

### Tarjeta

Una tarjeta es un medio que describe la historia de un usuario (por ejemplo, puede ser una tarjeta física en forma de una hoja de papel colocada en un tablero scrum<sup>5</sup> o una entrada en un tablero electrónico). La tarjeta identifica el requisito, su importancia o prioridad, las limitaciones, el tiempo esperado de desarrollo y prueba y, muy importante, los criterios de aceptación de la historia. La descripción debe ser precisa, ya que se utilizará en la cartera de productos.

Los equipos ágiles pueden documentar historias de usuarios de diversas formas. Un formato popular es:

Como (usuario previsto)

Quiero (acción prevista), de

modo que (propósito/resultado de la acción, beneficio obtenido),

seguido de los criterios de aceptación. Estos criterios también se pueden documentar de diferentes maneras (ver Sección 4.5.2). Independientemente del enfoque adoptado para documentar historias de usuarios, la documentación debe ser concisa y suficiente para el equipo que la implementará y probará.

Las tarjetas representan los requisitos del cliente en lugar de documentarlos. Si bien la tarjeta puede contener el texto de la historia, los detalles se resuelven en la conversación y se registran en la confirmación [61].

### Conversación La

conversación explica cómo se utilizará el software. La conversación puede ser documentada o verbal. Los evaluadores, que tienen un punto de vista diferente al de los desarrolladores y representantes comerciales, realizan valiosas contribuciones al intercambio de pensamientos, opiniones y experiencias. La conversación comienza durante la fase de planificación del lanzamiento y continúa cuando se planifica la historia. Se lleva a cabo entre partes interesadas con tres perspectivas principales sobre el producto: el cliente/usuario, el desarrollador y el evaluador.

### Confirmación La

confirmación se presenta en forma de criterios de aceptación, que representan elementos de cobertura que transmiten y documentan los detalles de la historia de un usuario y que pueden usarse para determinar cuándo está completa una historia. Los criterios de aceptación suelen ser el resultado de una conversación. Los criterios de aceptación pueden verse como condiciones de prueba que el evaluador debe verificar para verificar que la historia esté completa.

Las historias de usuarios deben abordar características tanto funcionales como no funcionales. Normalmente, la perspectiva única del evaluador mejorará la historia del usuario al identificar detalles faltantes o requisitos no funcionales. El evaluador puede brindar información haciendo preguntas abiertas a los representantes comerciales sobre la historia del usuario, sugiriendo formas de probarla y confirmando los criterios de aceptación.

La autoría compartida de historias de usuarios puede utilizar técnicas como la lluvia de ideas o los mapas mentales.<sup>6</sup> Las buenas historias de usuario cumplen con las llamadas propiedades INVEST, es decir, son [62]:

<sup>5</sup> En Scrum, un tablero de scrum es un tablero físico de uso opcional que muestra el estado actual de una iteración. Proporciona una representación visual del cronograma de trabajo a realizar en una iteración determinada.

<sup>6</sup> Para obtener más información sobre mapas mentales, consulte, por ejemplo, <https://www.mindmapping.com/>.

- Independientes: no se superponen y pueden desarrollarse en cualquier orden. • Negociables: no son contratos de características explícitas; más bien, los detalles serán creados conjuntamente por el cliente y el desarrollador durante el desarrollo.
- Valiosos: una vez implementados, deberían aportar valor agregado al cliente. • Estimable: el cliente debería poder priorizarlos y el equipo debería poder estimar su tiempo de finalización para gestionar más fácilmente el proyecto y optimizar los esfuerzos del equipo.
- Pequeño: esto hace que su alcance sea fácil de entender para el equipo y la creación de las historias mismas son más manejables.
- Comprobable: ésta es una característica general de un buen requisito; la corrección de la implementación de la historia debe ser fácilmente verificable.

Si un cliente no sabe cómo probar algo, puede indicar que la historia no es lo suficientemente clara o que no refleja algo de valor para el cliente o que el cliente simplemente necesita ayuda con las pruebas [62].

Ejemplo El siguiente ejemplo muestra una historia de usuario escrita desde la perspectiva de un cliente de una aplicación web de banca electrónica. Tenga en cuenta que los criterios de aceptación no necesariamente se derivan directamente del contenido de la historia en sí, sino que pueden ser el resultado de una conversación entre el equipo y los clientes. El evaluador utiliza estos criterios para diseñar pruebas de aceptación que verificarán que la historia se haya implementado total y correctamente.

Historia de usuario US-001-03

Como cliente del banco

Quiero poder iniciar sesión en el sistema.

Para poder utilizar productos bancarios

Criterios de aceptación

- El inicio de sesión debe ser una dirección de correo electrónico válida. • El sistema rechaza el intento de inicio de sesión con una contraseña incorrecta. • El sistema rechaza el intento de inicio de sesión por un inicio de sesión de usuario inexistente. • El sistema permite al usuario ingresar en los campos "login" y "contraseña" únicamente caracteres alfanuméricos, así como un punto y el símbolo "@".
- Los campos de inicio de sesión y contraseña pueden tener hasta 32 caracteres. • Después de hacer clic en el enlace "recordar contraseña" e ingresar la dirección de correo electrónico del usuario, se envía a esta dirección un enlace al sistema de recordatorio de contraseña. • El sistema inicia el proceso de inicio de sesión cuando se hace clic en el botón "iniciar sesión" o cuando se presiona la tecla Enter; en el último caso, el proceso de inicio de sesión comienza cuando se cumplen tres condiciones simultáneamente: (1) el campo "iniciar sesión" no está vacío, (2) la ventana activa es la ventana "contraseña", (3) el campo "contraseña" no está vacío.

## 4.5.2 Criterios de aceptación

Los criterios de aceptación son condiciones que un producto (en la medida descrita por la historia del usuario o el Elemento de la cartera de productos del que forman parte estos criterios de aceptación) debe cumplir para ser aceptado por el cliente. Desde esta perspectiva, los criterios de aceptación pueden verse como condiciones de prueba o elementos de cobertura que deben verificarse mediante pruebas de aceptación.

Se utilizan criterios de aceptación:

- Definir los límites de la historia del usuario
- Llegar a un consenso entre el equipo de desarrollo y el cliente
- Describir escenarios de prueba tanto positivos como negativos
- Como base para las pruebas de aceptación de la historia del usuario (ver Sección 4.5.3)
- Como herramienta para una planificación y estimación precisas

Los criterios de aceptación pueden considerarse de gran ayuda para determinar y evaluar la Definición de Listo (DoR) o Definición de Hecho (DoD). Un equipo puede decidir no iniciar la implementación cuando los criterios de aceptación de una historia de usuario no se obtienen de manera exhaustiva. De manera similar, un equipo puede decidir que una Historia de Usuario no se considera candidata para lanzamiento (o demostración) cuando no se cumplen todos los Criterios de Aceptación (cobertura de los Criterios de Aceptación por debajo del 100%).

Los criterios de aceptación se discuten durante la conversación (ver Sección 4.5.1) y se definen en colaboración entre representantes comerciales, desarrolladores y evaluadores.

Los criterios de aceptación, si se cumplen, se utilizan para confirmar que la historia del usuario se ha implementado completamente y de acuerdo con la visión compartida de todas las partes interesadas. Proporcionan a los desarrolladores y evaluadores una visión ampliada de la función que validarán los representantes comerciales (o sus representantes). Se deben utilizar pruebas tanto positivas como negativas para cubrir los criterios. Durante la confirmación, diferentes partes interesadas desempeñan el papel de probador. Estos pueden variar desde desarrolladores hasta especialistas centrados en el rendimiento, la seguridad, la interoperabilidad y otros atributos de calidad. El equipo ágil considera que una tarea está completa cuando se considera que se ha cumplido un conjunto de criterios de aceptación.

No existe una forma única y establecida de escribir criterios de aceptación para una historia de usuario. Los dos formatos más comunes son:

- Criterios de aceptación orientados a escenarios.
- Criterios de aceptación orientados a reglas.

**Criterios de aceptación orientados a escenarios** Este formato para escribir criterios de aceptación suele utilizar el formato Dado/Cuándo/Entonces conocido por la técnica BDD. Sin embargo, en algunos casos, resulta difícil encajar los criterios de aceptación en dicho formato, por ejemplo, cuando el público objetivo no necesita los detalles exactos de los casos de prueba. En ese caso, se puede utilizar un formato orientado a reglas.

**Criterios de aceptación orientados a reglas**

Normalmente, en este formato, los criterios de aceptación toman la forma de una lista de verificación con viñetas o una forma tabular de asignación de entradas a salidas. Algunos frameworks y lenguajes

para escribir los criterios de estilo Dado/Cuándo/Entonces (p. ej., Gherkin) proporcionan mecanismos que Le permite crear un conjunto de reglas dentro de un escenario, cada una de las cuales será probada. por separado (esta es una forma de prueba basada en datos).

La mayoría de los criterios de aceptación se pueden documentar en uno de los dos formatos mencionados. arriba. Sin embargo, el equipo puede utilizar cualquier otro formato no estándar, siempre y cuando el Los criterios de aceptación están bien definidos y son inequívocos.

Junto a los criterios de aceptación orientados a escenarios y a reglas mencionados arriba, los equipos menos maduros generalmente obtienen y documentan los criterios de aceptación de forma gratuita. formato, como adiciones o subsecciones a una Historia de Usuario. Cualquiera que sea el formato que decida un equipo de uso, la trazabilidad bidireccional entre la historia de usuario y sus criterios de aceptación Es vital.

Ejemplo La historia de usuario descrita en la sección anterior (iniciar sesión en el sistema de banca electrónica) tiene criterios de aceptación descritos en forma de reglas. Algunos de pueden detallarse y representarse en forma de mapeo de entradas y salidas mediante definir ejemplos específicos de datos de prueba. Por ejemplo, en Gherkin, los primeros tres puntos de la lista de criterios de aceptación para esta historia:

- El inicio de sesión debe ser una dirección de correo electrónico válida.
- El sistema rechaza el intento de inicio de sesión con una contraseña incorrecta.
- El sistema rechaza el intento de inicio de sesión por un inicio de sesión de usuario inexistente.

se puede especificar de la siguiente manera:

Esquema del escenario: inicios de sesión y contraseñas correctos e incorrectos

El usuario dado ingresa <login> como inicio de sesión  
Y el usuario ingresa <contraseña> como contraseña  
Cuando el usuario hace clic en el botón "iniciar sesión"  
Entonces el resultado de inicio de sesión es <resultado>.

Ejemplos:

```
| iniciar sesión | contraseña | resultado |
| edina.monsoon@gmail.com | abFab | Aceptar |
| patsy.stone@gmail | martini | NO ESTÁ BIEN |
| azafrán@mail.abc.com | | NO ESTÁ BIEN |
| mods→london@gmail.com | balanceándose | NO ESTÁ BIEN |
| | algunaContraseña | NO ESTÁ BIEN |
```

En el ejemplo anterior, hay referencias a variables (entre paréntesis "<" y ">") en la descripción de los criterios de aceptación presentados en formato Dado/Cuándo/Entonces, y los datos de prueba específicos se encuentran en la sección Ejemplos a continuación. Esta prueba se ejecutará cinco veces, cada vez con un conjunto diferente de datos de prueba. El nombre de la sección donde los datos de la prueba se colocan sugiere que las pruebas están en forma de ejemplos, que muestran en datos específicos cómo se supone que debe comportarse el sistema. Por ejemplo, un inicio de sesión es Se supone que es correcto si el nombre de usuario y la contraseña cumplen con las condiciones establecidas (el nombre de usuario es un La dirección de correo electrónico, el nombre de usuario y la contraseña válidos no están vacíos y contienen únicamente

caracteres alfanuméricos). Por otro lado, si la dirección de correo electrónico no es válida (línea 2), la contraseña está en blanco (línea 3), se utilizan caracteres no alfanuméricos (línea 4) o la El inicio de sesión está en blanco (línea 5), el inicio de sesión no se realizará correctamente.

