## Contexto (Identificar la universidad , cátedra, grupo, etc.)

## Título:

Los cuadrantes del testing Ágil

## Introducción

En el presente poster presentaremos cómo los cuadrantes de testing nos ayuda a descubrir los diferentes tipos de testing y cómo abordarlos. Agile Testing Quadrants as a convenient way to categorize tests. The four quadrants serve as guidelines to ensure that all facets of product quality are covered in the testing and developing process

## Materiales y métodos

Se utilizó un método de análisis y recolección de información de diferentes fuentes.

## Resultados

La idea de los cuadrantes del testing fue presentada por Brian Marick, que luego Lisa Crispin y Janet Gregory las refinaron en su libro.

Los cuadrantes del testing ágil ayudan a los testers a asegurarse de que han considerado todos los diferentes tipos de pruebas que se necesitan para entregar valor, planificar sus pruebas y asegurarse de que tienen todos los recursos que necesitan para realizarlas.

Para entender esto considere en un eje, dividimos la matriz en pruebas que apoyan al equipo y las pruebas que critican el producto. Entonces el significado y las connotaciones de la palabra "prueba" difieren enormemente en los dos casos.

Los cuadrantes de la izquierda incluyen pruebas que apoyan al equipo a medida que se desarrolla el producto. Este concepto de prueba para ayudar a los programadores es la mayor diferencia entre el testing en un proyecto tradicional y el testing en un proyecto ágil.

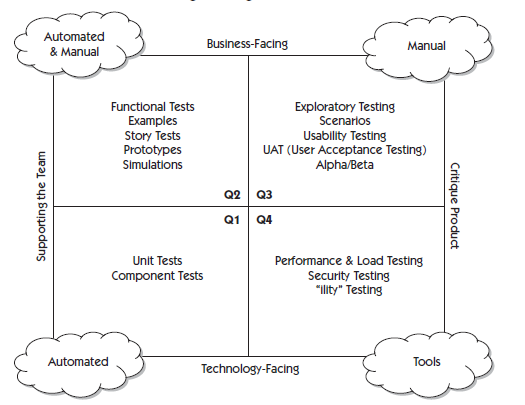
Escribes una prueba para ayudarte a aclarar tu pensamiento sobre un problema. Lo usa como un ejemplo ilustrativo de la forma en que debe comportarse el sistema. Es, afortunadamente, un ejemplo que verifica activamente el código, lo cual es tranquilizador. Estas pruebas también encuentran errores, pero ese es un propósito secundario.

Las pruebas que critican el producto apuntan a responder es el producto ¿competitivo? ¿La interfaz de usuario es tan intuitiva como debe ser? es la aplicación ¿seguro? ¿Están satisfechos los usuarios con el funcionamiento de la interfaz de usuario?

La palabra "crítica" no tiene un sentido negativo. Una crítica puede incluir elogios y sugerencias para mejorar. Se revisa el software de una forma constructiva para descubrir cómo se puede mejorarlo. Se pueden sugerir nuevos requerimientos y pruebas que apoyen al equipo y guíen el desarrollo.

El otro eje los divide en pruebas orientadas al negocio y orientadas a la tecnología. Estos enfoques nos permiten establecer criterios para saber cuándo cada característica ha cumplido con los criterios de funcionalidad y calidad del cliente.

El sistema de numeración por cuadrantes NO implica ningún orden. No trabajas a través de los cuadrantes del 1 al 4, en un estilo de cascada. Es solo una numeración arbitraria para hacer referencia a cada uno de manera resumida.



### Cuadrante 1

El cuadrante inferior izquierdo: desarrollo conducido por pruebas. Calidad interna

El propósito principal de las pruebas del cuadrante 1 es el desarrollo conducido por pruebas (TDD) o el diseño conducido por pruebas. Escribir pruebas ayuda a los programadores a diseñar bien su código, y les permite escribir código para sumar funcionalidades sin tener que preocuparse por si llegan a hacer algún cambio no deseado en el sistema.

A través de estas pruebas se puede verificar que las decisiones arquitectónicas y de diseño sean las correctas, así como también comprobar que la **calidad interna** sea la deseada, puesto que esta no se negocia.

### Cuadrante 2

El cuadrante superior: Desarrollo conducido por pruebas. calidad externa

Con respecto a este cuadrante también ayuda o da soporte al trabajo del desarrollo del equipo, pero a un alto nivel, más enfocado al negocio, ya que en este cuadrante aparecen test dirigidos al negocio, es decir que acá se define la calidad externa y los requerimientos que el cliente realmente quiere.

Además así como el cuadrante 1, la idea es que pueda proveer feedback de manera rápida mediante test automatizados, ya que estos tests primero guían al desarrollo de la funcionalidad y luego mediante los test automatizados ayudan a que no se introduzcan resultados inesperados a la hora de hacer refactoring.

