



Read this document and prepare me a one which I can use to recall any time

Here's a **concise, structured recall guide** covering the most important concepts, definitions, examples, and templates from your Agile-QA.docx document. This summary is ideal for quick review, team onboarding, or as a single-page personal reference.

Agile QA Quick Reference Guide

1. Key Testing Concepts

Term	Description	Analogy	Example
Test Suite	Group of related test cases for a feature	Recipe book	User Authentication Suite
Test Scenario	High-level situation/goal to be tested	"Bake a cake" goal	Login with valid credentials
Test Case	Step-by-step instructions and expected outcome	Detailed recipe	Steps to login & expect dashboard
Test Script	Manual or automated execution for a test case	Checklist/robot	Selenium script for login test

2. Organizing Test Suites (Example Table)

Level	Description	Example
Test Suite	Collection of test cases for a feature/module	"User Management Test Suite"
Test Scenario	High-level goal or user story	"Create new user"
Test Case	Step-by-step test & expected result	Create, edit, delete user workflows
Test Script	Actual execution (manual or automated)	Script for user creation / deletion

3. TDD vs. BDD

Feature	TDD	BDD
Focus	Code functionality	Application behavior
Language	Programming code	Plain English (Given-When-Then)

Feature	TDD	BDD
Who writes	Developers	Developers, QA, Business
Tools	JUnit, TestNG, pytest	Cucumber, SpecFlow, Behave
Output	Unit tests	Executable feature specs

- **TDD:** Write tests before code (developer-driven, code correctness).
- **BDD:** Write scenarios in natural language before implementation (collaborative, user-driven).

4. Quality Strategy & RACI (Sample Matrix)

Activity	Responsible	Accountable	Consulted	Informed
Test case creation	QA	QA Lead	Devs, BA	PM
Regression test exec.	QA	QA Lead	Devs	PM
Security testing	Security Eng	Tech Lead	QA	PM
Definition of Done	PM	Product Owner	QA, Dev	All

- **Definition of Done (DoD):** Agreed criteria for when a feature is “done” (e.g., peer review passed, test cases documented, no critical bugs remain).

5. QA Metrics Dashboard (Template)

Metric	Target	Current	Status	Trend	Notes
Test Coverage (%)	80%	76%	⚠	↗	Improving, focus next sprint
Automation Coverage (%)	60%	54%	⚠	→	Add new tests
Defect Leakage Rate (%)	0%	2%	✖	↗	2 bugs post-release
Avg. Defect Resolution	<2 days	3.1	⚠	↘	Review assignment process
Critical Defects (Open)	0	2	✖	→	Urgent fix needed

6. Risk Assessment Matrix (Template)

Feature/Module	Business Impact	Likelihood	Risk	Planned Tests	Owner	Status
Muse-VR Sync	High	High	High	E2E, regression, pairing	QA Lead	Testing
SNR Data Logging	High	Medium	High	Data validation, error cases	QA, Dev	Design
Admin Panel Update	Low	Low	Low	Smoke/regression	QA	Planned

7. Best Practices

- Organize all testing work in suites and scenarios for clarity.
- Use TDD for unit/integration, BDD for collaborative/behavior-driven features.
- Maintain and update your RACI and Definition of Done as the team evolves.
- Track and act on metrics; don't just report numbers—use them for focus and improvement.
- Prioritize testing by business and technical risk (risk-based testing).
- Keep your strategy “alive”—review and adapt each sprint.

Recall Tip:

Use this guide whenever you need to:

- Structure or review your QA process,
- Explain test concepts to teammates,
- Fill out dashboards or risk logs,
- Get clarity on roles, metrics, or quality strategies for Agile projects.

Pro Tip:

Keep this guide saved in your personal drive or a team wiki for fast reference mid-project or when welcoming new QA team members!

*
**

1. Agile-QA.docx