Ejemplo Consideremos otro ejemplo de una historia de usuario, esta vez para un CRM. sistema para un determinado banco. La historia se refiere a la implementación de ciertos negocios. reglas relacionadas con la oferta de tarjetas de crédito a clientes que cumplan ciertos requisitos.

Como institución financiera  
Quiero asegurarme de que sólo los clientes con ingresos anuales suficientes obtengan un tarjeta de crédito

Para que no se ofrezcan tarjetas de crédito a clientes que no podrán pagar el débito en la tarjeta

Escenario: Hay dos tipos de tarjetas: una con un límite de débito de \$2500, la otra con un límite de débito de \$5000. El límite máximo de la tarjeta de crédito depende del ganancias del cliente (redondeadas al dólar más cercano). El cliente debe tener un salario. de más de \$10,000 por mes para obtener el límite de débito más bajo. Si el salario excede \$15,000 por mes, el cliente obtiene un límite de débito más alto.

Dado un cliente con ganancias mensuales <monto de ganancias>.  
Cuando un cliente solicita una tarjeta de crédito  
Entonces la solicitud debe ser <aceptada> o <rechazada> y si es aceptada, el límite máximo de la tarjeta de crédito debe ser <límite máximo>.

En la Tabla 4.14 se muestran ejemplos de casos de prueba escritos basados en esta historia . Nota que en este ejemplo, el evaluador, mientras crea casos de prueba para verificar la regla descrita en la historia, al mismo tiempo intentó cubrir los valores límite del "salario mensual" dominio.

4.5.3 Desarrollo basado en pruebas de aceptación (ATDD)

El desarrollo basado en pruebas de aceptación (ATDD) es un enfoque que da prioridad a las pruebas (consulte Secta. 2.1.3). Los casos de prueba se crean antes de implementar la historia del usuario. Casos de prueba son creados por miembros del equipo con diferentes perspectivas sobre el producto, como clientes, desarrolladores y evaluadores [63]. Los casos de prueba pueden ser manuales o automatizados.

El primer paso es el llamado taller de especificación, durante el cual los miembros del equipo analizan, discuten y escriben la historia del usuario y sus criterios de aceptación. Durante esto proceso, todo tipo de problemas en la historia, como falta de completitud, ambigüedades,

Cuadro 4.14 Resultados empresariales

Ganancias de CT	Resultado esperado	Límite máximo	Comentarios	
1	\$10 000	Rechazado	\$0	Ingresos ≤ \$1000
2	\$10 001	Aceptado	\$2500	\$10,000 < ingreso ≤ \$15,000
3	\$15 000	Aceptado	\$2500	\$10,000 < ingreso ≤ \$15,000
4	\$15 001	Aceptado	\$5000	\$15,000 < ingresos

se solucionan contradicciones u otro tipo de defectos. El siguiente paso es crear pruebas.

Esto lo puede hacer colectivamente un equipo o individualmente un evaluador. En cualquier caso, una persona independiente, como un representante comercial, realiza la validación de las pruebas. Las pruebas son ejemplos, basados en criterios de aceptación, que describen características específicas de la historia del usuario (ver Sección 4.5.2). Estos ejemplos ayudan al equipo a implementar la historia del usuario correctamente. Debido a que los ejemplos y las pruebas son lo mismo, los términos a menudo se usan indistintamente.

Una vez diseñadas las pruebas, las técnicas de prueba descritas en las Secciones. Se pueden aplicar 4.2, 4.3 y 4.4. Normalmente, las primeras pruebas son positivas, confirman el comportamiento correcto sin excepciones ni errores, e implican una secuencia de acciones realizadas si todo sale como se esperaba. Se dice que este tipo de escenarios implementan los llamados caminos felices, es decir, caminos de ejecución donde todo va según lo planeado sin que se produzcan fallas. Después de ejecuciones de pruebas positivas, el equipo debe realizar pruebas negativas y pruebas relacionadas con los atributos no funcionales (por ejemplo, rendimiento, usabilidad). Las pruebas deben expresarse en términos que las partes interesadas puedan entender. Normalmente, las pruebas incluyen oraciones en lenguaje natural que contienen las condiciones previas necesarias (si las hay), entradas y salidas asociadas.

Los ejemplos (es decir, pruebas) deben cubrir todas las características de la historia del usuario y no deben ir más allá. Sin embargo, los criterios de aceptación pueden detallar algunos de los problemas descritos en la historia del usuario. Además, no deben haber dos ejemplos que describan las mismas características de la historia del usuario (es decir, las pruebas no deben ser redundantes).

Cuando las pruebas se escriben en un formato compatible con el marco de automatización de pruebas de aceptación, los desarrolladores pueden automatizar estas pruebas escribiendo código de soporte durante la implementación de la función descrita por la historia del usuario. Las pruebas de aceptación se convierten entonces en requisitos ejecutables. Un ejemplo de dicho código de soporte se muestra en la Sección. 2.1.3 al discutir el enfoque BDD.

Ejemplo Supongamos que un equipo tiene la intención de probar la siguiente historia de usuario.

Historia de usuario US-002-02

Como cliente bancario registrado

Quiero poder transferir dinero a otra cuenta

Para poder transferir fondos entre cuentas.

Criterios de aceptación

- (AC1) el monto de la transferencia no puede exceder el saldo de la cuenta • (AC2) la cuenta de destino debe ser correcta • (AC3) para datos válidos (monto, números de cuenta) el saldo de la cuenta de origen disminuye y el saldo de la cuenta de destino aumenta en el monto de la transferencia
- (AC4) el monto de la transferencia debe ser positivo y representar la cantidad correcta de dinero (es decir, tener una precisión máxima de dos decimales)

Teniendo en cuenta que las pruebas deben verificar que se cumplan los criterios de aceptación, los ejemplos de pruebas funcionales positivas aquí podrían ser las siguientes pruebas relacionadas con (AC3) y verificando también (AC4):

- TC1: realizar una transferencia "típica", por ejemplo, una transferencia por \$1500 desde una cuenta con un saldo de \$2735.45 a otra cuenta correcta; resultado esperado: saldo de la cuenta = \$1235.45, el saldo de la cuenta objetivo aumenta en \$1500
- TC2: realizar una transferencia correcta de todo el saldo de la cuenta a otra cuenta correcta (por ejemplo, una transferencia por \$21,37 de una cuenta con un saldo de \$21,37 a otra cuenta correcta); Resultado esperado: saldo de la cuenta = \$0, el saldo de la cuenta objetivo aumenta en \$21,37

El segundo caso de prueba utiliza el análisis de valores límite para la diferencia entre los saldo de la cuenta y el monto de la transferencia.

El siguiente paso es crear pruebas negativas, teniendo en cuenta que las pruebas son para verificar que se cumplan los criterios de aceptación. Por ejemplo, un evaluador podría considerar las siguientes situaciones:

- TC3: intentar realizar una transferencia a otra cuenta correcta, cuando el monto de la transferencia es mayor que el saldo de la cuenta de origen (verificación de AC1) • TC4: intentar realizar una transferencia con el monto y saldo de la transferencia correctos, pero a una cuenta inexistente (verificación de AC2)
- TC5: intentar realizar una transferencia con el monto y saldo de transferencia correctos a la misma cuenta (verificación de AC2)
- TC6: intento de realizar una transferencia por importe incorrecto (verificación de AC4)

Por supuesto, para cada una de estas pruebas, el evaluador puede definir una serie de datos de prueba para verificar el comportamiento específico del programa. Al crear estos casos de prueba, el evaluador puede utilizar técnicas de caja negra (por ejemplo, partición de equivalencia o análisis de valores límite). Por ejemplo, para TC1, vale la pena considerar transferencias por los siguientes montos: el mínimo posible (p. ej., \$0,01), "típico" (p. ej., \$1500), montos con diferentes niveles de precisión (p. ej., \$900.2, \$8321.06), o un cantidad muy grande (por ejemplo, \$8,438,483,784).

Para TC3, por otro lado, vale la pena comprobar las siguientes situaciones (aquí utilizamos la partición de equivalencia y el análisis de valores límite):

- Cuando el monto de la transferencia es mucho mayor que el saldo de la cuenta • Cuando el monto de la transferencia es mayor en 1 centavo (el incremento mínimo posible) que el saldo de la cuenta

Para TC4, una cuenta no válida se puede representar como:

- Una cadena de caracteres vacía • Un número con la estructura correcta (26 dígitos), pero que no corresponde a ninguna cuenta física • Un número con estructura incorrecta: demasiado corto (p. ej., 25 dígitos) • Un número con estructura incorrecta: demasiado largo (p. ej., 27 dígitos) • Un número con estructura no válida: contiene caracteres prohibidos, como letras

Finalmente, para TC6, vale la pena considerar en particular:

- Un número negativo que representa la cantidad correcta (por ejemplo, \$-150.25) • Número 0 • Una cadena de caracteres que contiene letras (por ejemplo, \$15B.20)

- Un número representado como resultado de una operación matemática (por ejemplo,  $\$15+\$20$ ) • Un número con más de dos decimales (por ejemplo,  $\$0,009$ )

Tenga en cuenta que cada uno de los conjuntos de datos de prueba proporcionados anteriormente verifica un riesgo potencial significativamente diferente que existe en el sistema que en los otros casos. Por tanto, las pruebas no son redundantes; no marcan "lo mismo".

## Preguntas de muestra

### Pregunta 4.1

(FL-4.1.1, K2)

Diseño de pruebas, implementación de pruebas y ejecución de pruebas basadas en la entrada del software.

El análisis de dominio es un ejemplo de:

- A. Técnica de prueba de caja blanca.
- B. Técnica de prueba de caja negra.
- C. Técnica de prueba basada en la experiencia.
- D. Técnica de prueba estática.

Elija una respuesta.

### Pregunta 4.2

(FL-4.1.1, K2)

¿Cuál es una característica común de técnicas como la partición de equivalencia, el estado?

¿Pruebas de transición o tablas de decisión?

- R. Al utilizar estas técnicas, las condiciones de prueba se derivan en función de la información sobre la estructura interna del software bajo prueba.
- B. Para proporcionar datos de prueba para casos de prueba diseñados en base a estas técnicas, el evaluador debe analizar el código fuente.
- C. La cobertura en estas técnicas se mide como la proporción de elementos del código fuente probados (por ejemplo, declaraciones) con respecto a todos los elementos identificados en el código.
- D. Los casos de prueba diseñados con estas técnicas pueden detectar discrepancias entre requisitos y su implementación real.

Elija una respuesta.

