



FLAMA: test.

CALIDAD DEL SOFTWARE
Y TESTING



Ciclo de Vida del Testing:

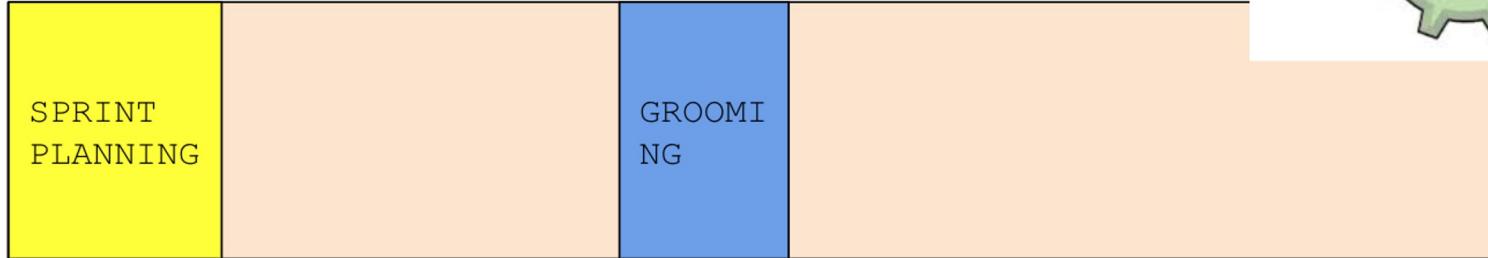
“Quality is never an accident. It is always the result of high intention, sincere effort, intelligent direction and skillfull execution. It represents the wise choice of many alternatives.”

