

Pirámide de testing

Introducción

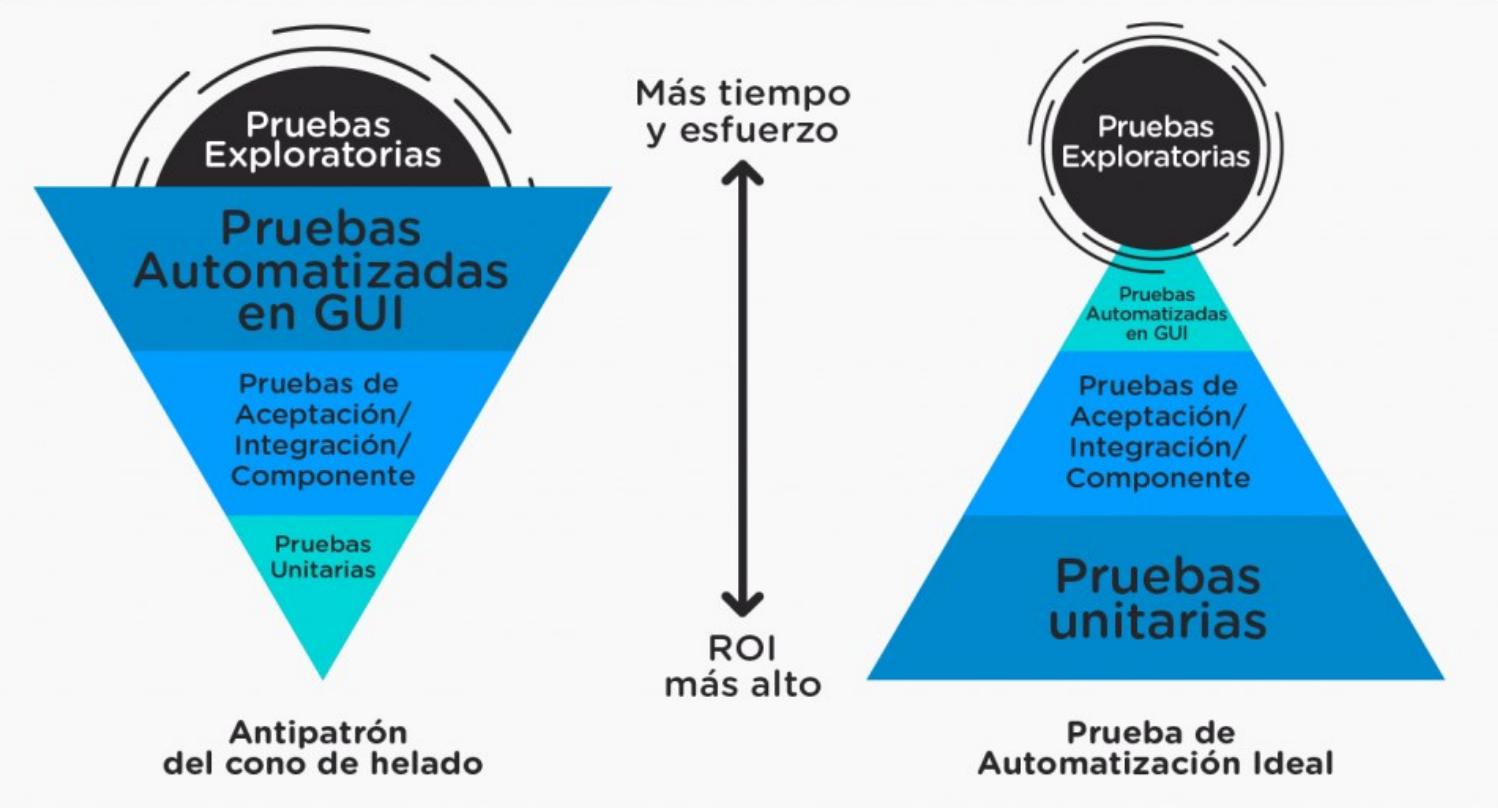
En los enfoques tradicionales, muchas veces en el proceso de test, se perdía mucho tiempo por la espera a un producto terminado para así comenzar a iniciar las pruebas con el fin de encontrar defectos en la construcción del sistema. Esto hacía que al momento de encontrar un bug, se dificultaba mantener el SW intentando solucionar errores que se podrían haber encontrado con antelación. Todas estas demoras hacen que para tener un release que fue estimado entregarlo en 6 meses de more uno o hasta dos años.