### Pregunta 4.3

(FL-4.2.1, K3)

El sistema asigna un descuento por compras dependiendo del monto de las compras expresado en \$, que es un número positivo con dos decimales. Las compras hasta el monto de \$99.99 no dan derecho a descuento. Las compras de \$100 a \$299.99 dan derecho a un 5% de descuento. Las compras superiores a \$299.99 tienen derecho a un 10% de descuento.

Indique el conjunto mínimo de valores (dados en \$) para lograr una técnica de partición del 100% de equivalencia.



- R. 0,01; 100,99; 500.
- B. 99,99; 100; 299,99; 300.
- C.1; 99; 299.
- D.0; 5; 10.

Elija una respuesta.

#### Pregunta 4.4

(FL-4.2.1, K3)

La máquina expendedora de café acepta monedas de 25 céntimos, 50 céntimos y 1 dólar. El café cuesta 75c. Después de insertar la primera moneda, la máquina espera hasta que la cantidad de monedas insertadas por el usuario sea igual o superior a 75c. Cuando esto sucede, se bloquea la ranura para monedas, se dispensa café y (si es necesario) se da cambio. Supongamos que la máquina siempre tiene suficientes números y denominaciones de monedas para dar cambio. Considere los siguientes escenarios de prueba:

Escenario 1:

Inserte monedas en orden: 25c, 25c, 25c.

Comportamiento esperado: la máquina expendedora dispensa café y no da cambio.

Escenario 2:

Inserte monedas en orden: 25c, 50c.

Comportamiento esperado: la máquina expendedora dispensa café y no da cambio.

Escenario 3:

Inserte una moneda de \$1.

Comportamiento esperado: la máquina expendedora dispensa café y cambio de 25 céntimos.

Escenario 4:

Inserte monedas en orden: 25c, 25c, 50c.

Comportamiento esperado: la máquina expendedora dispensa café y cambio de 25 céntimos.

Quiere comprobar si la máquina realmente da cambio cuando el usuario ha insertado monedas por más de 75 céntimos y si no da ningún cambio si el usuario ha insertado exactamente 75 céntimos.

¿Cuál de estos escenarios representa el conjunto mínimo de casos de prueba que logra este objetivo?

- A. Escenario 1, Escenario 2.
- B. Escenario 3, Escenario 4.
- C. Escenario 1, Escenario 3.
- D. Escenario 1, Escenario 2, Escenario 3.

Elija una respuesta.

#### Pregunta 4.5

(FL-4.2.1, K3)

Estás probando un sistema de gestión de tarjetas que te da derecho a descuentos en compras. Hay cuatro tipos de tarjetas, regulares, plateadas, doradas y diamantes, y tres posibles descuentos: 5%, 10% y 15%. Se utiliza la partición de equivalencia.

Técnica para comprobar si el sistema funciona correctamente para todo tipo de tarjetas y todos los descuentos posibles. Ya tienes preparados los siguientes casos de prueba:

TC1: tarjeta normal, descuento: 10%

TC2: tarjeta plata, descuento: 15%

TC3: tarjeta oro, descuento: 15%

TC4: tarjeta plata, descuento: 10%

¿Cuál es la MENOR cantidad de casos de prueba que debe preparar ADICIONALMENTE?

¿Cómo lograr una cobertura del 100% en "cada opción" tanto para tipos de tarjetas como para tipos de descuento?

A. 1

B. 3

C. 2

D. 8

Elija una respuesta.

Pregunta 4.6

(FL-4.2.2, K3)

El usuario define la contraseña ingresándola en el campo de texto y haciendo clic en el botón "Confirmar". De forma predeterminada, el campo de texto está en blanco al principio. El sistema considera correcto el formato de la contraseña si la misma tiene al menos 6 caracteres y no más de 11 caracteres. Identificó tres particiones de equivalencia: contraseña demasiado corta, longitud de contraseña correcta y contraseña demasiado larga. Los casos de prueba existentes representan un conjunto mínimo de casos de prueba que logran una cobertura BVA de 2 valores del 100 %. El director de pruebas decidió que, debido a la criticidad del componente sometido a prueba, sus pruebas deberían lograr una cobertura BVA de 3 valores del 100 %.

Contraseñas de qué longitudes deben probarse ADICIONALMENTE para lograr el cobertura requerida?

A. 5, 12 B.

0, 5, 12 C. 4,

7, 10 D. 1, 4,

7, 10, 13

Elija una respuesta.

Pregunta 4.7

(FL-4.2.2, K3)

Los usuarios del lavadero disponen de tarjetas electrónicas que registran cuántas veces han utilizado el lavadero hasta el momento. El lavado de coches ofrece una promoción: cada décimo lavado es gratuito. Está probando la corrección de ofrecer la promoción utilizando una técnica de análisis de valor límite.

¿Cuál es el conjunto MÍNIMO de casos de prueba que logra una cobertura BVA del 100% de 2 valores? Los valores en las respuestas indican el número del lavado dado (actual).

Tabla 4.15 Tabla de decisiones  
para reglas de asignación de tarifa gratuita

	R1	R2	R3
Condiciones			
¿Miembro del Parlamento?	Sí	-	-
¿Desactivado?	-	Sí	-
¿Alumno?	-	-	Sí
Acción			
¿Paseo libre?	Sí	Sí	NO

- R. 9, 10  
B.1, 9, 10  
C.1, 9, 10, 11  
D. No se puede lograr la cobertura requerida

Elija una respuesta.

Pregunta 4.8  
(FL-4.2.3, K3)

El analista de negocios preparó una tabla de decisión minimizada (Tabla 4.15) para describir las reglas comerciales para conceder viajes gratuitos en autobús. Sin embargo, la tabla de decisiones es defectuosa.

¿Cuál de los siguientes casos de prueba, que representan combinaciones de condiciones, muestra que las reglas de negocio en esta tabla de decisiones son CONTRADICTORIAS?

- A. Miembro del parlamento = Sí, discapacitado = NO, estudiante = Sí.  
B. Miembro del parlamento = Sí, discapacitado = Sí, estudiante = NO.  
C. Miembro del parlamento = NO, discapacitado = NO, estudiante = Sí.  
D. Miembro del parlamento = NO, discapacitado = NO, estudiante = NO.

Elija una respuesta.

Pregunta 4.9  
(FL-4.2.3, K3)

Crea una tabla de decisiones completa que tiene las siguientes condiciones y acciones:

- Condición "edad": valores posibles: (1) hasta 18; (2) 19–40; (3) 41 o más.
- Condición "lugar de residencia"—valores posibles: (1) ciudad; (2) pueblo.
- Condición "salario mensual"—valores posibles: (1) hasta \$4000; (2) \$4001 o más.
- Acción "otorgar crédito"—valores posibles: (1) Sí; (2) NO.
- Acción "ofrecer seguro de crédito"—valores posibles: (1) Sí; (2) NO.

¿Cuántas columnas tendrá la tabla de decisión completa para este problema?

- R.6  
B.11  
C.12  
D.48

Elija una respuesta.

Tabla 4.16 Transiciones válidas del sistema para el inicio de sesión proceso

Transición	Estado	Evento	siguiente estado
1	Inicial	Acceso	Inicio sesión
2	Inicio sesión	Iniciar sesiónOK	registrado
3	Inicio sesión	Error de inicio de sesión	Inicial
4	registrado	Cerrar sesión	Inicial

Pregunta 4.10  
(FL-4.2.4, K3)

La Tabla 4.16 muestra en sus filas TODAS las transiciones válidas entre estados en el sistema para manejar el proceso de inicio de sesión. El sistema contiene tres estados (Inicial, Registro, Registrado) y cuatro posibles eventos (Iniciar sesión, Iniciar sesión OK, Iniciar sesiónError, Cerrar sesión).  
¿Cuántas transiciones INVÁLIDAS hay en este sistema?

- R.8
- b.0
- C.4
- D.12

Elija una respuesta.

Pregunta 4.11  
(FL-4.2.4, K3)

La figura 4.17 muestra un diagrama de transición de estado para un determinado sistema.  
¿Cuál es el número MÍNIMO de casos de prueba que alcanzarán el 100% de validez?  
cobertura de transiciones?

- R.2
- B.3
- C.6
- D.7

Elija una respuesta.

Pregunta 4.12  
(FL-4.3.1, K2)

Después de la ejecución de la prueba, sus casos de prueba lograron una cobertura de declaración del 100%. cual de  
¿Las siguientes afirmaciones describen la consecuencia correcta de este hecho?

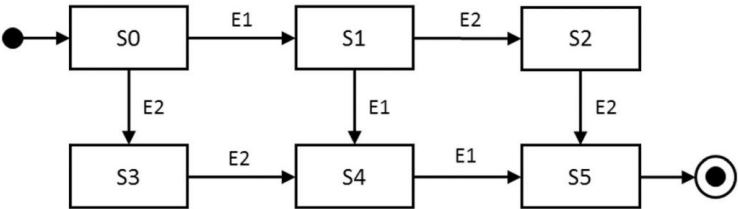


Fig. 4.17 Diagrama de transición de una máquina de estados

- R. Se logra una cobertura del 100% en sucursales.
- B. Se ejerció cada valor lógico de cada decisión en el código.
- C. Se aplicaron todos los resultados posibles que el programa bajo prueba puede devolver.
- D. Se ejerció cada declaración que contenía un defecto.

Elija una respuesta.

Pregunta 4.13

(FL-4.3.2, K2)

Considere un programa simple de tres declaraciones (suponga que las declaraciones 1 y 2 toman números enteros no negativos (0, 1, 2, etc.) y que estas declaraciones siempre se ejecutarán correctamente):

1. Obtener el valor de entrada x 2.
- Obtener el valor de entrada y 3.
- Devolver  $x + y$

¿Cuántos casos de prueba se necesitan para lograr una cobertura de sucursal del 100% en este código y por qué?

- R. Se necesitan dos casos de prueba, porque uno debe probar la situación en la que el valor devuelto es 0 y el otro debe probar la situación en la que el valor devuelto es un número positivo.
- B. No es necesario ejecutar ninguna prueba, porque la cobertura de bifurcaciones para este código se cumple por definición, ya que el programa consta de un pasaje secuencial de tres declaraciones y no hay bifurcaciones condicionales para probar.
- C. Se necesita un caso de prueba con valores de entrada arbitrarios x e y, porque cada prueba dará como resultado la ejecución de la misma ruta que cubre todas las ramas.
- D. Es imposible lograr cobertura de rama con un número finito de casos de prueba, porque para hacerlo, tendríamos que forzar todos los valores posibles de la suma de  $x+y$  en el enunciado 3, lo cual no es factible.

Elija una respuesta.

Pregunta 4.14

FL-4.3.3 (K2)

¿Cuál de las siguientes opciones describe MEJOR el beneficio de utilizar técnicas de prueba de caja blanca?

- A. La capacidad de detectar defectos cuando las especificaciones están incompletas.
- B. La capacidad de los desarrolladores para realizar pruebas, ya que estas técnicas requieren programación.  
habilidades de minería.
- C. Asegurarse de que las pruebas alcancen una cobertura del 100% de cualquier técnica de caja negra, ya que esto es implica lograr una cobertura de código del 100%.
- D. Mejor control del nivel de riesgo residual en el código, ya que está directamente relacionado con medidas como cobertura de estados de cuenta y cobertura de sucursales.