EL PROCESO DE PRUEBAS: Software Development Life Cycle (SDLC) - Sprint

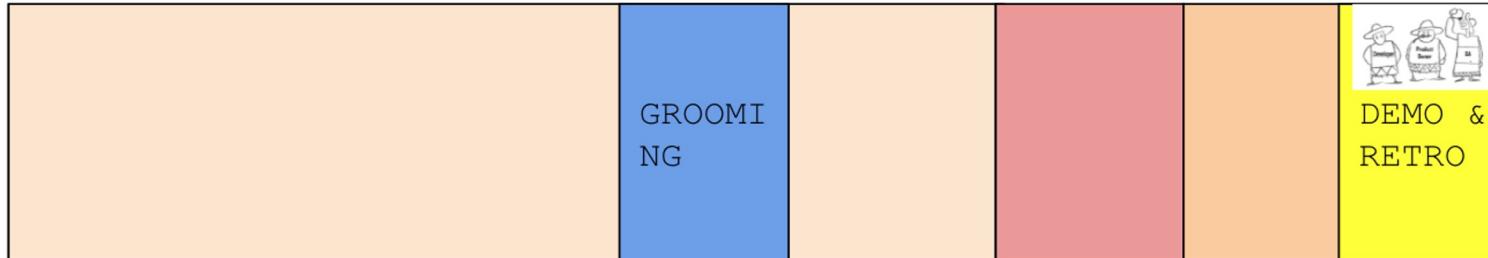


SDLC: 2w Sprint

Week1



Week2



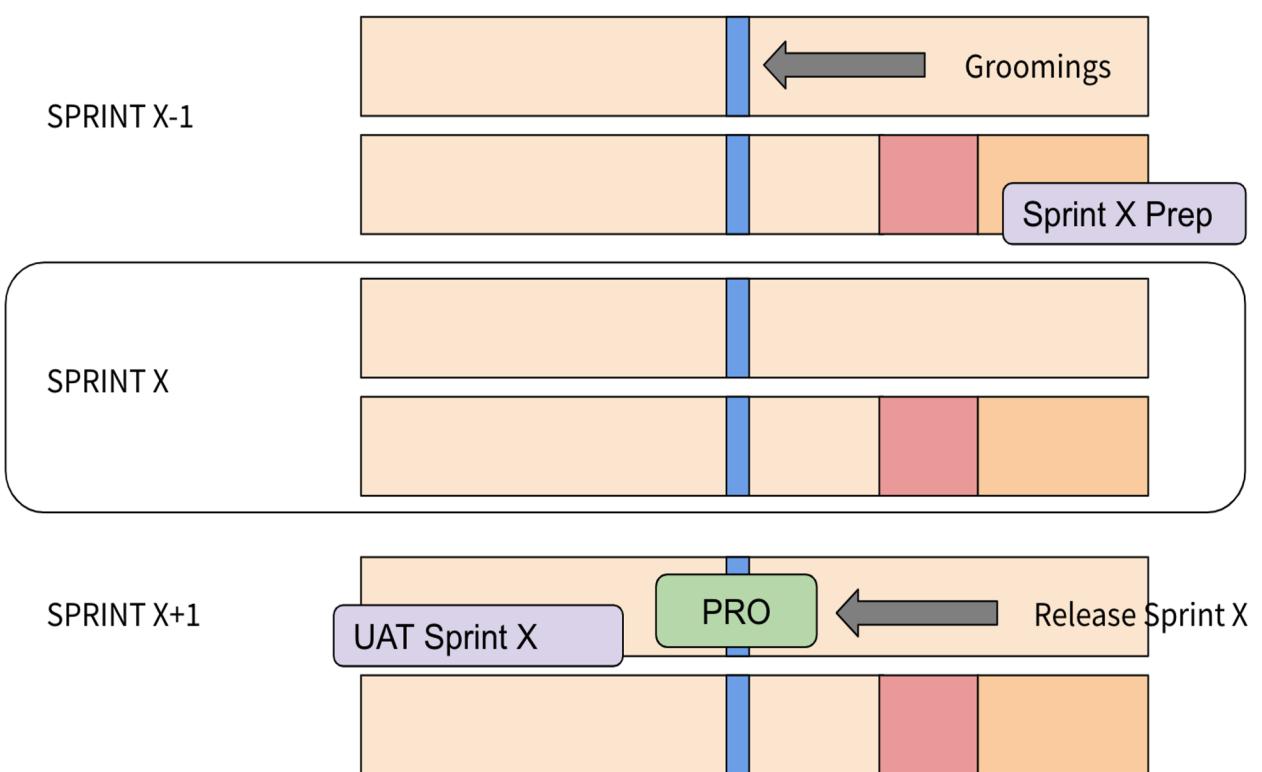


EL PROCESO DE PRUEBAS: Software

Development Life Cycle (SDLC) – Actividades

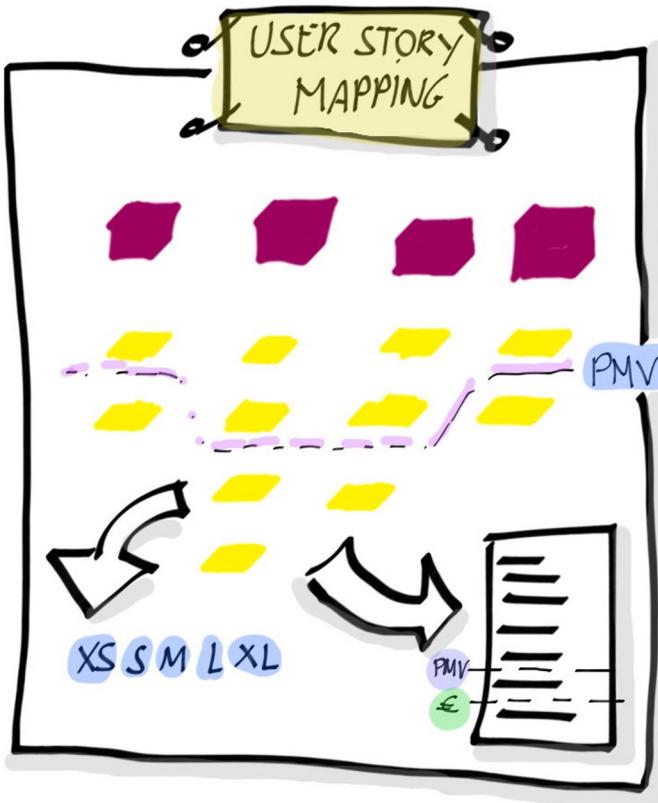
de QA

- Previo al Sprint
 - Refinamiento de Requisitos
 - Priorización de Defectos
 - Preparación del Sprint
- Sprint
 - Planificación
 - Inspección de Requisitos
 - *Pair-Testing*
 - Pruebas Exploratorias
 - Diseño de Pruebas
 - Ejecución de Pruebas
 - Verificación
 - Regresión
 - Automatización
- Despues del Sprint
 - QA Sign-Off
 - UAT - Validación
 - *Testing* en Producción
 - *Hot-Fixes*





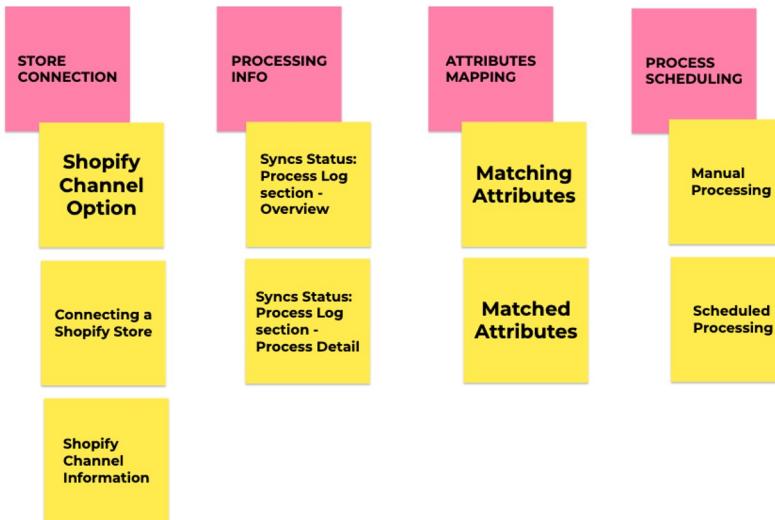
EL PROCESO DE PRUEBAS: Story Mapping





EL PROCESO DE PRUEBAS: Story Mapping

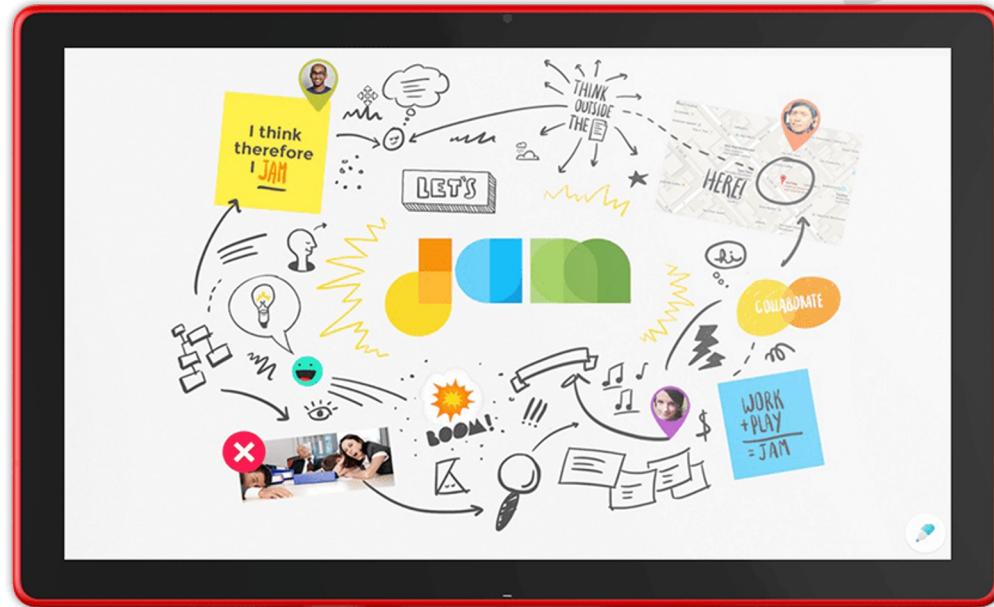
- Permite de una manera visual y colaborativa definir el mapa de funcionalidades de un producto.
- De un solo vistazo, tenemos el '*big picture*' de nuestro sistema en una sola instantánea. descompuesto de las ideas más generales a las más específicas.
- La dinámica del ordenar por prioridad, nos permite establecer las necesidades del negocio.
- Es muy sencillo agrupar las funcionalidades en *releases* para tener una hoja de ruta que seguir
- Este proceso no debe ser estático, si no que nuestro *mapping* puede evolucionar en función de las necesidades del proyecto.



