

Análisis y Planificación: Estimación de costos

Introducción

La calidad del software depende en gran medida de la capacidad para realizar pruebas exhaustivas y precisas, lo que a su vez implica una asignación adecuada de recursos financieros y humanos.

Seguramente, con tiempo y experiencia, tus estimaciones se vuelvan más acertadas. Es perfectamente normal hacer estimaciones erróneas durante las primeras tareas, pero esto sirve de experiencia.

La Importancia de la estimación en casos de pruebas se basa en algunos puntos principales:

- **Garantizar la Calidad del Software:** Una estimación precisa de pruebas permite asignar los recursos necesarios para garantizar la calidad del software.
- **Optimización de Recursos:** La estimación ayuda a identificar la cantidad adecuada de recursos, tanto humanos como materiales, necesarios para realizar pruebas de manera eficiente.
- **Planificación Efectiva:** La estimación proporciona una base sólida para la planificación del proyecto. Permite establecer plazos realistas y definir hitos alcanzables
- **Priorización de Casos de Pruebas:** Con un presupuesto claro, se pueden priorizar los casos de pruebas en función de su importancia y criticidad

Los principales factores que inciden en la estimación:

- **Experiencia del Equipo:** La experiencia y habilidades del equipo de pruebas también influyen en las estimaciones. Un equipo altamente capacitado puede realizar pruebas de manera más eficiente, mientras que un equipo menos experimentado puede requerir más tiempo y esfuerzo.
- **Herramientas y Tecnologías:** La elección de herramientas de prueba y tecnologías puede impactar en las estimaciones. Herramientas más

avanzadas pueden requerir inversiones adicionales, pero también pueden mejorar la eficiencia y la cobertura de las pruebas.

- **Requerimientos Cambiantes:** Cambios frecuentes en los requisitos del software pueden afectar los casos de pruebas y, por lo tanto, en las estimaciones calculadas. Una gestión efectiva de cambios es esencial para mantener la estimación bajo control.
- **Complejidad del Software:** La complejidad del software influye directamente en la cantidad de esfuerzo y recursos necesarios para realizar pruebas efectivas. Software más complejo requerirá más casos de pruebas.

A continuación, te comentamos algunos encuadres que pueden servir para realizar una estimación:

1. Basarse en métricas

Este método de estimación implica utilizar datos recopilados de proyectos anteriores, especialmente aquellos que son similares en naturaleza al proyecto actual. Esta estrategia se apoya en la premisa de que las experiencias pasadas pueden proporcionar una guía valiosa para prever el tiempo necesario para actividades específicas, en este caso, el desarrollo y la ejecución de pruebas.

Al considerar métricas, se examinan diversos elementos clave de proyectos anteriores relacionados. Estos pueden incluir:

- **Cantidad de Casos de Prueba:** Analizar la cantidad de casos de prueba escritos y ejecutados en proyectos previos proporciona una base para estimar cuánto tiempo podría requerir un plan de pruebas similar.
- **Tiempo de Desarrollo y Ejecución:** Evaluar el tiempo dedicado al desarrollo y ejecución de pruebas en proyectos anteriores permite establecer comparaciones útiles para determinar la duración esperada en el nuevo proyecto.
- **Cantidad de Defectos Encontrados:** Examinar la cantidad y complejidad de los defectos descubiertos durante las pruebas anteriores brinda información sobre la posible eficiencia del proceso de prueba y las actividades de corrección.
- **Tiempo de Corrección y Re-Testeo:** Analizar el tiempo empleado en la corrección de defectos identificados y el subsiguiente re-testeo ofrece una

visión detallada de los esfuerzos necesarios para abordar problemas y validar soluciones.

Estos datos históricos permiten obtener un estimado más preciso del tiempo necesario para el desarrollo y la ejecución de un plan de pruebas específico en un proyecto similar. Sin embargo, es crucial adaptar estas métricas a las particularidades del nuevo proyecto y considerar posibles cambios en la tecnología, el equipo o los requisitos que puedan afectar la duración de las pruebas.

2. Basarse en expertos en la materia (Wide band delphi approach)

Este encuadre consiste en consultar estimaciones a los responsables de cada una de las tareas o a personas consideradas expertas ya que son quienes tienen conocimiento sobre las herramientas a utilizar y experiencia en el área.

En este método, se busca la opinión de profesionales como consultores, analistas de calidad con experiencia, desarrolladores y cualquier persona que tenga un profundo conocimiento y comprensión del sistema en desarrollo. Su experiencia les permite evaluar de manera precisa los recursos requeridos y el tiempo necesario para llevar a cabo cada tarea.

Es válido considerar la posibilidad de **combinar** este enfoque con la metodología basada en métricas, aprovechando así la experiencia subjetiva de los expertos junto con datos objetivos obtenidos de proyectos anteriores. La flexibilidad para probar nuevas técnicas también se sugiere, ya que la mejora continua en los métodos de estimación puede conducir a resultados más precisos y adaptados a las particularidades de cada proyecto.

3. Base por puntos – Base por horas

La estimación de esfuerzo puede realizarse de diversas maneras, ya sea en términos de tiempo neto (horas, minutos, días, semanas, meses) o mediante la metodología ágil que emplea la estimación por puntos, también conocidos como story points (puntos de historia).

