**Gherkin**:

Gherkin is a structured approach to write behavioral tests. As called **Behavior Driven Development**.

*Instead of testing little bits of code, behavioral tests seek to follow true user workflow. Such as signing in or applying for a refund. This means a focus on how users interact with our system.*

Gherkin is a framework for writing user stories because

* *it gives a consistent approach for reviewing all scenarios,*
* *defines the definition of Done and*
* *provides acceptance criteria.*

Benefits of using Gherkin:

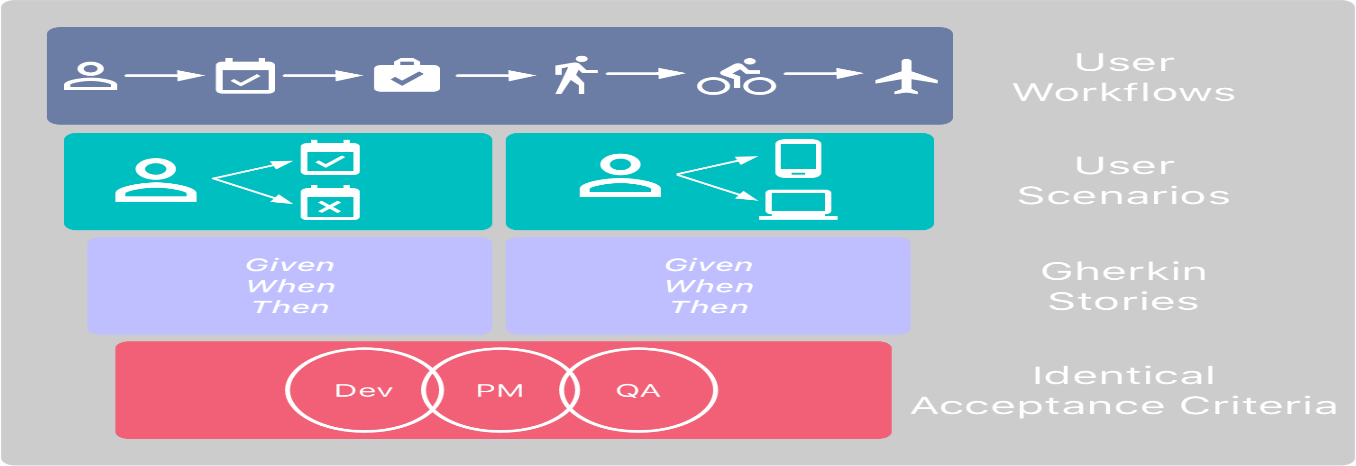
* *We can catch missing workflows before any work is started.*
* *Direct correlation to user workflows that we design and develop.*
* *We will know when the story is done because we have clear acceptance criteria. (Dev & PO).*
* *Consistent language across stories helps the teams focus delivering user value.*

**User Stories**: is a document description of a software features seen from the end user perspective. The user story describes what exactly the user wants the system to do. User stories ae organized in a backlog.

As a <type of user> I want <some goal> so that <some reason>.

**Acceptance criteria**: User stories must be accomplished by acceptance criteria. These are the conditions the product must satisfy to be accepted by a user, stakeholder or Product owner.

* Each User story mush have at least one acceptance criteria.



*Gherkins follows a very specific syntax*:

Scenarios: All the actions a user could take (including bad input)

GIVEN: sets the ***context.*** What page we are on and what state are we are in?

Is the user an admin? Signed-in?

WHEN: What **actions** the user is performing, what event occurred?

THEN: What should the system do in **response**? What is the **expected outcome**?

*Feature: Refund item*

*Scenario: Jeff returns a faulty microwave*

*Given Jeff has bought a microwave for $100*

*And he has a receipt*

*When he returns the microwave*

*Then Jeff should be refunded $100*

e.g. A familiar example for product managers:

Feature*: As a user I want to sign in so I can see my marketing campaigns*

Scenario*: User supplies correct user name and password*

Given *that I am on the sign-in page*

When *I enter my user name and password correctly*

And *click ‘Sign In’*

Then *I am taken to the dashboard*

Scenario*: User does NOT supply correct user name and password*

Given *that I am on the sign-in page*

When *I enter my user name and password incorrectly*

And *click ‘Sign In’*

Then *I see an error message ‘Sorry, incorrect user name or password.”*

**USER STORIES and ACCEPTANCE CRITERIA:**

User stories and Acceptance criteria – the main formats of documenting requirements.

A user story is a natural language description of feature. It’s usually accompanied by acceptance criteria.

Acceptance criteria are the conditions that a software product must meet to be accepted by a user, a customer, or other system. They are unique for each user story and define the feature gehaviour from end user’s perspective.

The high-quality software documentation could help avoid problems. User story and acceptance criteria as the main formats of document requirements.

Main purpose of Acceptance criteria:

* Defines the boundaries of user stories.
* Require the system to recognize invalid inputs (negative scenarios).
* Ensure everyone has a common understanding of requirements.
* Each acceptances criterion must be independently testable (clear pass or fail scenarios).

**Acceptance Criteria types and structures:**

1. Scenario-oriented (Given/When/Then)
2. Rule-oriented (checklist).

**1.Scenario-oriented acceptance criteria: (Known a Given/When/Then)**

Given some pre-condition (context)

When I do some actions (perform action)

Then I expected some result

This approach is inherited from Behavior-driven development (BDD).

Scenario – the name of the behavior that will be described

Given – the beginning stat of the scenario

When – specific action that the user makes

Then – the outcome of the action in ‘When’

And – used to continue any of three previous statements

e.g. 1

User story: *As a user, I want to able to recover the password to my account, so that I will be able to access my account in case I forgot the password.*

Scenario: Forgot password

Given: The user had navigated to the login page

When: The user selected forgot password option

And: Entered a valid email to receive a link for password recovery

Then: The system sent the link to the entered email

Given: The user received the link via the email

When: The user navigated through the link received in the email

Then: The system enables the user to set a new password

e.g. 2

Example 2

User story: As a user, I want to be able to request the cash from my account in ATM so that I will be able to receive the money from my account quickly and in different places.

Acceptance criteria 1:

Given: that the account is creditworthy

And: the card is valid

And: the dispenser contains cash

When: the customer requests the cash

Then: ensure the account is debited

And: ensure cash is dispensed

And: ensure the card is returned

Acceptance criteria 2:

Given: that the account is overdrawn

And: the card is valid

When: the customer requests the cash

Then: ensure the rejection message is displayed

And: ensure cash isn’t dispensed

**2.Rule-oriented acceptance criteria format:** (checklist)

In some cases, it’s difficult to fit acceptance criteria not the Given/When/Then structure.

* User stories that describe system level functionality needs other methods.
* Target audience for acceptance criteria doesn’t need details of test scenarios.
* Describing design and user experience constraints of a feature.

e.g.

User story: *As a user, I want to use a search field to type a city, name, or street, so that I could find matching hotel options.*

Basic search interface acceptance criteria

* *The search field is placed on the top bar*
* *Search starts once the user clicks “Search”*
* *The field contains a placeholder with a grey-colored text: “Where are you going?”*
* *The placeholder disappears once the user starts typing*
* *Search is performed if a user types in a city, hotel name, street, or all combined*
* *Search is in English, French, German, and Ukrainian*
* *The user can’t type more than 200 symbols*
* *The search doesn’t support special symbols (characters). If the user has typed a special symbol, show the warning message: “Search input cannot contain special symbols.”*