EL PROCESO DE PRUEBAS: Story Mapping

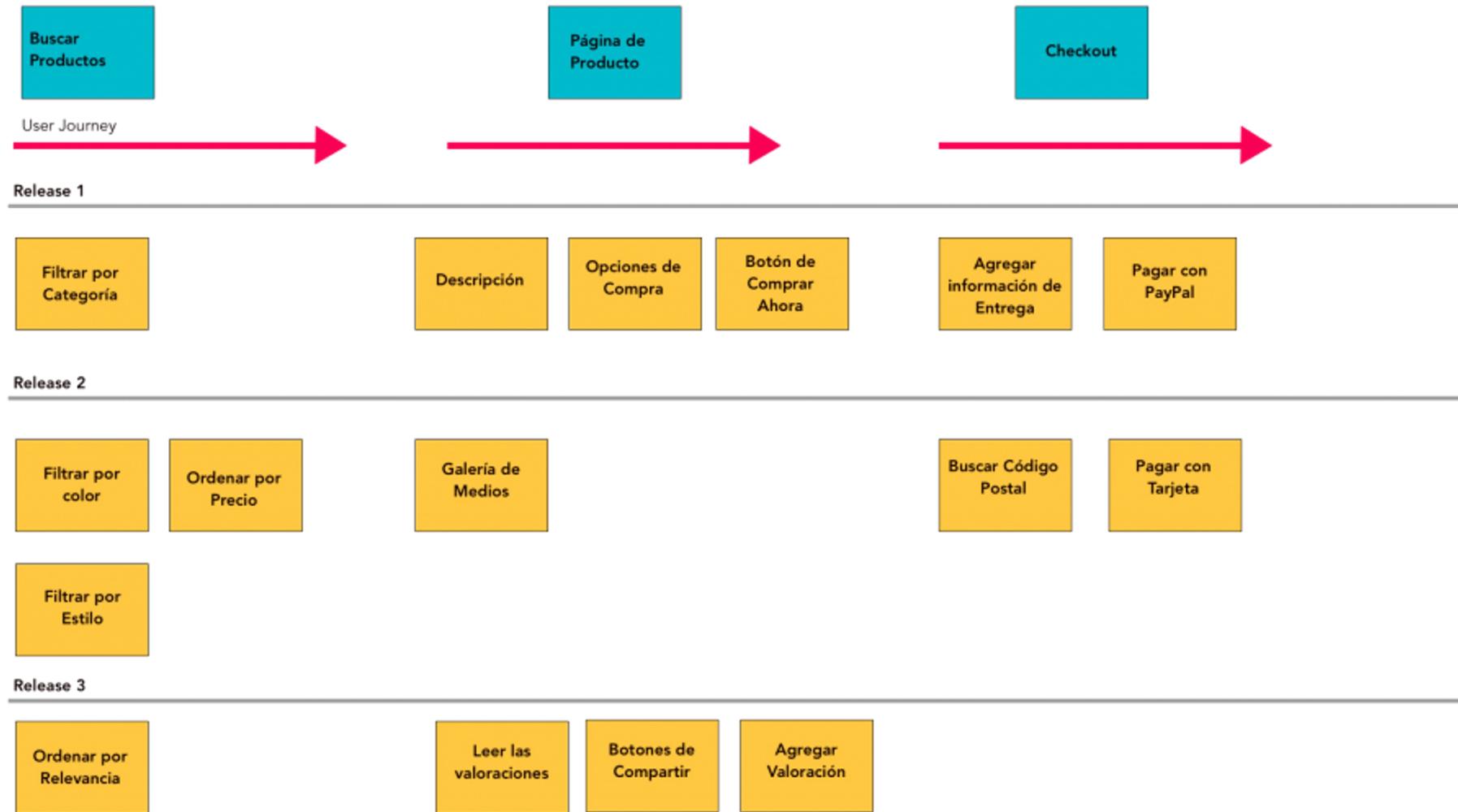


- En mi tienda online necesito los productos mostrados y visualizados de forma que pueda filtrarlos por criterios como precio, categoría, etc y ordenarlos.
- Además podré seleccionarlo para compra.
- Además necesito visualizar detalles de mi producto como descripción, fotos, opiniones.
- Quiero compartirlo en redes sociales.
- En el proceso de compra tendrá diferentes opciones de pago como paypal o tarjeta.
- En el resumen de pago quiero ver los datos de entrega, el importe total y un resumen de mis artículos.