Además, la totalidad de pruebas se concentraban en sobre la UI, y considerando las UI pueden sufrir modificaciones durante el desarrollo del SW, volver a crear los casos de pruebas aunque sean automatizadas, consume tiempo y son muy costosas realizarlas. Por lo que este paradigma comienza a cambiar a partir de estas problemáticas.

Niveles de la pirámide de Cohn

En esta estratificación encontramos tres niveles de pruebas:

1. **Pruebas unitarias:** Realizadas por los desarrolladores. Les proporcionan datos específicos sobre errores en la construcción el release de manera temprana, antes del pasaje al testing.
2. **Pruebas de la capa de servicio:** Aquí se realizan todas las pruebas de aceptación, integración, etc., de cada componente de los servicios del SW, separados de la Interfaz de Usuario.
3. **Pruebas de IU:** Se focaliza en la correcta conexión entre los servicios y el componente de UI que corresponda, mostrando así los valores correctos en el campo de resultados.



Objetivo

Esencialmente, la pirámide de testing (pirámide de Cohn), establece una estratificación de los tipos de pruebas que se deben automatizar en función de la cantidad que se deberían hacer.

En este sentido, en un entorno ágil, las que se encuentran en la base de la pirámide son las que más deberían realizarse, ya que son las que menos tiempo y esfuerzo requiere además de que nos devuelven un ROI más alto, ya que está focalizado ese primer estrato a encontrar errores puntuales de manera más temprana. Y en la cima de la pirámide, encontramos los tipos de pruebas que son de las que menor cantidad de automatizaciones posee, al ser más costosas y menos mantenibles en el tiempo, dejando el foco de atención a ciertas pruebas manuales que se recomiendan hacer en ese nivel.

Conclusión

En los procesos ágiles a diferencia de los enfoques tradicionales podemos observar un mayor compromiso no sólo desde el lado del tester, si no de todo el equipo en su conjunto en la elaboración del producto de SW. En este sentido, Encontramos un cambio de paradigma a la hora de desarrollar software que va de ciclos estructurados a desarrollos orientados a pruebas.

Teniendo en cuenta este marco, notamos un gran cambio en el enfoque de las automatizaciones de las pruebas, dónde se observa que el foco pasó de ser del final del ciclo a ser la base de la estructura piramidal del desarrollo, dejando para explorar nuevos escenarios las pruebas manuales.

TRADICIONAL VS ÁGIL

Testing Tradicional	Testing Ágil
Los Unit Test tenían poca importancia, el programador se centraba en el desarrollo del sistema y prácticamente no había testing hasta llegar a dicha etapa.	Los Unit Test, son la base de pirámide, es por ello que deben ser lo suficientemente robustas y la mayor cantidad posible para proveer la retroalimentación que necesita el desarrollador. Cabe destacar que el desarrollo es del tipo TDD (desarrollo guiado por pruebas)
El testing, se centraba en pruebas (ya sean automatizadas y/o manuales) sobre la interfaz de usuario, para probar el producto en su totalidad realizado por el equipo de desarrollo	Debido a la robustez de las UT, las pruebas que realiza el tester en los niveles siguientes se orientan a lo específico de la funcionalidad nueva en el relase en curso.
Las pruebas demoraban más tiempo y eran más costosas, al ser sobre la interfaz de usuario, teniendo en cuenta que la misma puede sufrir variaciones durante el desarrollo, y un cambio sobre la misma puede ocasionar daños sobre uno o varios componentes	Las pruebas que se realizan sobre la UI, son automatizadas en la menor medida posible, debido a su alto costo. Es por ello que a este nivel se recomiendan realizar pruebas manuales exploratorias, con el objetivo de encontrar escenarios que no se hayan contemplado en las pruebas. Para así agregarlas como parte del backlog de ser necesario para su posterior implementación.

Grupo 2

Brizuela Marcelo	Legajo	69413
Jornet, Pablo	Legajo	62901
Lopez, Lautaro	Legajo	65576
Montiel, Fernando	Legajo	62106