

Estimación de tres puntos

En esta técnica basada en expertos, los expertos realizan tres tipos de estimación: la más optimista (a), la más probable (m) y la más pesimista (b). La estimación final (E) es su media aritmética ponderada calculada como

$$E = \frac{a + 4m + b}{6}$$

La ventaja de esta técnica es que permite a los expertos calcular directamente también el error de medición:

$$DE = \frac{b - a}{6}$$

Por ejemplo, si las estimaciones (en horas-persona) son $a = 6$, $m = 9$ y $b = 18$, la estimación final es 10 ± 2 horas-persona (es decir, entre 8 y 12 horas-persona), porque

$$mi = \frac{6 + 4 \cdot 9 + 18}{6} = 10$$

y

$$DE = \frac{18 - 6}{6} = 2$$

Tenga en cuenta que el resultado de la estimación no siempre será igual al valor más probable m, porque dependerá de qué tan lejos esté este valor de los valores optimistas y pesimistas. Cuanto más cerca esté el valor de m del valor de estimación extremo (a o b), mayor será la distancia entre el valor de m y el resultado de la estimación E.

La fórmula que se muestra arriba es la más utilizada en la práctica. Se deriva del método de la técnica de evaluación y revisión de programas (PERT). Sin embargo, los equipos a veces usan otras variantes, ponderando el factor m de manera diferente, por ejemplo, con un peso de 3 o 5. Entonces la fórmula toma la forma $E = (a + 3m + b) / 5$ o $E = (a + 5m + b) / 7$.

Para obtener más información sobre estas y otras técnicas de estimación, consulte [45, 65, 66].

Ejemplo Un equipo utiliza un póquer de planificación para estimar la implementación y el esfuerzo de prueba de una determinada historia de usuario. El poker de planificación se juega en sesiones según el siguiente procedimiento.

1. Cada jugador tiene un juego completo de cartas.
2. El moderador (generalmente el propietario del producto) presenta la historia del usuario estimado.
3. A continuación se hace una breve discusión del alcance del trabajo, aclarando ambigüedades.
4. Cada experto elige una carta.
5. A la señal del facilitador, todos arrojan simultáneamente su tarjeta al centro de la mesa.
6. Si todos han elegido la misma carta, se llega a un consenso y la reunión finaliza; el resultado es el valor elegido por todos los miembros del equipo. Si después de un número fijo

Después de iteraciones, el equipo no ha llegado a un consenso, el resultado final se determina de acuerdo con la variante establecida y aceptada del procedimiento (ver más abajo).

7. Si las estimaciones difieren, la persona que eligió la estimación más baja y la persona que eligió la más alta presentan su punto de vista.
8. Si es necesario, sigue una breve discusión.
9. Volver al punto 4.

La planificación del poker, como cualquier técnica de estimación basada en el juicio de expertos, sólo será tan efectiva como lo sean los buenos expertos que participan en la sesión de estimación (ver Fig. 5.5).

Por lo tanto, las personas que realicen estimaciones deben ser expertos, preferiblemente con años de experiencia.

El equipo tiene las siguientes opciones posibles para determinar la estimación final cuando no se ha alcanzado un consenso después de un cierto número de iteraciones de póquer.

Supongamos que el equipo que realiza la estimación del esfuerzo para una determinada historia de usuario está formado por cinco personas y cada participante tiene una baraja completa de cartas con valores 0, 1, 2, 3, 5, 8, 13, 20, 40 y 100. Supongamos que en la última ronda, cinco personas presenten simultáneamente una carta de su elección. Los resultados son:

3, 8, 13, 13, 20.

Variante 1 Se promedia el resultado y la estimación final es el promedio (11,4) redondeado al valor más cercano del mazo, que es 13.

Variante 2 Se elige el valor más común que es 13.

Variante 3 El resultado es la mediana (valor medio) de los valores 3, 8, 13, 13 y 20, es decir, 13.

Variante 4 Además de proporcionar una estimación, cada persona también califica el grado de confianza en esa estimación, en una escala de 1 (alta incertidumbre) a 5 (alta confianza). El grado de certeza puede reflejar, por ejemplo, el nivel de experiencia y conocimientos en el área en la que se realiza la estimación. Supongamos que los estimadores 3, 8, 13, 13 y 20 calificaron la certeza de su elección en 4, 5, 2, 3 y 2, respectivamente.