¿En qué consiste la estimación por puntos?

La estimación por puntos implica representar el tiempo necesario para completar una tarea o proyecto en función del esfuerzo y la complejidad asociados. Esta modalidad tiene en cuenta no solo el volumen de trabajo, sino también factores como la incertidumbre y los imprevistos que puedan surgir durante el desarrollo.

En un entorno ágil, caracterizado por iteraciones periódicas, cada ciclo de trabajo se planifica en función de estos puntos. Al iniciar un ciclo o sprint, el equipo conoce su capacidad, expresada, por ejemplo, en 8 o 34 puntos. Durante la planificación, si la capacidad del equipo es de 34 puntos y se identifica una tarea o historia que requiere 32 puntos, esa será la única tarea asignada para ese ciclo.

Es común que los equipos utilicen la serie Fibonacci para asignar puntos de historia, donde cada número en la serie es la suma de los dos anteriores. Esta elección se debe a que la serie Fibonacci permite una representación equilibrada y consistente del esfuerzo, ya que la diferencia entre los números refleja de manera tangible la variación en la complejidad.

A diferencia de las escalas lineales (por ejemplo: 1, 2, 3, 4, 5, 6, 7... puntos), donde la diferencia entre los números es constante, la serie Fibonacci ofrece una proporcionalidad que facilita la interpretación del esfuerzo. Por ejemplo, el número 2 podría representar dos historias pequeñas (de 1 y 1 punto), mientras que el número 13 podría ser equivalente a una historia mediana (8) más otra un poco más pequeña (5). tangible la variación en la complejidad.

Esta proporcionalidad permite que los equipos utilicen un número como referencia y calculen la complejidad de cada historia en función de ese parámetro y de su experiencia acumulada. La estimación por puntos proporciona una herramienta flexible y equitativa para planificar y ejecutar proyectos de manera ágil.

TIPS ÚTILES

Cuando no se tiene una idea precisa del tiempo que puede estimar ejecutar una prueba, es conveniente, ejecutar la prueba más larga y compleja de principio a fin.

Se toma nota del tiempo de ejecución y se usa como parámetro para calcular el tiempo probable de las otras pruebas de menor complejidad. Siempre es recomendable, estimar los tiempos con holgura.

El buffer time es tu aliado

El tiempo de reserva es tu aliado. Recuerda que una estimación es sólo una aproximación. Es beneficioso estimar los tiempos y esfuerzos con holgura para evitar que el equipo de desarrollo se vea presionado por el tiempo.

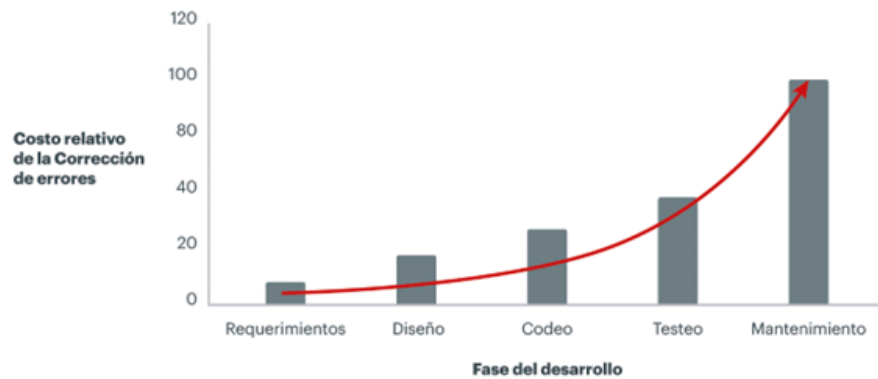
Para esto siempre es bueno agregar un **buffer time**, es decir, un margen adicional, para que el equipo pueda hacer frente a imprevistos sin tener que recurrir a horas extras.

Aprender de MI propia experiencia

Aprender de la experiencia es fundamental en la mejora continua de la estimación de esfuerzos. Si inicialmente estimé que una prueba, como la del login, requeriría 4 horas o 2 puntos y, en la práctica, el tiempo resultó insuficiente, esta experiencia me proporciona un nuevo punto de referencia.

Cada proyecto y cada tarea son oportunidades para perfeccionar la habilidad de estimar. Aprender de la experiencia pasada proporciona una base sólida para tomar decisiones más informadas y realizar estimaciones más precisas en el futuro.

CONSECUENCIAS DE UN ANÁLISIS DE REQUISITOS LEVE/BAJO



“Consecuencias en el costo de un análisis deficiente”.

El cuadro muestra la relación entre la fase del desarrollo y el costo de la corrección de errores. Si el análisis de requisitos no se realiza en la fase inicial del SDLC, entonces su impacto es enorme para solucionarlo en fases posteriores. Cuando el análisis de requisitos no se produce de forma satisfactoria, pueden aparecer algunas consecuencias como ser: la entrega incorrecta de funciones, mala calidad del producto, una gran cantidad de cambios para corregir fallas del sistema, extensión de los plazos iniciales del proyecto, entre otros. Cuanto más se extiende el tiempo en analizar el requisito, más cuesta hacerlo luego y eso afecta la entrega y la calidad del proyecto.