Test Driven Agile

Based around the Agile Testing Matrix by Brian Marick

Test Driven Development (TDD)

A - Write the test first!

B - Run the test and watch it fail 1st time. This tests the 'test'.

C - Write code until the test passes. Refactor to improve quality.

D - Automate for reusability, and create your incremental test system.

FAIL   PASS

TESTS

Q2 — User Facing Story and Feature Acceptance Tests

Test Case Input → Black Box → Test Case Ouput

Current Sprint

PO   Team

A - Written in the language of the business domain.

B - Developed jointly by the team and Product Owner.

C - Black Box testing verifies user satisfaction not coding methods.

D - Created during the Sprint in which the Story is implemented.

E - Test effectiveness is owned by the Product Owner.

Q3 — System Acceptance Tests

Aggregate   Explore   Use   Access

AND   OR   NOT

A - Determines if the aggregate behavior of the system meets usability and functionality requirements

B - Includes exploratory, user acceptance, accessibility testing, and more.

C - Largely manual, using the system in actual or simulated deployment covering various usage scenarios.

Program Level Feature Acceptance Tests

P   F
Program Increment

	High Value Low Effort	High Value High Effort
DO THIS FIRST		
	Low Value Low Effort	Low Value High Effort

Value ↑   Effort →

Inspect & Adapt

A - Program Increment (PI) testing. An aggregation of sprint values (features) worthy of release.

B - Program Increments chosen on the basis of Weighted Shortest Job First (WSJF) for best value.

C - Hold a Program Retrospective – We follow an Inspect & Adapt (IA) process at the end of each PI.

Agile Testing Matrix

Business Facing

Automated And Manual   Manual

Supporting Development   *Critiquing the Solution*

STEP ONE *Test First Code Later*

System Functionality
Story Acceptance Tests
Feature Acceptance Tests

APPROVED *Validate*

System Acceptance
Scenario Tests
Exploratory Tests
User Acceptance Tests
Alpha & Beta Tests

Q2   Q3

Q1   **STEP ONE** *Test First Code Later*

System Elements
Unit Tests
Component Tests

Q4 *Continuous Testing*

System Quality
Performance and Load Tests
Security & Penetration Tests
Other NFRs

Automated   Tools

Technology Facing

Acceptance Test Driven Development (ATDD)

YES   NO   Maybe

Process → Process

TESTS

A - Work Meets Product Owner expectations.

B - Ensure requirements are clear and unambiguous.

C - Product Owner provides business examples.

D - Document the acceptance criteria.

E - Creates tests to 'test' the acceptance criteria is met. Automate if possible.

Q1 — Unit Tests and Component Tests

Unit   Component

Incremental Testing
Passed
Test 4
Test 3
Test 2
Test 1
New Code

TESTS

A - Unit tests are the lowest atomic level of testing. Developer created to ensure code works as intended.

B - Tests with a higher level of integration. Checks developer code integrates with system components.

C - Each sprint creates an incremental test platform for all new code.

D - Automation provides low cost persistence of tests to ensure stability as code added.

Automating Acceptance Testing

404

Tests ↑   Stories →

User Interface
Application Logic
SDLC Methods

A - Manual testing is time consuming, and prone to errors.

B - The Faster you go (Sprints)...

C - The Faster they grow (Tests)...

D - The Slower you go (Velocity)...

E - Test business logic, not presentation layers, or implementation methods.

Q4 — System Quality Tests

Authorized Personnel Only

hacker inside

A - Security Testing

B - Penetration Testing

C - Performance Testing

D - Load Testing

E - Test continuously during every sprint using automation tools to ensure we don't violate any Non Functional Requirements (NFRs).