Luego, la puntuación se calcula como un promedio ponderado:

$$\frac{3 \times 4 + 8 \times 5 + 13 \times 2 + 13 \times 3 + 20 \times 2}{4 + 5 + 2 + 3 + 2} = \frac{112}{16} = 7$$

Luego, el resultado se puede redondear al valor más cercano en la escala, que es 8. La ventaja de este enfoque es que los resultados de aquellos más convencidos de la exactitud de su estimación pesan más en la evaluación promediada. Por supuesto, la precisión del método dependerá en particular de la precisión de las puntuaciones de confianza de los evaluadores individuales.

Las opciones anteriores no agotan todas las formas posibles de estimación. Cada equipo puede adoptar sus propias reglas individuales para determinar el valor final del parámetro que se somete al procedimiento de estimación. También hay que tener en cuenta que todos estos

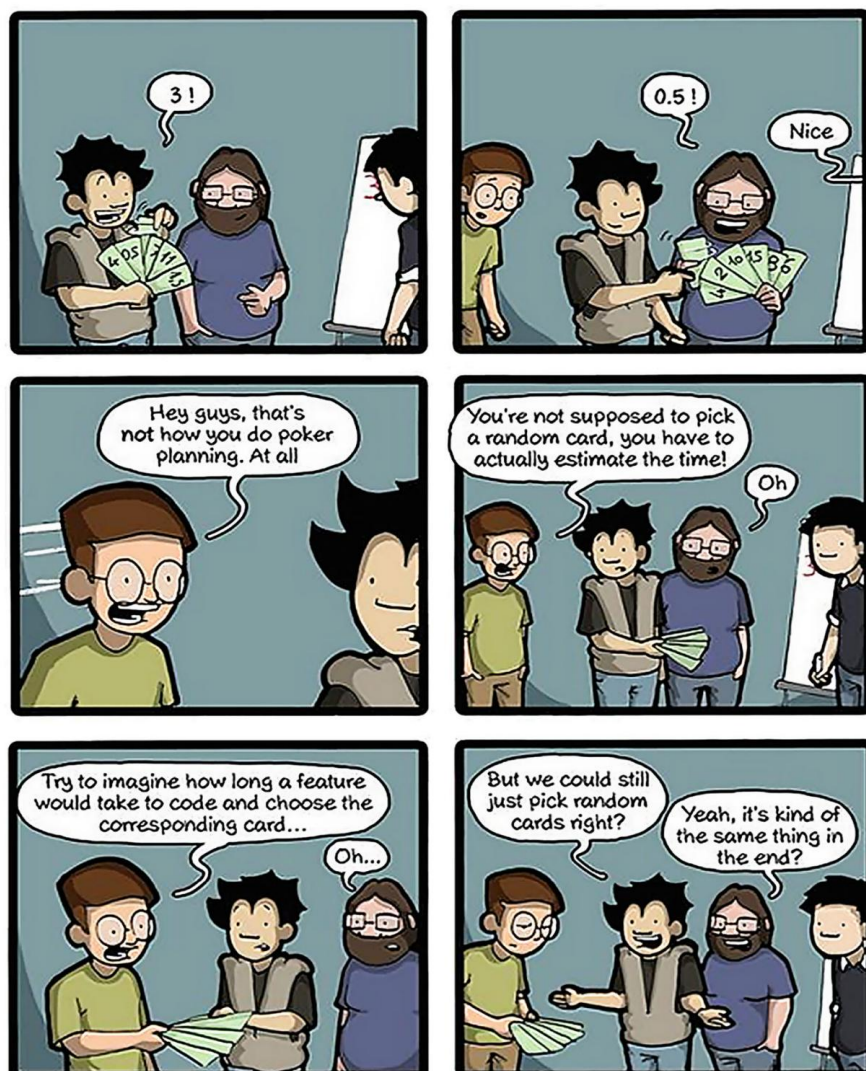


Fig. 5.5 Con humor sobre la planificación del poker (www.commitstrip.com/en/2015/01/22/poker-planning)

Las variantes para seleccionar la respuesta final en ausencia de consenso siempre están sujetas a algún error. Por ejemplo, tomar la mediana (13) de los valores 8, 13, 13, 13, 13 y 13 estará sujeto a menos error que, por ejemplo, de los valores 1, 8, 13, 40 y 100, debido a la varianza mucho mayor de los resultados en el último caso.

