

Testing Ágil: Estrategias y Herramientas

Estrategias de testing ágil

En el ámbito del desarrollo de software, es esencial establecer una estrategia de trabajo que facilite la planificación de procesos y la adaptación a posibles cambios en la composición del equipo. Una estrategia validada y debidamente documentada asegura la retención del conocimiento y la información, incluso cuando hay rotación de miembros o modificaciones en los integrantes del equipo.

La **estrategia de testing**, específicamente, se enfoca en definir cómo deben llevarse a cabo las tareas relacionadas con las pruebas de software. ¿Qué actividades se realizan? ¿Cómo se abordan? ¿En qué secuencia? ¿Por quiénes?

Existen diversos enfoques para abordar estas tareas, y desde una perspectiva general, se pueden clasificar en diferentes tipos de estrategias.

Analítica	Basada en el análisis sistemático de los requerimientos y/o en el análisis de los riesgos asociados al proyecto.
Basada en modelos	Analiza el comportamiento del sistema utilizando modelos como UML, tablas de decisión o diagramas de transición de estados.
Metódica	Basadas en sets de pruebas estándar, como un checklist que especifica pruebas que deben realizarse de manera consistente.
Basada en standards	Sigue los lineamientos de algún estándar externo o normativa como ISO para llevar a cabo las actividades de testing.
Reactiva	Estrategia que reacciona a los desarrollos listos para testearse. Se analiza y testea cuando el sistema ya está completamente desarrollado.

Consultiva	Estrategia basada en el consejo de expertos externos al equipo de testing, utilizando la experiencia y conocimientos de personas especializadas.
-------------------	--

Puede que algunas empresas utilicen un mix de estas estrategias sin tener una definida. En ocasiones alguna empresa o grupos de trabajo tienen estrategias bien consolidadas que se aplican a conciencia. De hecho, el equipo podría decidir modificar el tipo de estrategia elegida dependiendo del tipo de proyecto en el que esté trabajando.

En el marco del desarrollo ágil es valioso ser ordenado, responsable y metódico.

Es útil adelantar todo el trabajo posible para maximizar la eficiencia y la eficacia (en nuestro rol de identificar defectos). También es clave tener la flexibilidad para adaptarse y decidir cómo seguir cuando un plan cambia,

En resumen: Es recomendable anticipar las tareas de análisis, diseño de plan y preparación de ambiente para las pruebas. En caso de suspender el diseño de nuestro plan de pruebas y necesitar comenzar la ejecución de pruebas directamente, es bueno contar con al menos un borrador de las pruebas más críticas.

Aquí te dejamos un ejemplo de que se podría hacer en la etapa estática:

- Analizar requerimientos y modelos disponibles, para darle forma a las pruebas y cubrir requerimientos, flujos y transiciones de estado, siempre contemplando la variable riesgo para asignar prioridad.
- Revisar si hay sets de pruebas comunes que apliquen para el objeto bajo test. De ser así, agregarlo al plan. Por ejemplo: si el sistema tiene formularios es posible que se puedan aplicar pruebas que son comunes y requeridas para formularios
- Revisar si el sistema bajo test aplica para ser testeado bajo algún estándar. Por ejemplo: si el sistema debe cumplir con reglas de accesibilidad entonces deberían agregarse las pruebas correspondientes al plan de pruebas.
- Consultar con expertos en la materia para asegurarse de que se están cubriendo todos los aspectos claves del sistema y en caso de que se nos hubiera pasado algún caso de uso, agregarlo. Quizás el sistema en desarrollo está

relacionado a un área que no resulta familiar. Siempre es útil consultarla con un experto o SME – subject matter expert.

- Reportar los defectos que se hayan encontrado en la documentación, aclarar las dudas con las personas correspondientes, terminar de armar el plan de pruebas.

Una vez armado el plan de pruebas preparar el ambiente para ejecutar las pruebas cuando el sistema esté listo.

Si se avanza con el análisis en la etapa estática, es decir, antes de que el sistema esté listo para testearse lo único que queda, es ejecutar las pruebas. Hecho así, la etapa de ejecución puede ser una tarea bastante simple. Incluso puede delegarse o distribuirse entre varias personas.

Información de la industria :



A la hora del diseño del plan de pruebas, es recomendable empezar por una lista de pruebas que consistan en: Id, título y una descripción. Una vez que se generó el borrador con todas las descripciones de casos de prueba trabajar en los detalles de cada prueba.

Ten en cuenta que el título y la descripción deberían ser suficientes para entender exactamente qué debe probar la prueba.

De esta manera tendremos una lista con una descripción de todo lo que debemos probar en su totalidad.

Herramientas para gestión de pruebas

¿Qué es una herramienta de gestión de pruebas? Una herramienta de gestión de pruebas es un software diseñado para facilitar al menos una actividad relacionada con las pruebas, como planificación, control, diseño, construcción de datos, ejecución y análisis. Estas herramientas brindan eficiencia al proceso de testing, aunque su implementación y configuración pueden percibirse inicialmente como un esfuerzo costoso.

