# Entry Criteria, Exit Criteria and Acceptance Criteria

**# Source**: **Chatgot (GPT-5-Standard) at [8/17/2025]**

Great 👌 — these three terms (**Entry Criteria, Exit Criteria, Acceptance Criteria**) often sound similar but mean different things in **software testing**. Let’s break them down clearly:

## 1️⃣ Entry Criteria

### 📌 Definition

**# Source**: **Chatgot (GPT-5-Standard) at [8/17/2025]**

The **preconditions** that must be met before testing can begin.  
 It ensures that the test team doesn’t start testing until the environment, code, and resources are ready.

### ✅ Examples

**# Source**: **Chatgot (GPT-5-Standard) at [8/17/2025]**

* Requirements are signed off.
* Test environment is set up.
* Test data is prepared.
* Test cases are reviewed and approved.
* Build is deployed successfully.

👉 **Simple Example:** Before functional testing of a “Login Feature,” the **entry criteria** could be:

* Login page is developed and unit-tested.
* Test environment is up and stable.
* Test cases for login are ready.

### Definition of Ready

**# Source**: **Chatgot (GPT-5-Standard) at [9/14/2025]**

**Definition of Ready (DoR)** in Agile is a **set of clear conditions or criteria** that a **User Story (or backlog item)** must meet **before the team can start working on it** in development or testing.

**📌 Key Points about DoR:**

* It’s an **entry criterion** for a sprint/iteration.
* Ensures that the backlog item is **well-prepared, clear, and feasible** for the team to work on.
* Helps avoid confusion, rework, and wasted effort once the sprint begins.
* Provides a **shared understanding** between Product Owner, testers, and developers about when a story is truly ready.

**✅ Examples of DoR:**

A User Story is **Ready** when:

* It is **clearly defined and understandable** by the team.
* Acceptance Criteria are written and testable.
* Dependencies are identified and manageable.
* Priority and business value are clear.
* The story has been estimated by the team.
* Required **test data, environments, or tools** are available.

🔑 **In short:**  
The **Definition of Ready** makes sure the team doesn’t start a story that’s vague, incomplete, or not testable. It reduces risk and sets the stage for smoother development and testing.

## 2️⃣ Exit Criteria

### 📌 Definition

**# Source**: **Chatgot (GPT-5-Standard) at [8/17/2025]**

**Exit criteria** (also called **test completion criteria**) are The **conditions that must be met before testing can be considered complete**.  
 It ensures that testing has achieved its goals before moving to the next phase (or releasing).

### ✅ Examples

**# Source**: **Chatgot (GPT-5-Standard) at [8/17/2025]**

* All planned test cases executed.
* 95% of test cases passed.
* No open Severity-1 or Severity-2 defects.
* Test coverage ≥ 90%.
* Test summary report prepared and signed off.

👉 **Simple Example:** For the “Login Feature,” the **exit criteria** could be:

* All login test cases executed and passed.
* No critical login defects remain open.
* Regression tests for login completed successfully.

### Definition of Done

**# Source**: **Chatgot (GPT-5-Standard) at [9/14/2025]**

**Definition of Done (DoD)** in Agile is a **shared agreement within the team** that clearly defines the conditions a product backlog item (e.g., a user story, feature, or task) must meet to be considered **complete and releasable**.

**📌 Key Points about DoD:**

* It’s an **exit criterion** for development and testing.
* Ensures **transparency**: everyone (team, Product Owner, stakeholders) has the same understanding of what “done” means.
* Provides **objective quality checks** before delivering an increment.
* Helps avoid misunderstandings like “it works on my machine” or “code is done but not tested.”

**✅ Examples of DoD:**

A user story is **Done** when:

* All **acceptance criteria** are met.
* Code is written, reviewed, and merged.
* Unit and integration tests are implemented and passed.
* Regression tests are executed with no critical defects.
* Documentation (if required) is updated.
* The increment is potentially shippable/releasable.

🔑 **In short:**  
The **Definition of Done** is a checklist that ensures the team delivers **high-quality, completed work** that could be released to users at the end of the iteration/sprint.

## 3️⃣ Acceptance Criteria

### 📌 Definition

**# Source**: **Chatgot (GPT-5-Standard) at [8/17/2025]**

The **conditions that a software product must meet to be accepted by the customer/business**.  
 It is defined for each **user story or requirement** and is business-focused (not just testing-focused).

* Used in **Agile**: part of user stories.
* Ensures the feature delivers **business value**.

### ✅ Examples

**# Source**: **Chatgot (GPT-5-Standard) at [8/17/2025]**

* *User Story:* “As a user, I want to log in to the website so that I can access my account.”
* *Acceptance Criteria:*
  + User can log in with valid username & password.
  + Error message is displayed for invalid credentials.
  + Account locks after 3 failed attempts.
  + Login works across Chrome, Firefox, and Safari.