Factores que afectan el esfuerzo de la prueba.

Los factores que afectan el esfuerzo de prueba incluyen:

- Características del producto (p. ej., riesgos del producto, calidad de las especificaciones (es decir, base de prueba), tamaño del producto, complejidad del dominio del producto, requisitos para las características de calidad (p. ej., seguridad y confiabilidad), nivel requerido de detalle en la documentación de prueba, requisitos regulatorios, requisitos de conformidad)
- Características del proceso de desarrollo de software (p. ej., estabilidad y madurez de la organización, modelo SDLC utilizado, enfoque de prueba, herramientas utilizadas, proceso de prueba, presión de tiempo)
- Factores humanos (p. ej., habilidades y experiencia de los evaluadores, cohesión del equipo, capacidad del gerente).
habilidades)
- Resultados de las pruebas (p. ej., número, tipo e importancia de los defectos detectados, número de correcciones requeridas, número de pruebas fallidas)

La información sobre los tres primeros grupos de factores generalmente se conoce de antemano, a menudo antes de iniciar el proyecto. Los resultados de las pruebas, por otro lado, son un factor que funciona "desde adentro": no podemos incluir los resultados de las pruebas en la estimación antes de ejecutarlas. Por lo tanto, sus resultados pueden afectar significativamente la estimación más adelante en el proyecto.

5.1.5 Priorización de casos de prueba

Una vez que los casos de prueba y los procedimientos de prueba se crean y organizan en conjuntos de pruebas, estos conjuntos se pueden organizar en un programa de ejecución de pruebas, que define el orden en el que se ejecutarán. Se deben tener en cuenta varios factores al priorizar los casos de prueba (y, por lo tanto, el orden en que se ejecutan).

Un cronograma es una forma de planificar tareas a lo largo del tiempo. El cronograma debe tener en cuenta cuenta:

- Prioridades •
- Dependencias • La
- necesidad de pruebas de confirmación y regresión • El orden más
- efectivo para realizar las pruebas

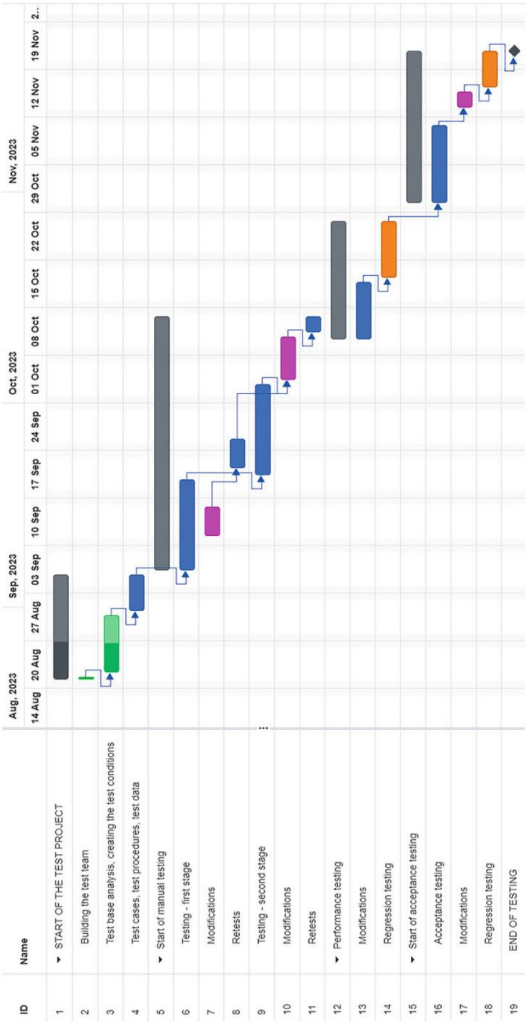
Al crear un cronograma de ejecución de pruebas, es importante considerar:

- Las fechas de las actividades centrales dentro del cronograma del proyecto • Que el
- cronograma esté dentro del cronograma del proyecto • Los hitos: inicio y fin de cada etapa

Una de las formas gráficas más típicas de presentar el cronograma es el llamado diagrama de Gantt (ver Fig. 5.6). En este gráfico, las tareas se representan como rectángulos que expresan su duración y las flechas definen diferentes tipos de relaciones entre tareas.