Entonces en ese sentido podemos decir que este cuadrante ayuda a dar soporte al equipo de desarrollo.

Este cuadrante está orientado a confirmar si el comportamiento del sistema es el deseado a un alto nivel, por eso hablamos de que está enfocado al negocio, ya que ayuda al equipo a entregar valor al negocio requerido por los clientes.

Esto se logra mediante test que corren a un **nivel funciona**l, el cual cada uno verifica si se satisface los requerimientos del negocio.

Al tratarse de un desarrollo ágil, esos tests se derivan de **ejemplos** proporcionados por el equipo del cliente, los cuales describen los detalles de cada historia, teniendo de esta manera **pruebas de historia**. Además hay otro grupo de pruebas que también pertenecen a este cuadrante, mock-ups, los cuales son **prototipos** que ayudan a los expertos de la interacción del usuario a validar la GUI diseñada, en este caso este tipo de test no es automatizado.

### Cuadrante 3

Cuadrante superior derecho: Calidad percibida, y fallas ocultas.

Incluso cuando los programadores escriben código que pasa las pruebas enfocadas al negocio, puede que no estén entregando lo que el cliente realmente quiere.

En el cuadrante 3 se verifica que la aplicación realmente le entregue el valor de negocio esperado a los clientes a través de pruebas orientadas al negocio para criticar al producto. Se intenta emular la forma en la que un usuario real usará la aplicación, es decir que se utilizan pruebas manuales.

* Las pruebas de aceptación del usuario (UAT) le dan la posibilidad a los clientes de probar las nuevas funcionalidades y/o sugerir nuevas ideas para el futuro. También son estas las pruebas que normalmente se utilizan para que el Product Owner apruebe o no las Stories completadas.
* En las pruebas de usabilidad se utiliza un grupo focal para obtener información acerca de la forma en que la gente utiliza el sistema y así poder hacer un sistema más intuitivo.
* Durante las sesiones de testing exploratorio, el tester diseña las pruebas y las ejecuta como lo haría un usuario final. Al principio de cada proyecto, el tester planea diferentes escenarios de prueba, y a medida que están disponibles para probar, ejecutan las pruebas, las analizan y de esta forma encuentran nuevas áreas para explorar. Gracias a este tipo de pruebas se encuentran la mayoría de fallas grandes.

### Cuadrante 4

Desempeño. Requerimientos de calidad

Las pruebas del cuadrante 4 están orientadas a la tecnología, es decir que hablamos de pruebas más enfocadas a lo técnico que en términos de negocio, ya que la intención de estas pruebas es de alguna manera criticar las características del producto como la performance, robustez y la seguridad.

Es decir son pruebas relacionadas a los requerimientos no funcionales, en donde por lo general estos requerimientos no funcionales los clientes no lo dejan tan explícito. Los clientes asumen que los desarrolladores se encargan de cuestiones como la seguridad o la velocidad. Mientras que los desarrolladores esperan que eso quede más explícito.

Si conocemos los requisitos de rendimiento, seguridad, interacción con otros sistemas y otros atributos no funcionales antes de comenzar a codificar, es más fácil de diseñar y codificar con eso en mente.

Para crear y correr estos tipos de test se requiere el uso de herramientas especializadas y de experiencia adicional por parte de los testers, lo cual estas herramientas que automatizan ciertas pruebas son ideales tales como lacarga y la performance de pruebaque no se podrían hacer de otra forma

El equipo ya posee muchas de las habilidades necesarias para hacer estas pruebas.

## Conclusiones

Los cuadrantes son usados como una herramienta o guía para las conversaciones a la hora de planear las actividades de testing, ya que la mayoría de los problemas del software es debido a la pobre comunicación. Por lo tanto los cuadrantes, son una gran manera de facilitar las conversaciones entre las personas de diferentes especialidades como los testers, diseñadores, analistas del negocio, desarrolladores, etc. Ya que las diferentes perspectivas ayudan a evitar los “puntos ciegos” , por lo tanto ayuda a asegurar que el equipo está cubriendo con todos los diferentes tipos de prueba. Pero teniendo en cuenta que cada organización, producto y equipo es única, por lo cual se debe adaptar y debe ser considerada como una “herramienta” y no como una regla.

## Referencias y Agradecimientos

#### [**http://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants/**](http://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants/)

<https://www.shino.de/2012/07/30/the-testing-quadrants-we-got-it-wrong/>

<https://www.cutter.com/article/using-and-abusing-agile-testing-quadrants-part-i-468966>

<https://lisacrispin.com/downloads/AdpTestPlanning.pdf>

<http://www.exampler.com/old-blog/2003/08/22/#agile-testing-project-2>