A pesar de la inversión inicial y el mantenimiento continuo que requieren, las herramientas de testing ofrecen beneficios significativos en comparación con la ejecución manual de actividades de testing. Algunos de los beneficios incluyen el ahorro de tiempo en tareas repetitivas, mejor trazabilidad, organización y acceso eficiente a la información, así como la capacidad de diseñar y gestionar procesos y flujos de trabajo.

Algunos de los objetivos que cumplen las herramientas de gestión de pruebas son:

- De manejo de defecto
- De manejo de requerimientos
- De integración continua
- De soporte para pruebas estáticas (de revisión)
- De soporte para diseño de casos de prueba e implementación
- De preparación de datos para pruebas
- De soporte para ejecución de pruebas
- De captura y grabación de pantalla

Es recomendable, que antes de adquirir o utilizar una herramienta de gestión de pruebas, es crucial evaluar varios aspectos. Algunas consideraciones importantes incluyen:

- ¿La herramienta cumple con los requerimientos del equipo y del proyecto?
- ¿Necesita integrarse con otros sistemas para proporcionar la funcionalidad requerida?
- ¿Hay alguien en el equipo con conocimientos técnicos para integrar, configurar y mantener la herramienta?
- ¿Existen políticas empresariales o de seguridad que puedan afectar la implementación de la herramienta?
- ¿Hay presupuesto y recursos disponibles para la implementación y mantenimiento?
- ¿Es necesaria una estrategia de adopción para garantizar que los miembros del equipo utilicen la herramienta de la manera esperada?
- Si la herramienta se ejecutará en un entorno de testing, ¿es compatible con ese entorno, el lenguaje utilizado y el modo en que se ha escrito el código?

Es decir, para la selección de una herramienta de ejecución de tests automatizados, analiza qué prestaciones tiene la herramienta y si cumple con lo que tu proyecto requiere. A su vez, debes considerar si la herramienta es compatible con el sistema que se está desarrollando.



¿Quieres saber más sobre las herramientas para la gestión de pruebas? Par empezar a utilizar alguna herramienta de automatización, puedes ver la siguiente serie de videos:

<https://testautomationu.applitools.com/setting-a-foundation-for-successful-test-automation/> (verlos todos te llevará aproximadamente 40 minutos. Y recuerda, que si prefieres, puedes añadirle los subtítulos en español).

HERRAMIENTA : JIRA

Jira es un software de gestión de proyectos y su uso está muy extendido en el mundo del desarrollo ágil.

El foco principal de esta herramienta es la **gestión de proyectos**. Sin embargo, tiene cierto grado de configurabilidad y permite la integración a otras herramientas que amplían sus prestaciones, pudiendo abarcar varios aspectos relacionados a la gestión de prueba tales como: manejo de pruebas, defectos, requerimientos e [integración continua](#).

Jira integrado a Zephyr soporta el diseño de casos de prueba, la preparación de datos para pruebas e implementación y la ejecución de las mismas.

A su vez, Zephyr puede integrarse a herramientas de automatización como Cucumber, TestNG, PyTest u otras¹

Hay que tener en cuenta que para que esta herramienta nos ayude a cerrar el ciclo de acciones relacionadas a la gestión de pruebas, requiere que la configuración refleje los procesos reales del trabajo del equipo.

Puntos a tener en cuenta para configurar:

1

(https://smartbear.com/blog/level-up-test-automation-with-zephyr-scale/?utm_source=beamer&utm_medium=standalone&utm_campaign=Level-Up-Test-Automation-With-Zephyr-Scale&utm_content=ctalink)

- El equipo usa Épicas, historias de usuario, subtareas e incidencias para gestionar sus proyectos y reportes de defectos? ¿Usa otra terminología o clasificaciones?
- ¿Qué clasificaciones de severidad utiliza? ¿Cuántos niveles de prioridad hay? ¿Qué pasos o estados puede atravesar un caso de desarrollo (o historia de usuario)?
- ¿Qué pasos o estados puede atravesar un caso para una incidencia o defecto? ¿Cómo es el proceso de reporte de defectos?

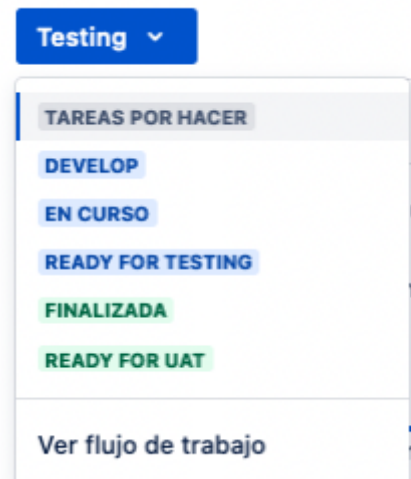
A continuación les mostramos cómo se visualiza en un tablero tipo Kanban, dentro de Jira, los diferentes estados del trabajo en curso y terminados. Cada columna representa un estado. Cada registro de caso de desarrollo, incidencia o de los tipos de proyecto que se usen, se ubican bajo la columna que representa el estado actual del proyecto.