👉 **Simple Example:** If a “Payment Feature” is developed, the **acceptance criteria** could be:

* Payment succeeds with valid credit card.
* Error is shown for expired card.
* Transaction receipt is generated.

### 📌 Ways to Write Acceptance Criteria for User Stories

#### 1. Checklist Style

**# Source**: **Chatgot (GPT-5-Standard) at [9/13/2025]**

👉 Simple **list of rules or conditions** that must be true for the story to be considered complete.

**Example (Password Reset):**

* System must send reset link to user’s email.
* Reset link must expire after 24 hours.
* New password must have at least 8 characters.
* Password must include at least one number and one letter.

**Example (Shopping Cart):**

* Users can add items to the cart.
* Users can remove items from the cart.
* Cart updates total price automatically.
* Cart prevents checkout when empty.

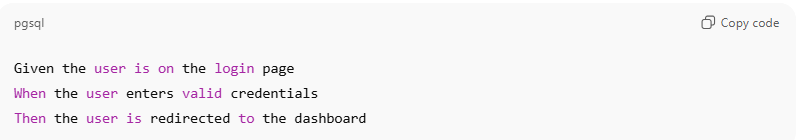
✅ **Best For:** Quick checks, QA teams, rule-heavy features.

#### 2. Scenario-Oriented (Given–When–Then / Gherkin Style)

**# Source**: **Chatgot (GPT-5-Standard) at [9/13/2025]**

👉 Behavior described as **scenarios**.

**Example (Login):**



Given the user is on the login page

When the user enters valid credentials

Then the user is redirected to the dashboard

✅ **Best For:** BDD, automation, stakeholder-friendly.

#### 3. Use Case-Oriented Style

**# Source**: **Chatgot (GPT-5-Standard) at [9/13/2025]**

👉 Acceptance criteria follow **main + alternative flows**.

**Example (Checkout):**

* **Main Flow:** User enters payment → payment succeeds → order confirmed.
* **Alternative Flow:** Payment fails → system shows retry option.

✅ **Best For:** Capturing happy path + edge cases.

#### 4. State/Condition-Oriented Style

**# Source**: **Chatgot (GPT-5-Standard) at [9/13/2025]**

👉 Focuses on **system states and conditions**.

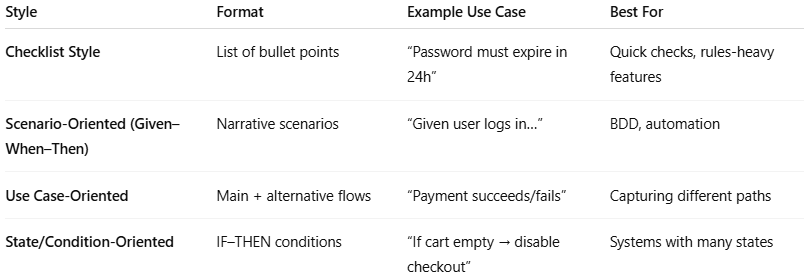
**Example (Cart):**

* If cart is empty → checkout disabled.
* If item out of stock → cannot be added.
* If payment succeeds → status = Confirmed.

✅ **Best For:** Systems with multiple state transitions.

#### 📊 Comparison Table

**# Source**: **Chatgot (GPT-5-Standard) at [9/13/2025]**



| **Style** | **Format** | **Example Use Case** | **Best For** |
| --- | --- | --- | --- |
| **Checklist Style** | List of bullet points | “Password must expire in 24h” | Quick checks, rules-heavy features |
| **Scenario-Oriented (Given–When–Then)** | Narrative scenarios | “Given user logs in…” | BDD, automation |
| **Use Case-Oriented** | Main + alternative flows | “Payment succeeds/fails” | Capturing different paths |
| **State/Condition-Oriented** | IF–THEN conditions | “If cart empty → disable checkout” | Systems with many states |

## 🔑 Key Difference Between Them

**# Source**: **Chatgot (GPT-5-Standard) at [8/17/2025]**



|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria Type** | **Focus** | **Who Defines It?** | **Example** |
| **Entry Criteria** | Preconditions to *start testing* | Test Manager / QA Lead | Test environment ready, test cases reviewed |
| **Exit Criteria** | Conditions to *end testing* | Test Manager / QA Lead | 95% test cases passed, no critical defects |
| **Acceptance Criteria** | Business rules for *accepting a feature* | Product Owner / Business Analyst | “Login must lock after 3 failed attempts” |

✅ **In short:**

* **Entry Criteria** = "When can we *start* testing?"
* **Exit Criteria** = "When can we *stop* testing?"
* **Acceptance Criteria** = "What conditions must be met for the *customer to accept* the feature?"