Elija una respuesta.

## Pregunta 4.15

(FL-4.4.1, K2)

El evaluador utiliza el siguiente documento para diseñar los casos de prueba:

1. Ocurrencia de un error aritmético provocado al dividir por cero.
2. Ocurrencia de error de redondeo.
3. Forzar un resultado de valor negativo.

¿Qué técnica utiliza el probador?

- A. Análisis de valores en la frontera.
- B. Pruebas basadas en listas de verificación.
- C. Pruebas basadas en casos de uso.
- D. Error al adivinar

Elija una respuesta.

## Pregunta 4.16

(FL-4.4.2, K2)

¿Cuál de las siguientes es la oración correcta con respecto al uso de técnicas de prueba formales (es decir, de caja negra y de caja blanca) como parte de una prueba exploratoria basada en sesiones?

- R. Se permiten todas las técnicas de prueba formales, porque las pruebas exploratorias no imponer ningún método de operación específico.
- B. Todas las técnicas de prueba formales están prohibidas, porque las pruebas exploratorias se basan en el conocimiento, las habilidades, la intuición y la experiencia del evaluador.
- C. Todas las técnicas de prueba formales están prohibidas, porque los pasos realizados en la exploración Las pruebas teóricas no están planificadas con antelación.
- D. Se permiten todas las técnicas de prueba formales, porque un evaluador exploratorio necesita una base de prueba de la cual derivar casos de prueba.

Elija una respuesta.

## Pregunta 4.17

(FL-4.4.3, K2)

¿Qué beneficio se puede lograr mediante el uso de pruebas basadas en listas de verificación?

- A. Valoración de las pruebas no funcionales, que a menudo se subestiman.
- B. Permitir una medición precisa de la cobertura del código.
- C. Aprovechar la experiencia del evaluador.
- D. Consistencia de prueba mejorada.

Elija una respuesta.

## Pregunta 4.18

(FL-4.5.1, K2)

Durante una reunión de planificación de iteraciones, el equipo comparte ideas sobre la historia del usuario. El propietario del producto quiere que el cliente tenga un formulario de pantalla única para ingresar información. El desarrollador explica que esta característica tiene algunas

limitaciones técnicas, relacionadas con la cantidad de información que se puede capturar en el pantalla.

¿Cuál de las siguientes opciones representa MEJOR la continuación de la escritura de esta historia de usuario?

- R. El evaluador decide que el formulario debe caber en la pantalla y describe esto como uno de los criterios de aceptación de la historia, ya que será él quien realizará las pruebas de aceptación más adelante.
- B. El evaluador escucha los puntos de vista del propietario del producto y del desarrollador y crea dos pruebas de aceptación para cada una de las dos soluciones propuestas.
- C. El evaluador informa al desarrollador que los criterios de aceptación del desempeño deben basarse en un estándar: un máximo de 1 segundo por registro de datos. El desarrollador describe esto como uno de los criterios de aceptación, ya que el desarrollador es responsable del rendimiento de la aplicación.
- D. El evaluador negocia con el desarrollador y el propietario del producto la cantidad de información de entrada necesaria y juntos deciden reducir esta información a la más importante para que quepa en la pantalla.

Elija una respuesta.

Pregunta 4.19

(FL-4.5.2, K2)

Considere la siguiente historia de usuario:

Como cliente potencial de la tienda electrónica,  
quiero poder registrarme completando el formulario de registro, para poder utilizar  
todas las funciones de la tienda electrónica.

¿Cuáles DOS de los siguientes son ejemplos de criterios de aceptación comprobables para esta historia?

- R. El proceso de registro debe realizarse con la suficiente rapidez.
- B. Después del registro, el usuario tiene acceso a la función "pedidos a domicilio".
- C. El sistema rechaza el registro si el usuario ingresa como inicio de sesión un correo electrónico ya existente en la base de datos.
- D. El operador del sistema puede enviar el pedido realizado por el usuario para su procesamiento.
- E. Los criterios de aceptación para esta historia de usuario deben estar en formato Dado/Cuando/Entonces.

Seleccione DOS respuestas.

Pregunta 4.20

(FL-4.5.3, K3)

La regla comercial para el préstamo de libros en el sistema de bibliotecas universitarias establece que un lector puede pedir prestados libros nuevos si se cumplen las dos condiciones siguientes:

- El tiempo de conservación del libro más antiguo no supera los 30 días.
- Después del préstamo, el número total de libros prestados no excederá los cinco libros.  
para un estudiante y diez libros para un profesor.

El equipo debe implementar una historia de usuario para este requisito. La historia se crea, utilizando el marco ATDD, y los siguientes criterios de aceptación están escritos como Ejemplos en el formato Dado/Cuándo/Entonces:

Dado que <UserType> ya ha tomado prestados <Number> de libros  
Y el tiempo del libro retenido por más tiempo es <Días> días.  
Cuando el usuario quiere pedir prestado <Solicitar> libros nuevos  
Luego el sistema <Decisión> de prestar los libros.

Ejemplos:

No	Tipo de usuario	Número	Días	Solicitar	Decisión
1	Estudiante	3	30	2	no permite
2	Estudiante	4	1	1	permite
3	Profesor	6	32	3	no permite
4	Profesor	0	0	6	permite

¿Cuántos de los cuatro casos de prueba anteriores están definidos INCORRECTAMENTE, es decir, ¿violar las reglas comerciales del préstamo de libros?

- R. Ninguno: todos siguen la regla empresarial.
- Hueso.
- C. Dos.
- D. Tres.

Ejercicios

Ejercicio 4.1

(FL-4.2.1, K3)

El usuario rellena un formulario web para adquirir entradas para conciertos. Los boletos están disponibles para tres conciertos: Iron Maiden, Judas Priest y Black Sabbath. Para cada uno de los conciertos se pueden adquirir dos tipos de entradas: un sector frente al escenario y un sector alejado del escenario. El usuario confirma la elección marcando el cuadros en las listas desplegadas correspondientes (una de las cuales contiene los nombres de los bandas, el otro, el tipo de entrada). Sólo se puede adquirir una entrada para una sola banda. comprado por sesión.

Quiere comprobar la corrección del sistema para cada banda y, independientemente dentemente, para el tipo de billete.

- A) Identificar los dominios y sus particiones de equivalencia.
- B) ¿Hay particiones no válidas? Justifica tu respuesta.
- C) Diseñar el conjunto más pequeño posible de casos de prueba que alcancen el 100% de equivalencia. Cobertura de partición.

Ejercicio 4.2

(FL-4.2.1, K3)



Tabla 4.17 Reglas de descuento

Cantidad	Descuento concedido
Hasta \$300	No
Más de \$300, hasta \$800	5%
Más de \$800	10%

Un número natural mayor que 1 se llama primo si es divisible sólo por dos números: por 1 y por sí mismo. El sistema toma un número natural (ingresado por el usuario en el campo del formulario) como entrada y devuelve si es primo o no. El campo del formulario para los datos de entrada tiene un mecanismo de validación y no permitirá ingresar ninguna cadena que no represente una entrada válida (es decir, un número natural mayor que uno).

- A) Identificar el dominio y su partición de equivalencia.
- B) ¿Hay particiones no válidas en el problema? Justifica tu respuesta.
- C) Diseñe el conjunto más pequeño posible de casos de prueba que logren una cobertura de partición de equivalencia del 100%.

Ejercicio 4.3

(FL-4.2.2, K3)

Estás probando la funcionalidad de pago de la tienda electrónica. El sistema recibe una cantidad positiva de compras (en \$ con una precisión de 1 centavo). Luego, este monto se redondea al número entero más cercano y, en función de este valor redondeado, se calcula un descuento de acuerdo con las reglas descritas en la Tabla 4.17: Desea aplicar un BVA de 2 valores para verificar la exactitud del cálculo del descuento. Los datos de entrada en el caso de prueba son la cantidad antes del redondeo. Realice particiones de equivalencia, determine los valores límite y diseñe los casos de prueba.

Ejercicio 4.4

(FL-4.2.2, K3)

El sistema calcula el precio del marco del cuadro basándose en los parámetros dados: ancho y alto del cuadro (en centímetros). El ancho correcto del cuadro está entre 30 y 100 cm inclusive. La altura correcta del cuadro está entre 30 y 60 cm inclusive.

El sistema toma ambos valores de entrada a través de una interfaz que acepta sólo valores válidos. valores de los rangos especificados anteriormente.

El sistema calcula el área de la imagen como el producto del ancho y el alto. Si el precio del marco es La superficie supera los 1600 cm<sup>2</sup> El precio, \$500. De lo contrario, el encuadre es de \$450.

Aplique BVA de 2 valores al problema anterior:

- A) Identificar las particiones y sus valores límite.
- B) Diseñe el menor número posible de casos de prueba que cubran los valores límite para todos los parámetros relevantes. ¿Es posible lograr una cobertura total de BVA de 2 valores? Justifica tu respuesta.

Ejercicio 4.5

(FL-4.2.3, K3)

El operador del sistema de soporte para el examen de licencia de conducir ingresa la siguiente información en el sistema para un candidato que realiza los exámenes por primera vez:

- El número de puntos del examen teórico (número entero de 0 a 100).
- El número de errores cometidos por el candidato durante el examen práctico (número entero número 0 o mayor).

El candidato deberá realizar ambos exámenes. A un candidato se le concede una licencia de conducir si cumple las dos condiciones siguientes: obtuvo al menos 85 puntos en el examen teórico y no cometió más de dos errores en el examen práctico. Si un candidato no aprueba uno de los exámenes, deberá repetirlo. Además, si el candidato no aprueba ambos exámenes, deberá realizar horas adicionales de lecciones de conducción.

Utilice pruebas de tablas de decisión para llevar a cabo el proceso de diseño de casos de prueba para el problema anterior. Siga el procedimiento de cinco pasos descrito en la Sección. 4.2.3, es decir:

- Identifique todas las condiciones posibles y enumérelas en la parte superior de la tabla.
- Identifique todas las posibles acciones que pueden ocurrir en el sistema y enumérelas en la parte inferior de la tabla.
- Generar todas las combinaciones de condiciones. Elimine las combinaciones no factibles (si las hay) y enumere todas las combinaciones factibles restantes en columnas individuales en la parte superior de la matriz.
- Para cada combinación de condiciones así identificadas, determine, con base en la especificación, qué acciones deben realizarse en el sistema e introdúzcalas en las columnas correspondientes en la parte inferior de la tabla.
- Para cada columna, diseñe un caso de prueba que incluya un nombre (que describa lo que prueba el caso de prueba), condiciones previas, datos de entrada, resultados esperados y condiciones posteriores.

#### Ejercicio 4.6

(FL-4.2.3, K3)

La Figura 4.18 (según Graham et al., Foundations of Software Testing) muestra el proceso de asignación de asientos a los pasajeros en función de si tienen una tarjeta dorada y si hay asientos disponibles en las clases ejecutiva y económica, respectivamente.

Describe este proceso con una tabla de decisiones. ¿Notaste algún problema con el especificación al crear la tabla? Si es así, sugiera una solución al problema.