EL PROCESO DE PRUEBAS: Story Mapping



EL PROCESO DE PRUEBAS: Historias de Usuario



Historia de Usuario

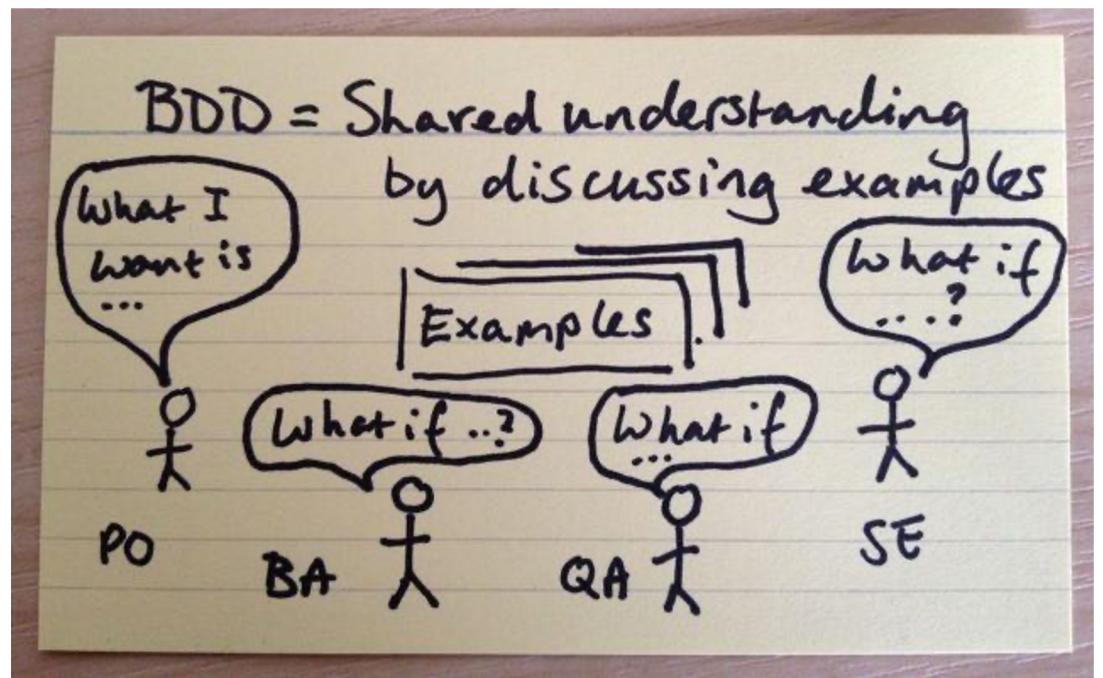
- **Independiente:** Tanto como sea posible, la dependencia entre historias harán la planificación, priorización y estimación mucho más difícil
- **Negociable:** Los detalles de la historia pueden ser cuestionados durante la planificación. Una historia con demasiado detalles puede a veces limitar las conversaciones
- **Valor** al cliente
- **Estimable:** Tiene que tener suficiente detalle para el equipo de desarrollo para poder estimar y priorizar la historia
- **Pequeña:** Cada historia debe representar un pequeño esfuerzo
- **Testeable:** El criterio de aceptación debe ser claro y no ambiguo, requisitos como “tiene que ser fácil de usar” o “no deber ir muy lento” no ayudan.

Keep
It
Simple



EL PROCESO DE PRUEBAS: Historias de Usuario

- Todas las definiciones BDD se escriben en un idioma común.
- El principal objetivo es que el equipo describa los detalles de cómo se debe comportar la aplicación a desarrollar, y de esta forma será comprensible por todos.
- Acceso a una comunicación más clara y con la mínima jerga tecnológica.
- Hace posible que la colaboración entre los equipos técnicos y no técnicos se ejecute con mayor eficiencia.





EL PROCESO DE PRUEBAS: Historias de Usuario

- Cada requisito debe convertirse en historias de usuario, defiendo ejemplos concretos.
- Cada ejemplo debe ser un escenario de un usuario en el sistema.
- Ser consciente de la necesidad de definir "la especificación del comportamiento de un usuario"

Este patrón también se utiliza en BDD para ayudar a la creación de historias de usuarios:

As a 'Como': Se especifica el tipo de usuario.

I want 'deseo': Las necesidades que tiene.

So that 'para que': Las características para cumplir el objetivo.

Un ejemplo práctico de historia de usuario sería:

- **Como** cliente interesado, **deseo** ponerme en contacto mediante el formulario, **para que** atiendan mis necesidades.



EL PROCESO DE PRUEBAS: Historias de Usuario

Para definir los casos BDD para una historia de usuario se deben definir bajo el patrón ‘*Given-When-Then*’, que se define como:

Given ‘dado’: Se especifica el escenario, las precondiciones.

When ‘cuando’: Las condiciones de las acciones que se van a ejecutar.

Then ‘entonces’: El resultado esperado, las validaciones a realizar.

Un ejemplo práctico sería:

Given: Dado que el usuario no ha introducido ningún dato en el formulario.

When: Cuando hace clic en el botón Enviar.

Then: Se deben mostrar los mensajes de validación apropiados.



EL PROCESO DE PRUEBAS: Historias de Usuario

Para definir los casos BDD para una historia de usuario se deben definir bajo el patrón ‘*Given-When-Then*’, que se define como:

Given ‘dado’: Se especifica el escenario, las precondiciones.

When ‘cuando’: Las condiciones de las acciones que se van a ejecutar.

Then ‘entonces’: El resultado esperado, las validaciones a realizar.

Un ejemplo práctico sería:

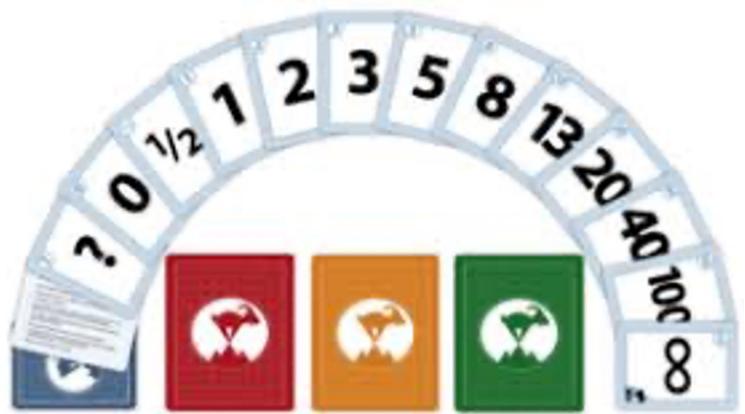
Given: Dado que el usuario no ha in

When: Cuando hace clic en el botón

Then: Se deben mostrar los mensaj

```
GoogleHomepage.feature
1 Feature: Google Homepage
2 This feature verifies the functionality of Google Homepage
3
4 Scenario: Check that main elements on Google Homepage are displayed
5 Given I launch Chrome browser
6 When I open Google Homepage
7 Then I verify that the page displays search text box
8 And the page displays Google Search button
9 And the page displays I'm Feeling Lucky button
```

EL PROCESO DE PRUEBAS: Historias de Usuario



1. Customer reads story.



Development team asks questions

2. Team estimates.
This includes testing.



Discussion ...