Por ejemplo, si una flecha va desde el final del rectángulo X hasta el comienzo del rectángulo Y, significa que la tarea Y solo puede comenzar cuando la tarea X haya finalizado.

Las pruebas bien planificadas deben tener lo que se conoce como una tendencia lineal: si conecta los hitos de "inicio del proyecto de prueba" y "fin de las pruebas" de la figura 5.6 con una línea, las tareas realizadas deben alinearse a lo largo de esta línea. Esto significa que las tareas relacionadas con la prueba



se realizan de tal manera que el equipo de pruebas trabaja a un ritmo constante, sin persecuciones de tiempo innecesarias.

La práctica muestra que el período de prueba final (>95% de los casos) requiere que se comprometa la cantidad máxima de recursos en la fase de prueba final (la fase final del proyecto); consulte la figura 5.7.

Las estrategias de priorización de casos de prueba más comunes son las siguientes.

- Priorización basada en riesgos, donde el orden de ejecución de las pruebas se basa en los resultados del análisis de riesgos (ver Sección 5.2.3). Primero se ejecutan los casos de prueba que cubren los riesgos más importantes.
- Priorización basada en cobertura, donde el orden de ejecución de la prueba se basa en la cobertura (por ejemplo, cobertura de código, cobertura de requisitos). Los casos de prueba que alcanzan la mayor cobertura se ejecutan primero. En otro enfoque, llamado priorización basada en cobertura adicional, el caso de prueba que logra la cobertura más alta se ejecuta primero. Cada caso de prueba posterior es el que logra la mayor cobertura adicional.
- Priorización basada en requisitos, donde el orden en que se ejecutan las pruebas se basa en prioridades de requisitos que están vinculadas a los casos de prueba correspondientes.

Las prioridades de los requisitos las determinan las partes interesadas. Primero se ejecutan los casos de prueba relacionados con los requisitos más importantes.

Idealmente, los casos de prueba deberían ordenarse para ejecutarse en función de sus niveles de prioridad, utilizando, por ejemplo, una de las estrategias de priorización mencionadas anteriormente. Sin embargo, es posible que esta práctica no funcione si los casos de prueba o las funciones que se prueban son dependientes. Aquí se aplican las siguientes reglas:

- Si un caso de prueba de mayor prioridad depende de un caso de prueba de menor prioridad, el caso de prueba de menor prioridad El caso de prueba prioritario debe ejecutarse primero.
- Si existen interdependencias entre varios casos de prueba, el orden de sus La ejecución debe determinarse independientemente de las prioridades relativas. •

Es posible que sea necesario priorizar las ejecuciones de pruebas de confirmación y regresión. •

Se debe equilibrar adecuadamente la eficiencia de la ejecución de las pruebas versus el cumplimiento de las prioridades establecidas.

El orden en que se ejecutan las pruebas también debe tener en cuenta la disponibilidad de recursos. Por ejemplo, es posible que las herramientas, los entornos o las personas necesarios solo estén disponibles dentro de un período de tiempo determinado.

Ejemplo Estamos probando un sistema cuyo CFG se muestra en la figura 5.8. Estamos adoptando un método de priorización de pruebas basado en la cobertura. El criterio que utilizaremos será la cobertura de declaraciones. Supongamos que tenemos cuatro casos de prueba TC1-TC4 definidos en la Tabla 5.1.

Esta tabla también proporciona la cobertura que logra cada caso de prueba. La cobertura más alta (70%) la logra el TC4, seguida del TC1 y TC2 (60% cada uno), y la más baja la del TC3 (40%). En consecuencia, la prioridad de ejecución de la prueba será la siguiente: TC4, TC2, TC1 y TC3, donde TC1 y TC2 se pueden ejecutar en cualquier orden ya que logran la misma cobertura.