#### Ejercicio 4.7

(FL-4.2.4, K3)

El cajero automático se encuentra inicialmente en estado de espera (pantalla de bienvenida). Después de que el usuario inserta la tarjeta, se lleva a cabo la validación de la tarjeta. Si la tarjeta no es válida, el sistema la devuelve y finaliza con el mensaje "Error de tarjeta". De lo contrario, el sistema solicita al usuario que ingrese un PIN. Si el usuario proporciona un PIN válido, el sistema cambia al estado "Registrado" y finaliza la operación. Si el usuario ingresa un PIN incorrecto, el sistema solicita ingresarlo nuevamente. Si el usuario ingresa el PIN incorrecto tres veces, la tarjeta se bloquea, el usuario recibe el mensaje "Tarjeta bloqueada" y el sistema pasa al estado final que representa la tarjeta bloqueada.

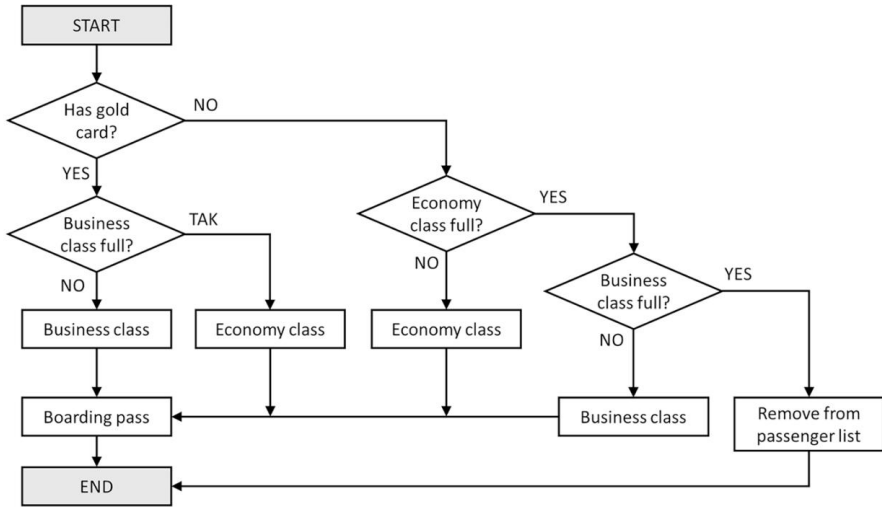


Fig. 4.18 El proceso de asignación de asientos en un avión.

- Identificar posibles estados, eventos y acciones, y diseñar un diagrama de transición de estados. para el escenario anterior sin utilizar condiciones de guardia.
- Diseñar un diagrama de transición de estado para el mismo escenario pero usando condiciones de guardia. Debería obtener un modelo con menos estados.
- Para el diagrama de transición de estado de (A), diseñe el menor número posible de casos de prueba que logren:

- Cobertura del 100% en todos los estados.
- Cobertura de transiciones 100% válida.

#### Ejercicio 4.8

(FL-4.2.4, K3)

El funcionamiento del perro robot mecánico se describe mediante la transición de estado. diagrama que se muestra en la Fig. 4.19. El estado inicial es "S".

- ¿Cuántas transiciones no válidas hay en este diagrama?
- Diseñar casos de prueba que logren una cobertura total de todas las transiciones. Adopte la regla de que al probar transiciones no válidas, un caso de prueba prueba solo una transición no válida.

#### Ejercicio 4.9

(FL-4.5.3, K3)

Como tester, comienzas a trabajar en el diseño de la prueba de aceptación para la siguiente historia de usuario:

US-01-002 Registro de nuevo usuario Como:

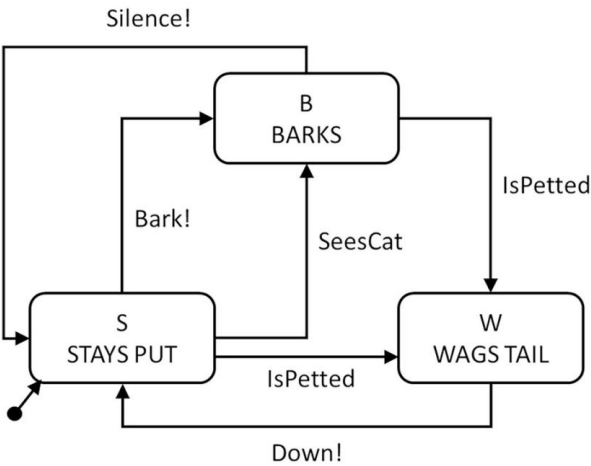
Estimación del esfuerzo: 3

cualquier usuario—cliente potencial Quiero:

poder completar el formulario de registro web Para que: Soy poder

utilizar las ofertas del proveedor de servicios Criterios de aceptación:

Fig. 4.19 Diagrama de transición de estado del perro robot



- AC1 El inicio de sesión del usuario es una dirección de correo electrónico válida.
- El usuario AC2 no puede registrarse utilizando un inicio de sesión existente
- AC3 Por motivos de seguridad, el usuario deberá introducir la contraseña en dos campos independientes, y dichas contraseñas deben ser idénticas; el sistema no permite al usuario pegar una cadena en estos campos; debe ingresarse manualmente
- La contraseña AC4 debe contener al menos 6 y como máximo 12 caracteres, al menos 1 número y al menos 1 letra mayúscula
- AC5 El registro exitoso da como resultado el envío de un correo electrónico que contiene un enlace, el cual al hacer clic se confirma el registro y se activa la cuenta del usuario.
- Proponer algunas pruebas de aceptación de muestra con datos de prueba específicos y el comportamiento esperado del sistema.

Capítulo 5 Gestión de las actividades de prueba



Palabras clave

Gestión de defectos	El proceso de reconocer, registrar, clasificar, investigar, resolver y eliminar defectos
Informe de defectos	Documentación de la ocurrencia, naturaleza y estado de un defecto. Sinónimos: informe de error
Criterio para entrar	El conjunto de condiciones para iniciar oficialmente una tarea definida. Referencias: Gilb y Graham
Criterio de salida	El conjunto de condiciones para completar oficialmente un determinado tarea. Después de Gilb y Graham. Sinónimos: finalización de la prueba criterios, criterios de finalización
Riesgo del producto	Un riesgo que afecta la calidad de un producto.
Riesgo del proyecto	Un riesgo que impacta el éxito del proyecto
Riesgo	Un factor que podría resultar en futuros negativos. consecuencias
Análisis de riesgo	El proceso general de identificación de riesgos y riesgos. evaluación
Evaluación de riesgos	El proceso para examinar los riesgos identificados y determinar el nivel de riesgo
Control de riesgo	El proceso general de mitigación de riesgos y riesgos. supervisión
Identificación de riesgo	El proceso de encontrar, reconocer y describir. riesgos. Referencias: ISO 31000
Nivel de riesgo	La medida de un riesgo definida por el impacto y la probabilidad. Sinónimos: exposición al riesgo
Gestión de riesgos	El proceso de gestión de riesgos. Después de ISO 24765
Mitigación de riesgos	El proceso mediante el cual se toman decisiones y Se implementan medidas de protección para reducir o mantener los riesgos a niveles específicos

Monitoreo de riesgos	La actividad que verifica e informa el estado de los conocidos. riesgos para las partes interesadas
Pruebas basadas en riesgos	Pruebas en las que la gestión, selección, priorización y uso de actividades y recursos de prueba se basan en los tipos y niveles de riesgo correspondientes. Después de ISO 29119-1
Enfoque de prueba	La forma de implementar las tareas de prueba.
Informe de finalización de la prueba	Un tipo de informe de prueba elaborado en los hitos de finalización que proporciona una evaluación de la prueba correspondiente elementos según los criterios de salida. Sinónimos: resumen de la prueba informe
Control de prueba	La actividad que desarrolla y aplica acciones correctivas. para encaminar un proyecto de prueba cuando se desvía de lo que fue planeado
Monitoreo de pruebas	La actividad que verifica el estado de las actividades de prueba. identifica cualquier variación respecto de lo planificado o esperado, y informa el estado a las partes interesadas
Plan de prueba	Documentación que describa los objetivos de la prueba que se deben conseguido y los medios y el calendario para lograrlo. ellos, organizados para coordinar las actividades de prueba. Referencias: ISO 29119-1
Planificación de pruebas	La actividad de establecer o actualizar un plan de pruebas.
Informe de progreso de la prueba	Un tipo de informe de prueba periódico que incluye el progreso de las actividades de prueba frente a una línea de base, riesgos y alternativas requiriendo una decisión. Sinónimos: informe de estado de prueba
Pirámide de prueba	Un modelo gráfico que representa la relación de la cantidad de pruebas por nivel, con más en la parte inferior que en la cima
Cuadrantes de prueba	Un modelo de clasificación de tipos de pruebas/niveles de pruebas en cuatro cuadrantes, relacionándolos con dos dimensiones de la prueba. objetivos: apoyar al equipo de producto versus criticar el producto y la tecnología frente a frente a los negocios

## 5.1 Planificación de pruebas

FL-5.1.1 (K2) Ejemplificar el propósito y contenido de un plan de prueba

FL-5.1.2 (K1) Reconocer cómo un evaluador agrega valor a la iteración y la planificación de lanzamientos

FL-5.1.3 (K2) Comparar y contrastar criterios de entrada y criterios de salida

FL-5.1.4 (K3) Usar técnicas de estimación para calcular el esfuerzo de prueba requerido

FL-5.1.5 (K3) Aplicar la priorización de casos de prueba FL-5.1.6

(K1) Recordar los conceptos de la pirámide de pruebas FL-5.1.7 (K2) Resumir los

cuadrantes de prueba y sus relaciones con los niveles y tipos de prueba

## 5.1.1 Propósito y contenido de un plan de prueba

El plan de pruebas es un documento que proporciona una descripción detallada de los objetivos del proyecto de prueba, los medios necesarios para lograr esos objetivos y un cronograma de actividades de prueba. En proyectos típicos, generalmente se crea un único plan de prueba, a veces llamado plan maestro de prueba o plan de prueba del proyecto. Sin embargo, en proyectos más grandes, puede encontrar varios planes, como un plan maestro de pruebas y planes correspondientes a los niveles de prueba definidos en el proyecto (plan de pruebas de nivel o plan de pruebas de fases). En tal situación, puede haber, por ejemplo, un plan de pruebas de integración de componentes, un plan de pruebas del sistema, un plan de pruebas de aceptación, etc. Los planes de pruebas detallados también pueden estar relacionados con los tipos de pruebas que se planean llevar a cabo en el proyecto (por ejemplo, un plan de pruebas de rendimiento).

El plan de prueba describe el enfoque de la prueba y ayuda al equipo a asegurarse de que se puedan iniciar las actividades de prueba y, cuando se completen, que se hayan realizado correctamente. Esto se logra definiendo criterios específicos de entrada y salida (ver Sección 5.1.3) para cada actividad de prueba o actividad en el plan de prueba. El plan de pruebas también confirma que, si se sigue, las pruebas se realizarán de acuerdo con la estrategia de pruebas del proyecto y la política de pruebas de la organización.