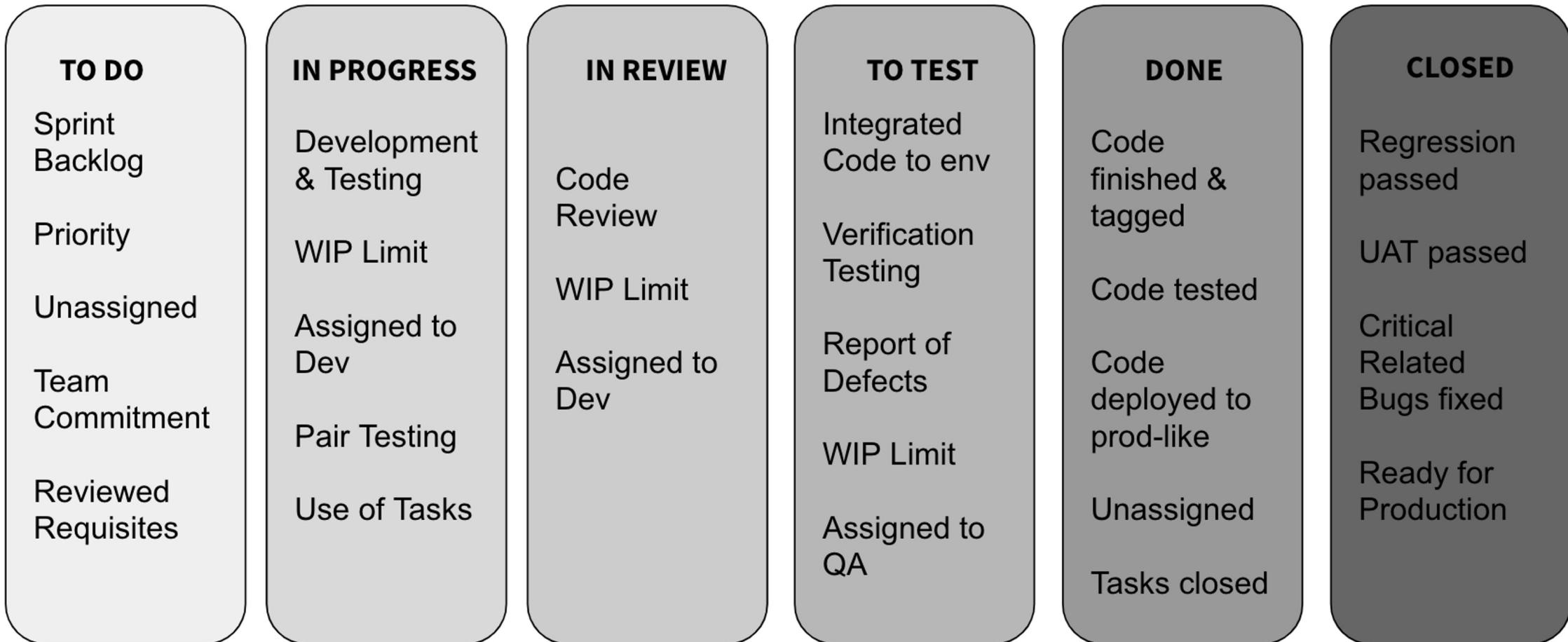
3. Team discusses.



4. Team estimates again.
Repeat until consensus reached.



EL PROCESO DE PRUEBAS: Historias de Usuario





EL PROCESO DE PRUEBAS: Historias de Usuario

Definition of „Done”

All unit tests passing



Acceptance tests satisfied

Code reviewed

Code coverage at 80%+

Code completed and checked in

Deployed

Story accepted

Definition of "Done"

- * Releasable
 - * Unit / Integration tested
 - * On demo server
 - * Acceptance tested

* Reduced technical debt
No increased ?



EL PROCESO DE PRUEBAS: Historias de Usuario

Proyectos / QA Proyecto
QA Proyecto
Proyecto de software de úl...

Sprint 1

Hoja de ruta Backlog Tablero Código Páginas Añadir elemento Configuración del pro...