TO DO	IN PROGRESS	DEVELOP	TESTING	READY FOR TESTING	READY FOR UAT	DONE
<div> <div> </div> <div> NM-1213 </div> <div> FINALIZADA </div> </div> <div> a sub-tarea [Scrum] Refinamiento del backlog </div>						<div> <div>Maxi</div> <div> </div> <div>NM-1219</div> </div> <div> <div>Pamela</div> <div> </div> <div>NM-1220</div> </div> <div> <div>Romina</div> <div> </div> <div>NM-12</div> </div> <div> <div>Diego</div> <div> </div> <div>NM-12</div> </div>

Estados:

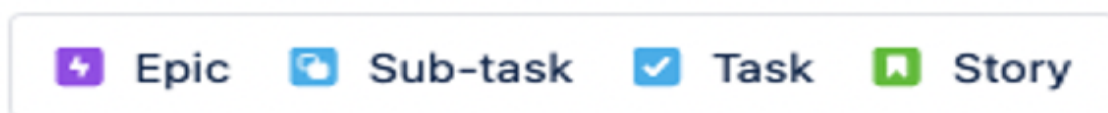
- Por hacer
- En progreso
- En desarrollo
- Listo para testear
- Testeando
- Listo para UAT
- Terminado

Para actualizar el estado de un proyecto simplemente se selecciona la opción correspondiente del listado.

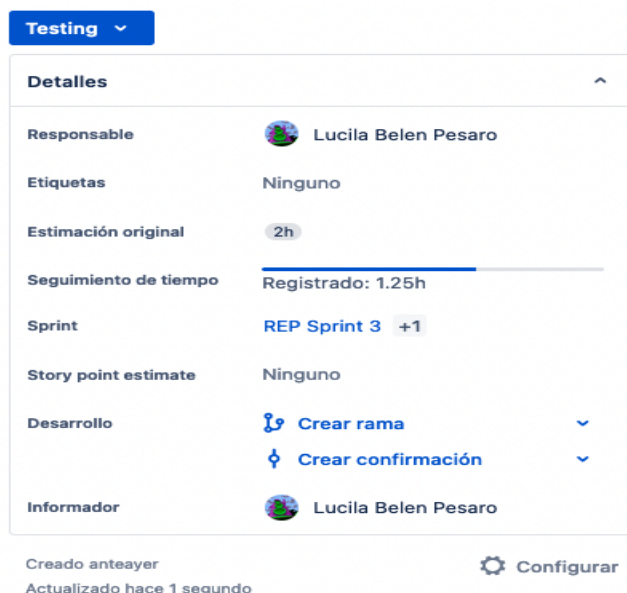


Esto es solo un ejemplo muy básico y tu equipo podría manejar otros estados u otra terminología para referirse a las diferentes fases por las que atraviesa cada proyecto.

Los tipos de proyecto también se pueden configurar. Por defecto podrás encontrar:



Así se observa el panel de detalles de un proyecto que está siendo testeado:



Tal como se observa, es posible asignar:

- Responsable al proyecto
- Nombre de quien creo o informó el proyecto

- Etiquetas
- Estimación y tracking real de horas dedicadas
- Story points si se estima en story points
- Sprint
- Rama de desarrollo

Ten en cuenta que se puede configurar los campos que se deben mostrar y su orden.

Jira permite filtrar la información para visualizarla en listas. Los usuarios con permisos para crear y guardar sus listas o reportes, pueden generar marcadores.

Esto es útil para visualizar el trabajo pendiente rápidamente.

MARCADOS COMO FAVORITOS

☰ Mis incidencias abiertas ★

☰ Informadas por mí ★

☰ Vistas recientemente ★

☰ Incidencias abiertas ★

☰ Actualizadas recientemente ★

☰ Lucila - time tracking ★

☰ READY FOR TESTING ★

☰ TO DO ★

T	Clave	Responsable	Pr	Estado	Resolución	Creada	Actualizada	Tiempo Trabajado	Estimación original
	REP-79	Lucila Belen Pesaro		TESTING	Sin resolver	01/mar/23	02/mar/23	1,25h	
	REP-2	Lucila Belen Pesaro		FINALIZADA	Listo	27/oct/22	16/dic/22	18,67h	15h

Filtrar información y generar tus propios reportes y listas, anclarlos a un menú de acceso rápido es verdaderamente importante para ayudarte a:

- Visualizar rápidamente los proyectos que tienen en proceso
- Tener visibilidad sobre proyectos o tareas prioritarias
- Encontrar la información de forma fácil y rápida

- Compartir información o reportes con otras personas



💡 **Información de la industria :**

Existen otras herramientas de gestión de prueba, pero te presentamos Jira y Zephyr Scale porque son las más populares y utilizadas. Además, pueden configurarse para cubrir todo el ciclo de desarrollo y testing de productos de software. Son intuitivas. ¡Tu próximo desafío es usarlas