Muy raramente el proyecto saldrá 100% como lo planeamos. En la mayoría de las situaciones, serán necesarias modificaciones menores o mayores. Entonces surge la pregunta natural: ¿por qué perder el tiempo planificando en un caso así? Dwight Eisenhower, general del ejército estadounidense y más tarde presidente de los Estados Unidos, dijo una vez: "Al prepararme para la batalla, siempre me he dado cuenta de que los planes son inútiles, pero la planificación es indispensable".<sup>1</sup>

De hecho, la parte más valiosa de la creación de un plan de pruebas es el proceso de planificación de pruebas en sí.

En cierto sentido, la planificación "obliga" a los evaluadores a centrar su pensamiento en los desafíos y riesgos futuros relacionados con el cronograma, los recursos, las personas, las herramientas, el costo, el esfuerzo, etc.

Durante la planificación, los evaluadores pueden identificar riesgos y pensar en el enfoque de prueba que será más efectivo. Sin planificación, los evaluadores no estarían preparados para los diferentes eventos indeseables que sucederán durante el proyecto. Esto, a su vez, crearía un riesgo muy alto de fracaso del proyecto, es decir, no completarlo o hacerlo tarde, por encima del presupuesto o con un alcance reducido del trabajo completado.

<sup>1</sup> [https://pl.wikiquote.org/wiki/Dwight\\_Eisenhower](https://pl.wikiquote.org/wiki/Dwight_Eisenhower)

El proceso de planificación está influenciado por muchos factores que los evaluadores deben considerar para planificar mejor todas las actividades del proceso de prueba. Entre estos factores, se deben considerar en particular los siguientes:

- Política de pruebas y estrategia de pruebas •

SDLC • Alcance de las

pruebas • Objetivos

- Riesgos •

Limitaciones •

Criticidad •

Capacidad de

prueba • Disponibilidad de recursos

Un plan de prueba típico incluye la siguiente información: Contexto de la prueba. Alcance, objetivos, limitaciones e información de la base de la prueba Supuestos y limitaciones del proyecto de prueba Partes interesadas. Roles, responsabilidades, influencia en el proceso de prueba (por ejemplo, poder versus interés) y necesidades de contratación y capacitación de personas. Comunicación. Tipos y frecuencia de comunicación y plantillas de documentos utilizados.

Lista de riesgos. Riesgos del producto y riesgos del proyecto (ver Sección 5.2)

Enfoques para las pruebas. Niveles de prueba (ver Sección 2.2.1), tipos de prueba (ver Sección 2.2.2), técnicas de prueba (ver Capítulo 4), productos del trabajo de prueba, criterios de entrada y salida (ver Sección 5.1.3), nivel de independencia de la prueba (ver Sección 1.5.3), definición de las métricas de prueba utilizadas (ver Sección 5.3.1), requisitos de datos de prueba, requisitos del entorno de prueba y desviaciones de las mejores prácticas organizacionales (con justificación)

Horario (ver Apartado 5.1.5)

A medida que se realiza el proyecto y se implementa el plan de prueba, aparece información adicional que detalla el plan. Por tanto, la planificación de pruebas es una actividad continua y se realiza durante todo el ciclo de vida del producto; a veces incluye una fase de mantenimiento. Es importante darse cuenta de que el plan de prueba inicial se actualizará, ya que la retroalimentación de las actividades de prueba se utilizará para identificar riesgos cambiantes y ajustar los planes en consecuencia.

Los resultados del proceso de planificación se pueden documentar en un plan maestro de pruebas y en Planes separados para niveles de prueba y tipos de prueba específicos.

Los siguientes pasos se realizan durante la planificación de la prueba (ver Fig. 5.1):

- Definir el alcance y los objetivos de las pruebas y los riesgos asociados • Determinar el enfoque general de las pruebas • Integrar y coordinar las actividades de prueba dentro de las actividades del SDLC • Decidir qué probar, qué personal y otros recursos se necesitarán para realizar las diversas actividades de prueba y cómo se deben realizar las actividades



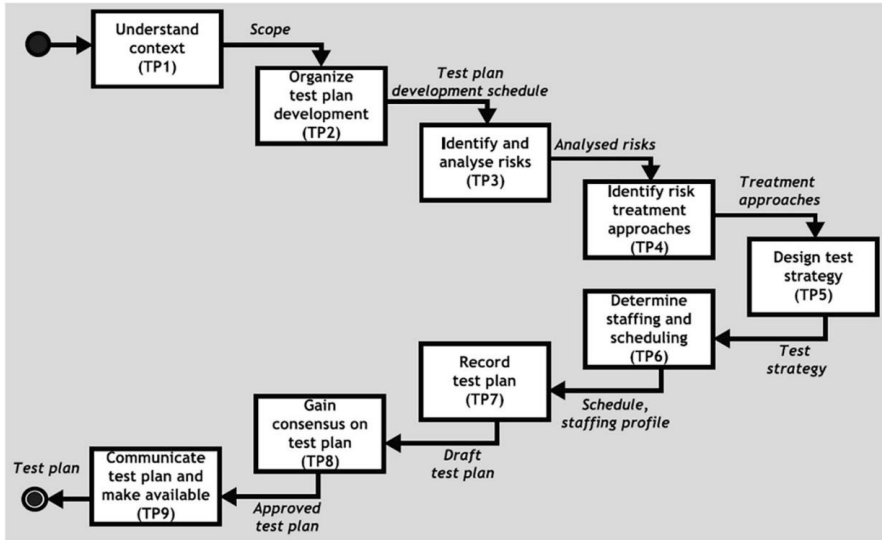


Fig. 5.1 Proceso de planificación de pruebas (según norma ISO/IEC/IEEE 29119-2)

- Planificar las actividades de análisis, diseño, implementación, ejecución y evaluación de pruebas estableciendo plazos específicos (enfoque secuencial) y colocando estas actividades en el contexto de iteraciones individuales (enfoque iterativo)
- Hacer una selección de medidas para el seguimiento y control de las pruebas
- Determinar el presupuesto para la recopilación y evaluación de métricas.
- Determinar el nivel de detalle y estructura de la documentación (plantillas o documentos de muestra).

Una parte del plan de prueba (el enfoque de prueba) será en la práctica la implementación de una estrategia de prueba específica vigente en la organización o proyecto. La estrategia de prueba es una descripción general del proceso de prueba que se está implementando, generalmente a nivel de producto u organización.

#### Estrategias de prueba

Las estrategias de prueba típicas son:

- Estrategia analítica. Esta estrategia se basa en el análisis de un factor específico (por ejemplo, requisitos o riesgo). Un ejemplo de enfoque analítico son las pruebas basadas en riesgos, en las que el punto de partida para el diseño y la priorización de las pruebas es el nivel de riesgo.
- Estrategia basada en modelos. Con esta estrategia, las pruebas se diseñan sobre la base de un modelo de un aspecto específico requerido del producto, por ejemplo, función, proceso de negocio, estructura interna o características no funcionales (como

(continuado)

como confiabilidad). Se pueden crear modelos basados en modelos de procesos de negocio, modelos de estado, modelos de crecimiento de confiabilidad, etc.

- **Estrategia metódica.** La base de esta estrategia es la aplicación sistemática de un conjunto predeterminado de pruebas o condiciones de prueba, por ejemplo, un conjunto estándar de pruebas o una lista de verificación que contiene tipos de fallas típicas o probables. Según este enfoque, se realizan, entre otras cosas, pruebas basadas en listas de verificación, ataques de fallas y pruebas basadas en características de calidad.
- **Estrategia compatible con el proceso (o compatible con el estándar).** Esta estrategia implica la creación de casos de prueba basados en reglas y estándares externos (derivados, por ejemplo, de estándares de la industria), documentación de procesos o identificación y uso riguroso de la base de prueba, así como cualquier proceso o estándar impuesto por la organización.
- **Estrategia dirigida (o consultiva).** Esta estrategia se basa principalmente en el asesoramiento y la orientación de las partes interesadas, expertos en la materia o expertos técnicos, incluidos aquellos ajenos al equipo de prueba o a la organización.
- **Estrategia adversa a la regresión.** Esta estrategia, motivada por el deseo de evitar la regresión de funcionalidades ya existentes, prevé la reutilización del software de prueba heredado (especialmente casos de prueba), la automatización extensiva de las pruebas de regresión y el uso de conjuntos de pruebas estándar.
- **Estrategia reactiva.** Con esta estrategia, las pruebas están más orientadas a reaccionar ante eventos que a seguir un plan predeterminado (como ocurre con las estrategias descritas anteriormente), y las pruebas se diseñan y pueden ejecutarse inmediatamente en función del conocimiento adquirido a partir de los resultados de pruebas anteriores.

Un ejemplo de este enfoque son las pruebas exploratorias, en las que las pruebas se ejecutan y evalúan en respuesta al comportamiento del software bajo prueba.

En la práctica, se pueden e incluso se deben combinar diferentes estrategias. Los conceptos básicos de la selección de estrategias de prueba incluyen:

- **Riesgo de fracaso del proyecto:**
  - Riesgos para el producto
  - Peligro para las personas, el medio ambiente o la empresa causado por fallos del producto
  - Falta de habilidades y experiencia de las personas en el proyecto.
- **Regulaciones (externas e internas) sobre el proceso de desarrollo**
- **Propósito de la empresa de prueba**
- **Misión del equipo de prueba**
- **Tipo y características específicas del producto**

### 5.1.2 Contribución del evaluador a la planificación de iteraciones y lanzamientos

En los enfoques iterativos para el desarrollo de software, existen dos tipos de planificación relacionada con los productos: planificación de lanzamiento y planificación de iteraciones. Existen otros tipos de planificación, a nivel superior, organizacional o estratégico (como la planificación de “gran sala” o la planificación de “incremento de producto”). Estos tipos de planificación no se analizan más.

#### Planificación de

**lanzamientos** La planificación de lanzamientos anticipa el lanzamiento de un producto. La planificación del lanzamiento define y redefine el trabajo pendiente del producto y puede incluir refinar historias de usuarios más grandes en una colección de historias más pequeñas. La planificación de lanzamiento proporciona la base para un enfoque de prueba y un plan de prueba que cubra todas las iteraciones del proyecto. Durante la planificación del lanzamiento, los representantes comerciales (en colaboración con el equipo) determinan y priorizan las historias de usuarios para un lanzamiento. Sobre la base de estas historias de usuarios, se identifican los riesgos del proyecto y del producto (ver Sección 5.2) y se realiza una estimación del esfuerzo de alto nivel (ver Sección 5.1.4).

Los evaluadores participan en la planificación del lanzamiento y agregan valor, especialmente en las siguientes actividades:

- Definir historias de usuario comprobables, incluidos los criterios de aceptación.
- Participar en el análisis de riesgos (de calidad) del proyecto y del producto.
- Estimar el esfuerzo de prueba relacionado con las historias de usuario.
- Definir los niveles de prueba necesarios.
- Planificar las pruebas para el lanzamiento.

#### Planificación de iteraciones

Una vez que se completa la planificación del lanzamiento, comienza la planificación de iteraciones para la primera iteración. La planificación de iteraciones mira hacia el final de una sola iteración y aborda el trabajo pendiente de iteración. Durante la planificación de la iteración, el equipo selecciona historias de usuarios de la cartera de productos priorizada, las refina (aclara) y las divide (cuando sea necesario), las desarrolla, realiza análisis de riesgos para las historias de usuarios y estima el trabajo necesario para implementar cada historia seleccionada ( ver sección 5.1.4). Si una historia de usuario no está clara y los intentos de aclararla han fallado, el equipo puede rechazarla y utilizar la siguiente historia de usuario según la prioridad. Los representantes comerciales deben responder las preguntas del equipo sobre cada historia para que el equipo comprenda qué debe implementar y cómo probar cada historia.

