Testing Artifacts (Based On My Previous Project)

Prepared by: **Omkar Ramesh Kore**

Overview: This document contains example artifacts commonly used in Agile product development and testing. Use them as templates: Product Backlog, Epics (and decomposition), User Stories, Sprint Plan, Test Plan for a user story, Test Cases, and a Requirements Traceability Matrix (RTM).

Table of Contents

1. Product Backlog

Who creates: Product Owner (PO) owns and creates the Product Backlog. Business Analysts (BAs), Stakeholders, and Subject Matter Experts contribute items. The Development Team and QA provide estimates and technical feedback.

Correct format: A product backlog can be maintained in JIRA, Azure DevOps, or a spreadsheet. Each backlog item (PBI) should include: ID, Title, Description, Priority, Estimate (Story Points), Acceptance Criteria, Epic Link, and Current Status.

How it's defined (guidelines): Use INVEST for writing backlog items - Independent, Negotiable, Valuable, Estimable, Small, Testable. Include clear Acceptance Criteria and Definition of Done (DoD). Prioritize using business value and risk.

Example backlog items (Appointment Scheduling):

ID	Title	Description	Priority	Estimate	Acceptance Criteria (brief)	Status

PB-001	Patient Search	Search patient by name/ID/date of birth	High	3	Search returns matching patients; autocomplete;	Open
					no PII leakage	
PB-002	Book Appointment	Patient can book appointment with clinician and time slot	High	8	Pick clinician, date/time; confirmation sent; slot reserved	Open
PB-003	Cancel Appointment	Patient can cancel within policy window	Medium	3	Cancellation updates schedule, notifies clinician and patient	Open
PB-004	Reschedule Appointment	Patient can reschedule to available slots	Medium	5	Reschedule updates slot availability and notifications sent	Open
PB-005	View Provider Availability	Show clinician availability calendar	High	5	Shows correct free/busy slots; respects working hours	Open
PB-006	Send Reminders	Automated SMS/email reminders 24h before appointment	Low	3	Reminders delivered; retry on failure logged	Open
PB-007	Overbooking Protection	Prevent double-booking for same clinician	High	5	System blocks conflicting booking attempts	Open
PB-008	Audit Trail	Record booking/cancel/reschedule actions for compliance	Medium	3	Audit entries created with user and timestamp	Open

2. Epic Document

Purpose: Epics are large bodies of work that can be broken down into smaller user stories or features. Who creates: Product Owner in collaboration with Stakeholders and Architecture/Tech Lead.

How to divide into smaller parts: Start with the highest-value epic, identify capabilities, then split into features, then user stories. Use vertical-slicing—each story should deliver user-visible value.

Example Epic: AP-EPIC-01 — Appointment Management

Description: End-to-end functionality to allow patients and staff to search, book, reschedule, cancel, and manage appointments.

Epic ID	Sub-Epic/Feature	Brief Description
AP-EPIC-01	Booking	Allow patients and staff to book appointments
AP-EPIC-01	Rescheduling	Enable changing appointment to different slot
AP-EPIC-01	Cancellation	Ability to cancel with appropriate rules
AP-EPIC-01	Notifications	Email/SMS reminders and confirmations
AP-EPIC-01	Availability	Provider calendar and free/busy logic

3. User Story Document

User Story Format: As a <role>, I want <goal>, so that <benefit>.
Each story should have Acceptance Criteria (Given-When-Then style) and estimates.

Who creates: Product Owner + BA write stories. Team discusses and refines during Backlog Grooming/Refinement.

Selected User Stories (detailed):

US-002: Book Appointment (detailed)

As a patient, I want to book an appointment with a clinician at an available time slot so that I can receive care.

Priority: High | Story Points: 8 Acceptance Criteria (AC):

AC1: Given the patient is authenticated, when they search for a clinician and choose an available slot, then the system reserves the slot and shows a confirmation.

AC2: Given the slot is reserved for the patient, when another user tries to book the same slot, then the slot is shown as unavailable.
AC3: Given an appointment is booked, when booking completes, then the patient receives a confirmation email/SMS with appointment details.

AC4: System updates provider calendar and audit trail.

US-004: Reschedule Appointment (detailed)

As a patient, I want to reschedule my appointment to a new available slot so that I can change my plan.

Priority: Medium | Story Points: 5 Acceptance Criteria (AC):

AC1: Given a booked appointment, when patient selects a new available slot, the system updates appointment and notifies clinician and patient.

AC2: Previous slot is freed for others.

US-003: Cancel Appointment (detailed)

As a patient, I want to cancel my appointment within the allowable window so that I can free the slot. Priority:

Medium | Story Points: 3

Acceptance Criteria (AC):

AC1: Cancellation within policy window is allowed and notifies clinician and patient.

AC2: Cancellation outside the window prompts a fee or different flow (business rule).

4. Sprint Document (Example Sprint - 2 weeks)

Who creates: Scrum Master prepares the Sprint plan with input from PO and Development Team.

Sprint Name: Sprint 1 — Appointment MVP

Start Date: 2025-10-06 | End Date: 2025-10-19

Sprint Goal: Deliver core appointment booking flow (search, book, confirmation) and basic notifications.

Team: 1 PO, 1 SM, 3 Developers, 2 QA

Sprint Backlog (stories assigned): US-002 (Book Appointment) — 8pts, US-005 (View Provider Availability) — 5pts, US-001 (Patient Search) — 3pts

Capacity planning: Total dev capacity 80 hours. QA allocation 40 hours. Leave buffer for bugs and review.

