptime (prod), # outages > 1 min	≥ 99.9%, 0 outages	UptimeRobot, Azure Monitor	Monthly

4. Defect Quality Metrics

Defect Discovery

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
Defect Detection Rate	# defects found per phase (dev, QA, UAT, prod)	> 90% pre-prod	Jira, TestRail	Sprint
Defect Density	Defects per 1000 LOC	≤ 0.4/1000 LOC	SonarQube, Jira	Release
Defect Severity Distribution	% severe vs. minor defects	< 2% severe in total	Jira, Bug Report Analysis	Release
Defect Aging/Resolution	Avg. time to resolve critical, high, medium issues	< 2 days (critical), < 5 days (high)	Jira, SLA Reports	Weekly

Defect Prevention

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
Root Cause Analysis Coverage	% critical defects with RCA documented	100%	Jira, RCA logs	Sprint
Defect Prevention Effectiveness	% preventive actions closed on time	≥ 90%	Jira, Action Tracker	Monthly
Process Improvement Impact	% reduction in recurring defects	≥ 20%/quarter	Defect Trends, RCA	Quarterly
Recurring Defect Patterns	# of repeated root causes per period	Declining trend	Jira, RCA	Quarterly

5. Customer Quality Metrics

User Satisfaction

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
UAT Pass Rate	% of test cases passed during User Acceptance	≥ 98%	UAT Reports, Jira	Release
Customer Satisfaction (CSAT)	Survey score (1-5 scale)	≥ 4.5/5	Post-release Survey	Quarterly
System Usability Score (SUS)	Standardized usability survey	≥ 85/100	SUS Survey	Quarterly
User Experience (UX) Issues	# major UX issues reported	< 3 per major release	Feedback, Issue Tracker	Release

Production Quality

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
System Availability/Uptime	% uptime (prod)	≥ 99.9%	Azure Monitor, UptimeRobot	Monthly

Metric	Measurement Criteria	Threshold/Target	Data Source/Tools	Reporting
Performance Under Load	Max concurrent users supported @ SLA	≥ 1000, ≤ 1% error rate	Load Testing, Azure	Quarterly
Production Defect Rate	# defects reported by users per 1000 users/month	< 0.2	Helpdesk, Jira	Monthly
Mean Time to Recovery (MTTR)	Avg. time to recover from critical incident	< 30 min	Incident Logs, PagerDuty	Monthly

6. Quality Reporting and Dashboards

Metrics Collection Methods

- **Automated Data Collection:**
  - CI/CD pipeline integrations (Jest, ESLint, Snyk, SonarQube)
  - API monitoring (NewRelic, Azure Monitor)
  - Code and test coverage reports
- **Manual Measurement:**
  - Process compliance audits
  - User acceptance and customer satisfaction surveys
- **Tool Integration:**
  - JIRA/Azure DevOps for defect and story tracking
  - APIs for test automation tools (Jest, TestRail)
- **Data Validation:**
  - Cross-check automated data with periodic manual audits
  - Use API/webhook integration for real-time accuracy

Reporting Framework

- **Dashboards:**
  - Real-time dashboards (Grafana, PowerBI, Azure Dashboards) for key metrics
  - Drill-down views for code quality, defects, test coverage, and production health
- **Reporting Frequency:**
  - **Daily:** Automated build, test, and deployment status
  - **Sprint:** Process and product quality metrics
  - **Monthly:** Defect metrics, customer quality, compliance
  - **Quarterly:** Trend and improvement analysis
- **Distribution:**
  - Shared with all stakeholders via email, Slack/Teams, and project Confluence
  - Executive summaries for leadership
- **Trend Analysis:**
  - Visualization of metric trends over time for early warning and improvement tracking

- **Action Item Tracking:**
    - Integrated with Jira/DevOps for assignment and closure of quality actions

---

7. Quality Improvement Actions

Threshold Management

- **Quality Gates:**
    - Enforced in CI/CD (block release if thresholds not met)
    - E.g., block deploy if test coverage < 90% or critical vulnerabilities detected
- **Escalation Procedures:**
    - Immediate notification to QA lead/PM for threshold breaches
    - Incident review and mandatory action plan for repeated breaches
- **Corrective Action Triggers:**
    - Automated triggers for defect spikes, performance degradation, or compliance failures
    - Formal RCA and improvement plan required for critical/recurring issues
- **Continuous Improvement:**
    - Retrospectives after each release/sprint
    - KPIs reviewed and revised quarterly

Metrics Analysis

- **Trend Identification:**
    - Automated dashboard analytics for increasing/decreasing trends (e.g., defect density, velocity)
    - Root cause investigation for negative trends
- **Improvement Opportunities:**
    - Prioritized backlog of process or product improvements based on metric analysis
    - Action plans assigned and tracked to completion
- **Success Measurement:**
    - Post-action measurement to confirm improvement effectiveness
    - Lessons learned documented and shared with team

Appendix: Example Quality Dashboard Widgets

- **CI/CD Compliance:** Code review, lint, test pass rates
- **Coverage Map:** Unit, integration, and E2E test coverage by module
- **Defect Heatmap:** Defect count and severity by release/module
- **Uptime SLAs:** Rolling 30-day system availability
- **Customer Feedback:** CSAT and NPS trends

This framework ensures that ADPA meets its enterprise quality, compliance, and customer value commitments—and provides actionable, transparent metrics for ongoing improvement.