El número de historias seleccionadas se basa en la llamada velocidad del equipo<sup>2</sup> y el tamaño estimado de las historias de usuario seleccionadas, así como en limitaciones técnicas. Una vez el

---

<sup>2</sup> La velocidad del equipo es la cantidad de trabajo determinada empíricamente que un equipo es capaz de realizar durante una sola iteración. Generalmente se expresa en términos de los llamados puntos de historia de usuario. El tamaño de cada historia también se estima en estas unidades, de modo que el equipo sepa cuántas historias de usuario puede incluir en el trabajo pendiente de iteración; la suma de su complejidad no puede exceder la velocidad del equipo. Esto reduce el riesgo de que el equipo no tenga tiempo para completar todo el trabajo a realizar en una iteración y que el equipo no termine el trabajo antes del final de la iteración, lo que provoca las llamadas ejecuciones vacías.


Una vez finalizado el contenido de la iteración, las historias de usuario se dividen en tareas que deben ejecutar los miembros del equipo correspondientes.

Los probadores participan en la planificación de iteraciones y agregan valor, especialmente en las siguientes actividades:

- Participar en análisis de riesgo detallados de historias de usuarios •
- Determinar la capacidad de prueba de las historias de usuarios
- Co-crear pruebas de aceptación para historias de usuarios •
- Dividir historias de usuarios en tareas (especialmente tareas de prueba) •
- Estimar el esfuerzo de prueba para todas las tareas de prueba •
- Identificar funciones funcionales y no -Aspectos de prueba funcionales del sistema bajo prueba. • Apoyar y participar en la automatización de pruebas en múltiples niveles de prueba.

### 5.1.3 Criterios de entrada y criterios de salida

#### Criterios de entrada


Los criterios de entrada  (más o menos similares a la Definición de Listo en un enfoque ágil) definen las condiciones previas que deben cumplirse antes de que pueda comenzar una actividad de prueba. Si no se cumplen, las pruebas pueden ser más difíciles y requerir más tiempo, costos y riesgos.

Los criterios de entrada incluyen la disponibilidad de recursos o software de prueba:

- Requisitos, historias de usuario y/o modelos comprobables • Elementos de prueba que cumplieron con los criterios de salida aplicables a niveles anteriores de prueba, principalmente en el enfoque en cascada • Entorno de prueba • Herramientas de prueba necesarias • Datos de prueba y otros recursos necesarios

así como el nivel de calidad inicial de un objeto de prueba (por ejemplo, todas las pruebas de humo pasan). Los criterios de entrada nos protegen de iniciar tareas para las que aún no estamos completamente preparados.

#### Criterios de salida

Los criterios de salida  (más o menos similares a la Definición de Hecho en un enfoque ágil) definen las condiciones que deben cumplirse para que la ejecución de un nivel de prueba o conjunto de pruebas se considere completa.

Estos criterios deben definirse para cada nivel de prueba y tipo de prueba. Pueden variar dependiendo de los objetivos de la prueba.

Los criterios de salida típicos son:

- Finalización de la ejecución de las pruebas programadas. • Lograr el nivel adecuado de cobertura (por ejemplo, requisitos, historias de usuarios, aceptación). criterios, código) •
- No exceder el límite acordado de defectos no reparados

- Obtener una densidad de defectos estimada suficientemente baja.
- Lograr índices de confiabilidad suficientemente altos.

Tenga en cuenta que a veces las actividades del examen pueden acortarse debido a:

- El uso de todo el presupuesto
- El paso del tiempo programado
- La presión de sacar un producto al mercado

En tales situaciones, las partes interesadas del proyecto y los propietarios de empresas deben conocer y aceptar los riesgos de ejecutar el sistema sin realizar más pruebas. Dado que estas situaciones ocurren a menudo en la práctica, los evaluadores (especialmente la persona que desempeña el rol de líder del equipo de pruebas) deben proporcionar información que describa el estado actual del sistema, destacando los riesgos.

Los criterios de salida suelen adoptar la forma de medidas de minuciosidad o integridad. Expresan el grado deseado de realización de un trabajo determinado. Los criterios de salida nos permiten determinar si hemos realizado determinadas tareas según lo previsto.

Ejemplo El equipo adoptó los siguientes criterios de salida de las pruebas del sistema:

- (EX1) logró una cobertura del 100% de los requisitos mediante casos de prueba
- (EX2) logró al menos un 75% de cobertura de declaración para cada componente probado
- (EX3) no hay defectos abiertos (no reparados) con el nivel más alto de gravedad
- (EX4) como máximo dos defectos abiertos de gravedad media o baja

Después de analizar el informe al final de la fase de prueba del sistema, quedó claro que:

- Para cada requisito, se diseñó al menos un caso de prueba.
- Para dos de los cuatro módulos, se logró una cobertura completa de la declaración; para el tercero, 80%; y para el cuarto, el 70%
- Una de las pruebas detectó un defecto de prioridad media que aún no está cerrado

Este análisis muestra que se cumplen los criterios (EX1), (EX3) y (EX4), mientras que no se cumple el criterio (EX2) para uno de los componentes. El equipo decidió analizar qué partes del código de este componente no están cubiertas y agregó un caso de prueba que cubre una ruta de flujo de control adicional en este módulo, lo que aumentó la cobertura de este módulo del 70 al 78 %. En este punto, se ha cumplido el criterio (EX2). Dado que en este punto se cumplen todos los criterios de salida de la fase de prueba del sistema, el equipo considera formalmente que esta fase está completa.

## 5.1.4 Técnicas de estimación

El esfuerzo de prueba es la cantidad de trabajo relacionado con las pruebas necesario para lograr los objetivos del proyecto de prueba. El esfuerzo de prueba es a menudo un factor de costo importante o significativo durante el desarrollo de software, y a veces consume más del 50% del esfuerzo total de desarrollo [64]. Una de las preguntas más comunes que los miembros del equipo escuchan de sus gerentes es "¿cuánto tiempo llevará realizar esta tarea?" Los problemas descritos anteriormente hacen la prueba.

La estimación es una actividad muy importante desde la perspectiva de la gestión de pruebas. Cada evaluador debe tener la capacidad de estimar el esfuerzo de la prueba, ser capaz de utilizar técnicas de estimación apropiadas, comprender las ventajas y desventajas de los diferentes enfoques y ser consciente de cuestiones como la precisión de la estimación (error de estimación).

El esfuerzo de prueba se puede medir como intensidad de mano de obra, una medida del producto (tiempo \* recursos). Por ejemplo, un esfuerzo de 12 días-persona significa que un trabajo determinado será realizado en 12 días por una persona, o en 6 días por dos personas, o en 4 días por tres personas, y así sucesivamente. En general, será un trabajo realizado en  $x$  días por  $y$  personas, donde  $x \cdot y = 12$ .

Sin embargo, hay que tener cuidado porque, en la práctica, las medidas de los productos "no escalan". Por ejemplo, si un equipo de tres personas realiza algún trabajo en 4 días, la intensidad laboral es de 12 días-persona. Sin embargo, si un gerente agrega una nueva persona al equipo, con la esperanza de que cuatro personas hagan el mismo trabajo en 3 días, puede sentirse decepcionado: un equipo tan grande aún puede hacer el trabajo en 4 días o incluso más. Esto se debe a una serie de factores, como un aumento en los esfuerzos de comunicación (más personas en el equipo) o la necesidad de que los miembros del equipo con más años de servicio dediquen tiempo a formar a nuevos miembros, lo que puede causar retrasos. Los directores de proyecto tienen un dicho que ilustra este fenómeno: "una mujer tendrá un bebé en nueve meses, pero nueve mujeres no tendrán un bebé en un mes".

Al realizar una estimación del esfuerzo, a menudo no se dispone de todo el conocimiento sobre el objeto de prueba, por lo que la precisión de la estimación puede ser limitada. Es importante explicar a las partes interesadas que la estimación se basa en muchos supuestos. La estimación se puede refinar y posiblemente ajustar más adelante (por ejemplo, cuando haya más datos disponibles). Una posible solución es utilizar un rango de estimación para proporcionar una estimación inicial (por ejemplo, el esfuerzo será de 10 meses-persona con una desviación estándar de 3 meses-persona, lo que significa que con alta probabilidad, el esfuerzo real será de entre 10 y 13 meses-persona).

La estimación de tareas pequeñas suele ser más precisa que la de las grandes. Por lo tanto, al estimar una tarea compleja, se puede utilizar una técnica de descomposición llamada estructura de desglose del trabajo (WBS). En esta técnica, la tarea o actividad principal (compleja) que se va a estimar se descompone jerárquicamente en subtareas más pequeñas (más simples). El objetivo es dividir la tarea principal en componentes fácilmente estimables pero ejecutables. La tarea debe dividirse con la mayor precisión posible basándose en la información actual, pero sólo en la medida necesaria para comprender y estimar con precisión una sola subtask. Después de estimar todas las subtareas en el nivel más bajo, las tareas de nivel superior se estiman (sumando las estimaciones de las subtareas relevantes) de manera ascendente. En el último paso se obtiene la estimación de la tarea principal.

En la figura 5.2 se muestra un ejemplo de uso del método WBS.

Queremos estimar el esfuerzo de todo el proyecto de prueba. Sin embargo, es tan grande que no tiene sentido estimar todo el esfuerzo de una vez, ya que el resultado estará sujeto a un error muy grande. Entonces dividimos el proyecto de prueba en las tareas principales: planificación, definición del entorno de prueba, pruebas de integración y pruebas del sistema. Supongamos que podemos estimar con bastante precisión el esfuerzo para las dos primeras tareas (1 y 4 días-persona, respectivamente). Sin embargo, las pruebas de integración requerirán mucho esfuerzo, por lo que dividimos jerárquicamente esta tarea en tareas más pequeñas. Identificamos dos

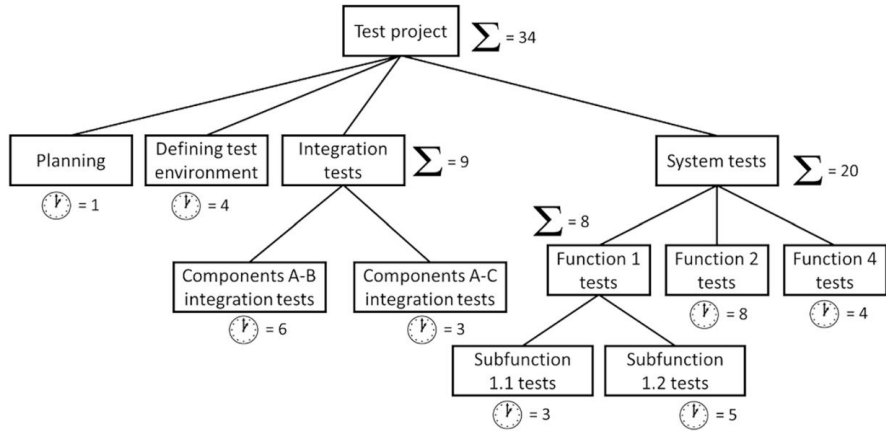


Fig. 5.2 Uso del método WBS para estimar el esfuerzo del proyecto de prueba

subtareas dentro de las pruebas de integración: integración de los componentes A y B (6 días-persona) e integración de los componentes A y C (3 días-persona). En las pruebas del sistema, identificamos tres funciones, F1, F2 y F3, como objetos de prueba. Para F2 y F3, estimamos que el esfuerzo de prueba será de 8 y 4 días-persona, respectivamente. La función F1 es demasiado grande, por lo que la dividimos en dos subtareas y estimamos el esfuerzo relacionado en 3 y 5 días-persona, respectivamente. Ahora podemos recopilar los resultados de los niveles más bajos y estimar las tareas en los niveles superiores sumando las subtareas correspondientes. Por ejemplo, el esfuerzo para las pruebas de integración (9 días-persona) será la suma del esfuerzo de dos subtareas con un esfuerzo de 6 y 3 días-persona, respectivamente. De manera similar, el esfuerzo para F1 es  $3+5 = 8$  días-persona, y el esfuerzo para probar el sistema es  $8+8+4 = 20$  días-persona. Al final, calculamos el esfuerzo de todo el proyecto de prueba como la suma de sus subtareas:  $1+4+9+20 = 34$  días-persona.