Sprint Ceremonies: Daily Standup (15m), Sprint Planning (2h), Backlog Refinement (1h mid-sprint), Demo/Review (1h), Sprint Retrospective (1h)

Definition of Done (DoD) for sprint stories: Code implemented, unit tested, peer reviewed, deployed to Staging, QA-tested (pass), Documentation updated, Release notes prepared.

5. Test Plan — US-002: Book Appointment

Test Plan ID: TP-US-002

Prepared by: QA Lead

Objective: Verify that an authenticated patient can successfully book an appointment with a clinician and receive confirmation; ensure slot reservation and backend updates.

Scope:

- In scope: UI booking flow, provider availability check, slot reservation, confirmation delivery (email/SMS), audit trail, concurrency handling.
- Out of scope: Billing/Payment (handled in separate epic), Third-party notification provider failures beyond retry.

Test Approach:

- Functional testing (positive and negative scenarios)
- Integration testing (calendar service, notification service)
- Concurrency/stress testing for booking the same slot
- Regression testing to ensure no impact on existing flows

Test Environment:

- Staging environment replicating production (DB snapshot, provider calendar mock, notification test switch).
- Test Accounts: patient_test_01, clinician_test_01

Test Data Strategy:

- Use anonymized production-like test data; create dedicated test patients and clinicians; seed provider availability slots for test window.

Roles & Responsibilities:

Role	Name/Team	Responsibility
QA Lead	QA Team	Prepare test plan, RTM, oversee execution
Tester	QA Team	Design and execute test cases, log defects

Developer	Dev Team	Fix defects, implement features
PO	Product	Clarify acceptance criteria, sign-off

Entry Criteria:

- Feature deployed to Staging; Smoke tests pass; Test data seeded; Acceptance Criteria finalized.

Exit Criteria:

- All critical and high defects fixed and verified; Test cases executed with pass rate >= 95% for this story; PO signoff.

Risks & Mitigations:

- Race conditions on slot reservation: add optimistic locking; include concurrency tests.

- Notification service downtime: use test stub and validate retry logic.

6. Test Cases — US-002: Book Appointment

Test cases are stored in a test management tool (e.g., TestRail, Zephyr) or spreadsheet. Each test case includes: ID, Title, Preconditions, Steps, Expected Result, Priority, Type, Automated (Y/N), Trace to Story/AC.

TC ID	Title	Precondition	Test Steps	Expected Result	Priority	Type	Automated?
TCAPPT-001	Book Appointment - Happy Path	Patient logged in; clinician has available slot	1. Login as patient 2. Search clinician 3. Select available slot 4. Confirm booking	Booking confirmed; slot reserved; confirmation message shown; email/SMS sent	High	Functional	Yes
TCAPPT-002	Book Appointment - Slot Taken During Booking	Patient A and Patient B attempt same slot	1. Patient A reserves slot 2. Patient B attempts to book same slot	Patient B receives 'slot unavailable' message; no doublebooking	High	Concurrency	No

TCAPPT-003	Book Appointment - Invalid Date/Time	Patient picks a past date or outside working hours	1. Choose past date or invalid time 2. Attempt to book	System rejects with validation error	Medium	Negative	Yes
TCAPPT-004	Booking Confirmation Content	After booking completes	1. Complete booking 2. Check notification content (email/SMS)	Confirmation contains patient name, clinician, date/time, instructions, UID	Medium	Integration	No
TCAPPT-005	Audit Trail Entry Created	After booking completes	1. Complete booking 2. Inspect audit log record for appointment	Audit record exists with user, timestamp, action	Low	Security/Compliance	No
TCAPPT-006	UI - Booking Form Validation	Booking form fields mandatory check	1. Leave required fields empty 2. Try to submit	Form highlights fields and shows validation messages	Medium	UI/UX	Yes

7. Requirements Traceability Matrix (RTM)

Purpose: Ensure every acceptance criterion / user story is covered by test cases and link defects back to requirements.

Req/Story ID	Description	Acceptance Criteria (key)	Test Case IDs	Test Execution Status	Comments
US-002	Book Appointment	AC1: Reserve slot; AC2: Prevent doublebooking; AC3: Send confirmation	TC-APPT-001, TCAPPT-002, TC-APPT004	Not Executed	

US-003	Cancel Appointment	AC1: Cancel within window; notify	TC-APPT-C001	Not Executed	To be created
US-004	Reschedule Appointment	AC1: Change slot updates calendar and notify	TC-APPT-R001	Not Executed	To be created

8. Step-by-step Testing Flow (summary for sharing)

1) Product Backlog creation by PO — capture PBIs with clear acceptance criteria.
2) Epic decomposition — break large epics into features and user stories.
3) Story refinement — team estimates and defines DoD; QA prepares acceptance tests.
4) Sprint planning — select stories for sprint, agree on sprint goal and capacity.
5) Test Planning — QA prepares test plan for selected stories (scope, environments, approach).
6) Test Case Design — QA writes test cases mapped to acceptance criteria (positive/negative/concurrency/regression).
7) Test Execution — run test cases in Staging, log defects, retest fixes.
8) RTM & Reporting — update RTM, share test results with PO, get acceptance and sign-off.
9) Demo & Release — feature demoed, deployed to production following release checks.
10) Retrospective — capture improvements for next sprint.