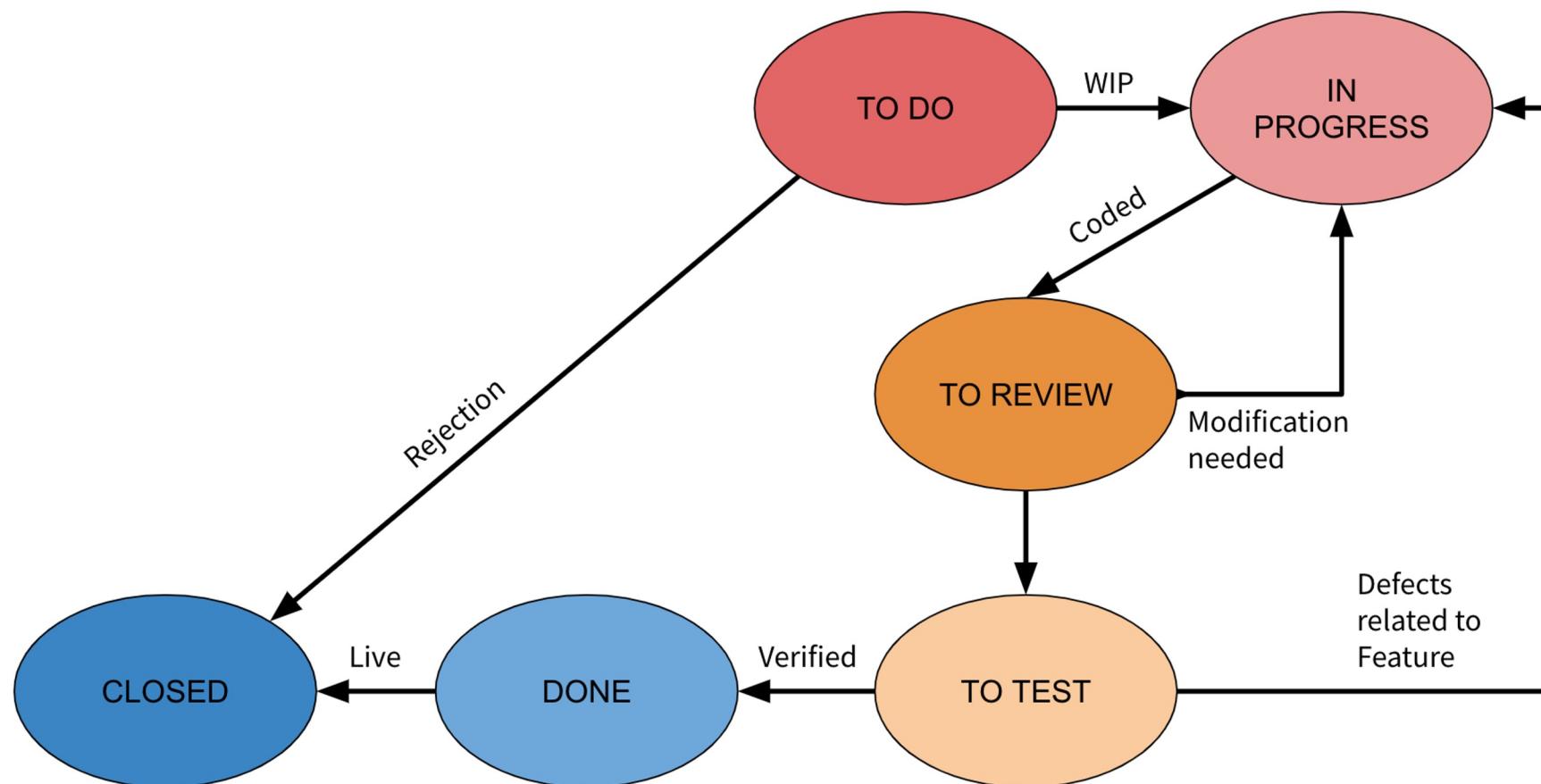
Restantes: 8 días Completar sprint ...

AGRUPAR POR Nada +

TO DO 2	IN PROGRESS 3	IN REVIEW 1	TO TEST 1	DONE 1 ✓
Watch the Payment summary QP-6 JG	Visualizar los detalles del pedido QP-2 IJ	Ver opiniones categorizadas QP-4 AC	Ability to share on Facebook a product from an online store along with its properties (pictures, description, price) QP-5 8 DH	Poder listar y elegir los diferentes medios de pago: Tarjeta. QP-1 DH
Ver los detalles de un producto QP-8 SL	Filtrar por precio QP-3 A	Ordenar productos por precio QP-7		



EL PROCESO DE PRUEBAS: Historias de Usuario



NIVELES DE PRUEBAS:



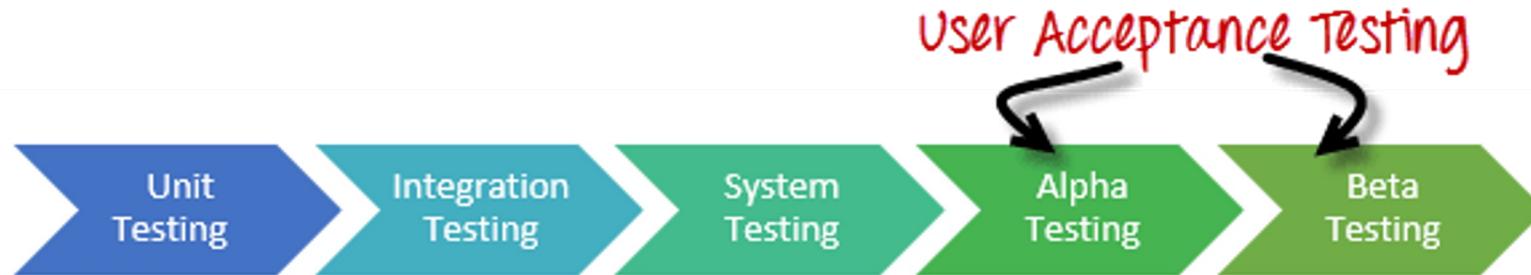
LEVELS OF TESTING

1	Unit Testing	Done by Developers
2	Integration Testing	Done by Testers
3	System Testing	Done by Testers
4	Acceptance Testing	Done by End Users

© www.SoftwareTestingMaterial.com



NIVELES DE PRUEBAS:



- Probar un sólo método se divide en un conjunto de pruebas aisladas que llamamos *Tests Unitarios*.
- Se trata de *tests* que prueban una lógica de negocio muy acotada dentro de una entidad específica. Estos *tests* pueden llegar a involucrar más de una función, pero nunca más de una clase.
- Si hay más de una clase involucrada en una funcionalidad (lo que es habitual), debemos *mockear* dichas clases.