Las técnicas de estimación se pueden dividir en dos grupos principales:

- Técnicas basadas en métricas: donde el esfuerzo de prueba se basa en métricas de proyectos similares anteriores, en datos históricos del proyecto actual o en valores típicos (las llamadas líneas de base de la industria).
- Técnicas basadas en expertos: donde el esfuerzo de la prueba se basa en la experiencia de los propietarios de las tareas de prueba o de los expertos en la materia.

El programa de estudios describe las siguientes cuatro técnicas de estimación utilizadas con frecuencia en la práctica.

**Estimación basada en ratios** En esta

técnica basada en métricas, se recopila la mayor cantidad posible de datos históricos de proyectos anteriores, lo que permite derivar ratios "estándar" de varios indicadores para proyectos similares. Los ratios propios de una organización suelen ser la mejor fuente para utilizar en el proceso de estimación. Estos índices estándar se pueden utilizar luego para estimar el esfuerzo de prueba para un nuevo proyecto. Por ejemplo, si en un proyecto similar anterior la proporción de

El esfuerzo de implementación respecto al esfuerzo de prueba fue de 3:2, y en el proyecto actual se espera que el esfuerzo de desarrollo sea de 600 días-persona, el esfuerzo de prueba se puede estimar en 400 días-persona, ya que la misma o similar proporción de implementación a prueba. Lo más probable es que la prueba se realice como en el proyecto anterior similar.

#### Extrapolación En

esta técnica basada en métricas, las mediciones se toman lo antes posible para recopilar datos históricos reales del proyecto actual. Con suficientes observaciones de este tipo (puntos de datos), el esfuerzo requerido para el trabajo restante se puede aproximar extrapolando estos datos. Este método es muy útil en técnicas de desarrollo de software iterativo. Por ejemplo, un equipo puede extrapolar el esfuerzo de prueba en la cuarta iteración como un promedio del esfuerzo en las últimas tres iteraciones. Si el esfuerzo en las últimas tres iteraciones fue de 30, 32 y 25 días-persona, respectivamente, la extrapolación del esfuerzo para la cuarta iteración sería  $(30+32+25) / 3 = 29$  días-persona.

Tenga en cuenta que, procediendo de manera análoga, podríamos estimar el esfuerzo para iteraciones posteriores utilizando la extrapolación calculada para iteraciones anteriores. En nuestro ejemplo, la extrapolación del esfuerzo para la quinta iteración será el promedio de las iteraciones 2, 3 y 4, por lo que será  $(32+25+29) / 3 = 14,66$ . De forma similar, podemos extrapolar el esfuerzo para la sexta iteración:  $(25 + 29 + 14,66) / 3$  y así sucesivamente. Sin embargo, tenga en cuenta que cuanto más "más" sea el punto de datos que extrapolamos, mayor será el riesgo de cometer un error de estimación creciente, ya que la primera estimación ya tiene algún error. Aplicar dicho resultado a la siguiente estimación puede hacer que el error aumente (reducir la precisión del valor estimado), porque los errores pueden acumularse.

#### Wideband Delphi En el

método Wideband Delphi basado en expertos, los expertos hacen estimaciones basadas en su propia experiencia. Cada experto, de forma aislada, estima la carga de trabajo. Se recopilan los resultados y los expertos discuten sus estimaciones actuales. Luego se pide a cada experto que haga nuevas predicciones basadas en esta información. La discusión permite a todos los expertos repensar su proceso de estimación. Puede ser, por ejemplo, que algunos expertos no hayan tenido en cuenta ciertos factores al realizar la estimación. El proceso se repite hasta que se alcanza un consenso o hay un rango suficientemente pequeño en las estimaciones obtenidas.

En esta situación, por ejemplo, la media o mediana de las estimaciones de los expertos puede considerarse el resultado final.

La figura 5.3 muestra la idea detrás del método Delphi. De iteración en iteración, el rango de valores estimados por los expertos se estrecha. Una vez que sea lo suficientemente pequeño, se puede extraer, por ejemplo, la media o mediana de todas las estimaciones y considerarla el resultado del proceso de estimación. Muy a menudo, este resultado no se desviará demasiado del valor real. Este fenómeno se conoce coloquialmente como "sabiduría de la multitud" y resulta del simple hecho de que los errores se anulan entre sí: un experto puede sobreestimar un poco, otro puede subestimar un poco, otro puede subestimar mucho y otro más puede subestimar un poco. Puede sobreestimar mucho. Por tanto, el error suele tener una distribución normal con la media igual a cero. La dispersión de los resultados puede ser muy grande, pero promediarlos será a menudo una aproximación suficiente del valor real.

Una variación del método Delphi es el llamado póquer de planificación. Este es un enfoque comúnmente utilizado en metodologías ágiles. Al planificar el póquer, las estimaciones son



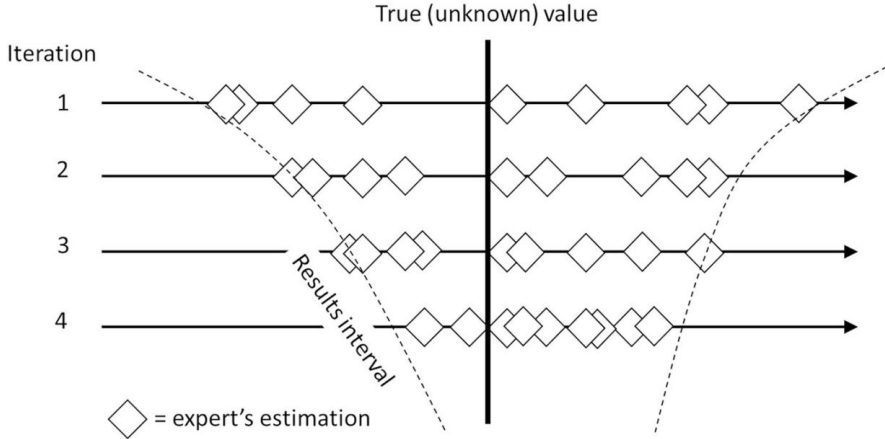


Fig. 5.3 El método Delphi de banda ancha

Realizado con tarjetas numeradas. Los valores de las tarjetas representan la cantidad de esfuerzo expresada en unidades específicas que están bien definidas y comprendidas por todos los expertos.

Las cartas de póquer de planificación suelen contener valores derivados de la llamada secuencia de Fibonacci, aunque para valores mayores puede haber algunas desviaciones. En la secuencia de Fibonacci, cada término sucesivo es la suma de los dos anteriores. Los siguientes valores se utilizan con mayor frecuencia:

1, 2, 3, 5, 8, 13, 20, 40, 100

Como puedes ver, los dos últimos valores ya no son la suma de los dos anteriores. Esto se debe a que son lo suficientemente grandes como para asumir aquí valores redondeados, simplemente representando algo lo suficientemente grande como para que ni siquiera tenga sentido estimar dicho valor con precisión. Si un equipo estima, por ejemplo, el esfuerzo necesario para implementar y probar alguna historia de usuario y la mayoría de los expertos tiran una carta de 40 o 100 sobre la mesa, esto significa que la historia es lo suficientemente grande como para que probablemente sea una épica y deba desglosarse en varias historias más pequeñas, cada una de las cuales ya puede estimarse razonablemente.

En la figura 5.4 se muestra un ejemplo de una baraja de cartas de póquer de planificación. Si una tarjeta con "?" es arrojado, significa que el experto no tiene suficiente información para hacer la estimación. Un valor de "0", por otro lado, significa que el experto considera que la historia es trivial de implementar y que el tiempo dedicado a ella será insignificante. Una tarjeta con una imagen de café significa "Estoy cansado, ¡hagamos un descanso y tomemos un café!".

Otros conjuntos de valores utilizados con frecuencia en la planificación del póquer son el conjunto de potencias sucesivas de dos:

1, 2, 4, 8, 16, 32, 64, 128

o las llamadas tallas de camiseta.

XS, S, M, L, XL.

En el último caso, la medición (estimación) no se realiza en una escala numérica de relación y, por lo tanto, no representa directamente el tamaño físico del objeto estimado.



Fig. 5.4 Planificación de cartas de póquer

esfuerzo. Las tallas de camisetas se definen únicamente en una escala ordinal, lo que permite únicamente comparar elementos entre sí. Por lo tanto, podemos decir que un piso estimado como “L” requiere más mano de obra que uno estimado como “S”, pero no podemos decir cuántas veces más esfuerzo se necesitará para implementar un piso de tamaño “L” en relación con uno. de talla “S”. En esta variante del método, sólo podemos priorizar historias por esfuerzo, agrupándolas en cinco grupos de tamaño de esfuerzo creciente. Sin embargo, esta no es una práctica recomendada, porque desde el punto de vista de la teoría de la medición, es importante que las diferencias entre los sucesivos grados de la escala sean constantes.

En el caso de que la escala exprese puntos de historia, no hay problema: la diferencia entre 4 y 5 puntos es la misma que entre 11 y 12 puntos y es de 1 punto.

Sin embargo, esto ya no es tan obvio en el caso de una escala como “tallas de camisa”. Así que se debe suponer que, por ejemplo, la diferencia entre S y XS es la misma que entre L y M, y así sucesivamente.

También se aceptan otras escalas. Cuál usar específicamente es una decisión del equipo o de la gerencia. Sólo es importante entender cómo una unidad de esta escala se traduce en una unidad de esfuerzo. A menudo, los equipos toman como unidad el llamado punto de la historia del usuario. Entonces, una unidad puede significar la cantidad de esfuerzo necesario para implementar/probar una historia de usuario promedio. Si el equipo sabe por experiencia, por ejemplo, que implementar una historia de usuario de 1 punto de historia generalmente les lleva 4 días-persona, entonces un componente estimado en 5 puntos de historia de usuario debería requerir aproximadamente  $5 * 4 = 20$  días-persona de esfuerzo.