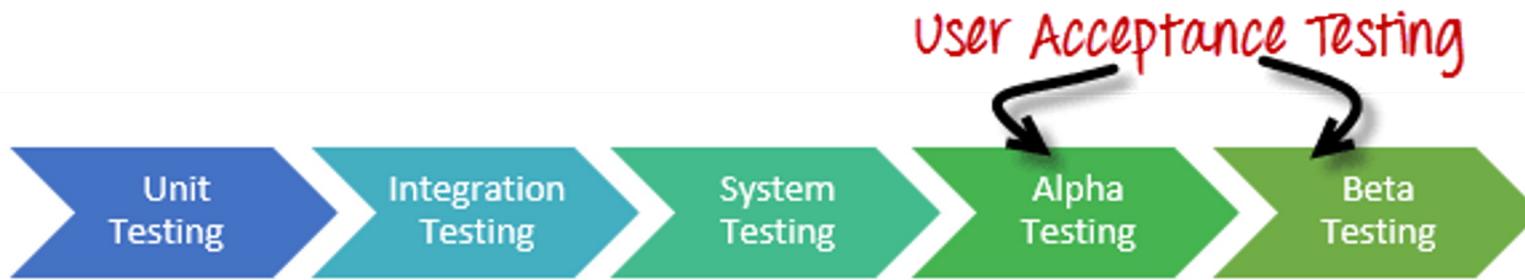


NIVELES DE PRUEBAS:

```
probar.py x
11
12 class TestOperaciones(unittest.TestCase):
13     def setUp(self):
14         # Aquí, opcionalmente, ejecuta lo que deberías ejecutar antes
15         # de comenzar cada test.
16         pass
17
18     def test_suma(self):
19         esperado = 3
20         actual = suma(1, 2)
21         # Pásalo en el orden: actual, esperado
22         self.assertEqual(actual, esperado)
23
24     def test_resta(self):
25         esperado = 5
26         actual = resta(10, 5)
27         # Pásalo en el orden: actual, esperado
28         self.assertEqual(actual, esperado)
29
30     def test_multiplicacion(self):
31         esperado = 50
32         actual = multiplicacion(10, 5)
```



NIVELES DE PRUEBAS:



- Un *test* de integración es aquel que realiza un *test* que involucra a más de una entidad (digamos clases).
- El objetivo no es probar cada una de ellas, sino cómo colaboran entre sí.
- Es importante aislar la prueba del resto de elementos de la plataforma, esto es, no depender de conexiones de red ni llamadas a servicios reales.
- Su objetivo es identificar errores introducidos por la combinación de programas o componentes probados unitariamente, para asegurar que la comunicación, enlaces y los datos compartidos ocurran apropiadamente.

NIVELES DE PRUEBAS:



The screenshot shows the Postman application interface. A blue callout box labeled "Test Description" points to the "Description" section of the request builder, which contains the following text:

This request tests the `gender` and `nat` (nationality) filters in combination. The results should contain a single, **male** user, from the **United States**. The tests verify that the user's gender and title are correct, and that the address is in one of the 50 U.S. states.

A blue callout box labeled "Test Script" points to the "Tests" tab in the request builder, which displays the following JavaScript code:

```
2 var data = JSON.parse(responseBody);
3 var user = data.results[0];
4
5 tests["A single user was returned"] = data.results.length === 1;
6
7 // Gender tests
8 tests["Gender is male"] = user.gender === "male";
9 tests["Title is Mr."] = user.name.title === "mr";
10
11 // Nationality tests
12 tests["The user is from the United States"] = user.nat === "US";
13 tests["The address is in the United States"] = arrayOfStates.includes(user.location.state);
14
15
```

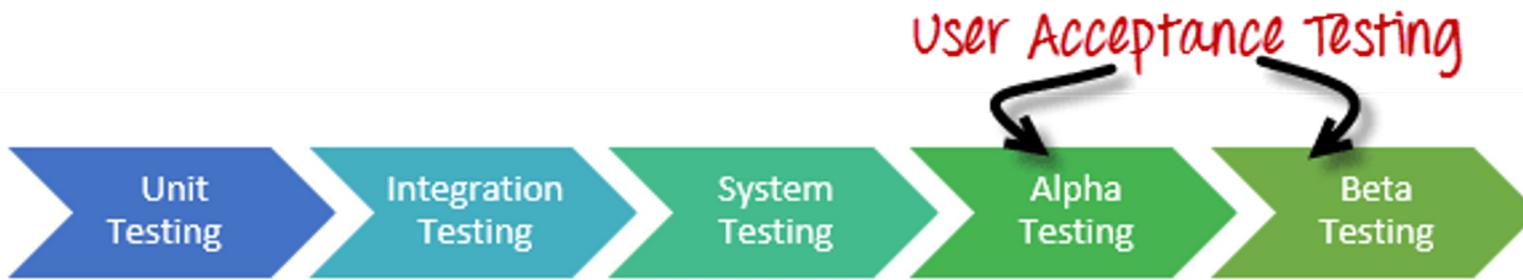
A blue callout box labeled "Test Results" points to the "Tests (5/5)" section of the results panel, which shows five successful test cases:

- PASS A single user was returned
- PASS Gender is male
- PASS Title is Mr.
- PASS The user is from the United States
- PASS The address is in the United States

The status bar at the bottom of the results panel indicates: Status: 200 OK, Time: 123 ms, Size: 1.24 KB.



NIVELES DE PRUEBAS:



- Cuando un test incluye todo el flujo (todas las capas) de un sistema, suele denominarse *Test End to End*.
- Una prueba E2E equivaldría a ejecutar la app en un dispositivo (emulador o real), y probar una funcionalidad, desde que el usuario interactúa con la UI hasta que el *feedback* se muestra por pantalla.
- Esta prueba tiene como objetivo verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las operaciones apropiadas funcionando como un todo.
- Es similar a la prueba de integración pero con un alcance mucho más amplio.

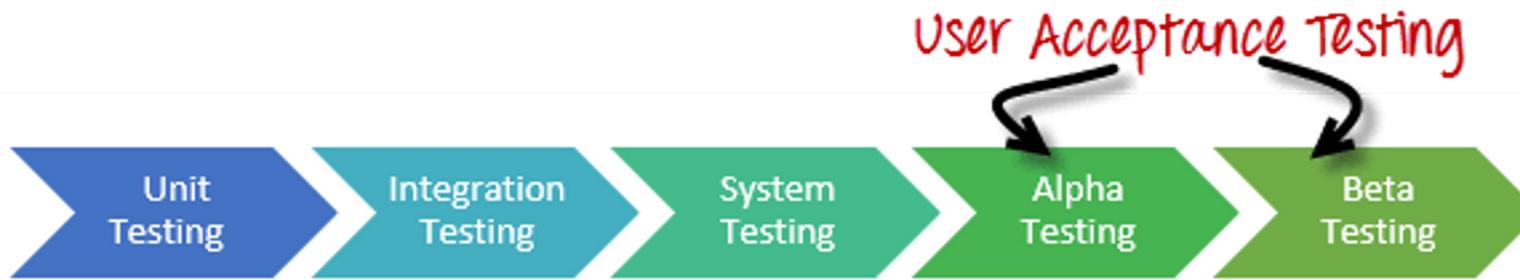


NIVELES DE PRUEBAS:

Project Name	Bank Website Functionality								
Test Case ID	Test Case Description	Pre Steps	Test Step	Preconditions	Test Data	Expected Result	Actual Result	Status	Comments
Test the Login Functionality in Banking	Verify login functionality with valid username & password		Navigate to login page			Able to see the login page	As expected	Pass	
			Enter valid username	Valid Username	username: choudaryac97@gmail.com	Credential can be entered	As expected	Pass	
			Enter valid password	Valid password	password: XXXXXXX@1	Credential can be entered	As expected	Pass	
			Click on login button			User logged	User logged successfully	Pass	
Test the Login Functionality in Banking	Verify login functionality with valid username & invalid password		Navigate to login page			Able to see the login page	As expected	Pass	
			Enter valid username	Valid Username	username: choudaryac97@gmail.com	Credential can be entered	As expected	Pass	
			Enter valid password	Invalid password	password: XXXXXXX@2	Credential can be entered	As expected	Pass	
			Click on login button			User logged	Unsuccessful login	Fail	



NIVELES DE PRUEBAS:



- Son realizadas principalmente por los usuarios con el apoyo del equipo del proyecto.
- El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales.
- La prueba de aceptación puede tener lugar a lo largo de semanas o meses, descubriendo así errores acumulados que pueden ir degradando el sistema.
- Es muy recomendable que las pruebas de aceptación se realicen en el entorno en que se va a explotar el sistema incluyendo el personal que lo va a manejar.
- En caso de un producto de interés general, se realizan pruebas con varios usuarios que reportarán sus valoraciones sobre el producto.



NIVELES DE PRUEBAS:

Alpha Testing is performed by the parent company's own developer & testing team.

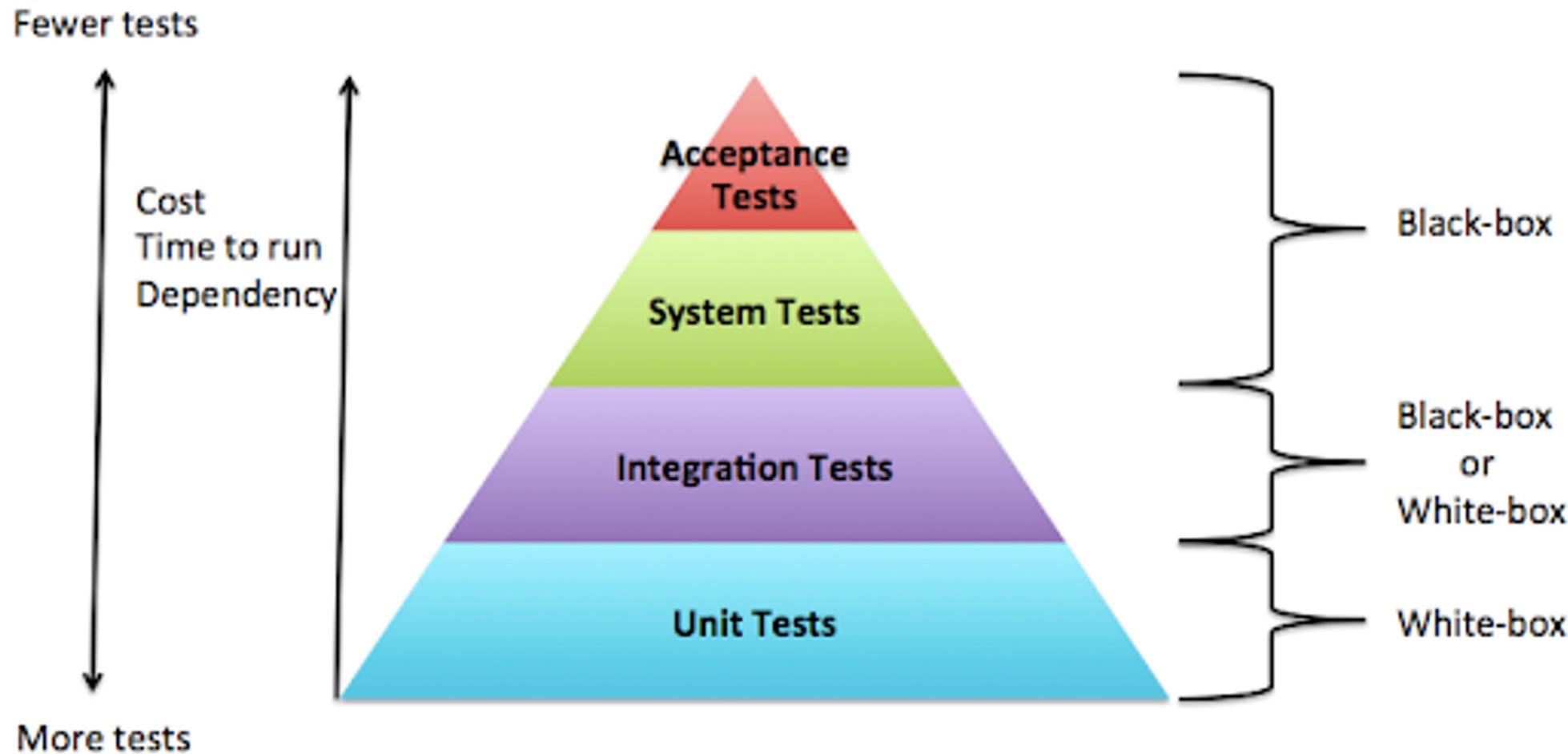
It involves the use of both black box as well as white box testing techniques

Beta testing is performed at the customer's end.

It incorporates the use of black-box technique in testing



NIVELES DE PRUEBAS:





FLAMA:*test*_
iG R A C I A S!