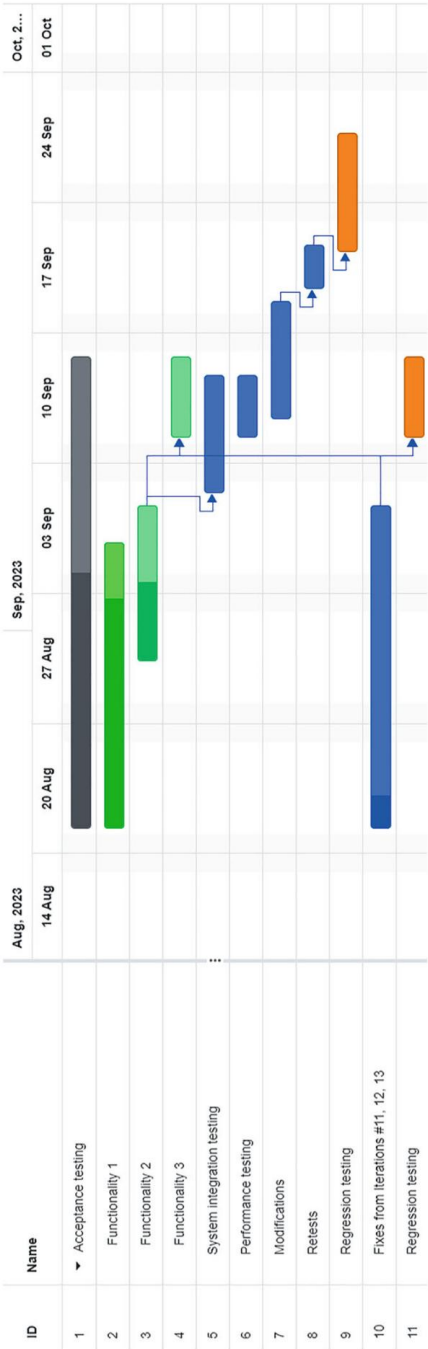


Fig. 5.8 CFG del sistema bajo prueba

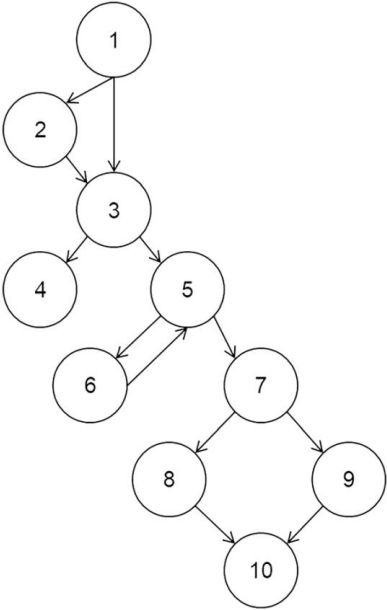


Tabla 5.1 Casos de prueba existentes y cobertura que logran

Caso de prueba	Camino ejercido	Cobertura	Prioridad
TC1	1→3→5→7→8→10	60%	2
TC2	1→3→5→7→9→10	60%	2
TC3	1→2→3→4	40%	3
TC4	1→3→5→6→5→7→8→10	70%	1

Sin embargo, tenga en cuenta que la ejecución de TC1, después de la ejecución de TC4 y TC2, no cubrirá ninguna declaración nueva. Entonces apliquemos una variante de priorización basada en cobertura adicional. Se dará máxima prioridad, como en la variante anterior, a el caso de prueba que logra la mayor cobertura, a saber, TC4. Ahora veamos cuantos declaraciones adicionales no cubiertas por TC4 están cubiertas por los otros casos de prueba.

El cálculo se muestra en la Tabla 5.2 en la penúltima columna. TC1 no cubrir cualquier declaración adicional, porque las declaraciones que cubre (es decir, 1, 3, 5, 7, 8, 10) ya están cubiertos por el TC4. TC2, respecto a TC4, cubre adicionalmente estado de cuenta 9, por lo que logra un 10% de cobertura adicional (cubrió un estado de cuenta nuevo de diez en total). TC3, en relación con TC4, cubre adicionalmente dos afirmaciones, 2 y 4, lo que resultó en una cobertura adicional del 20% (cubrió dos declaraciones nuevas de cada diez total). Por lo tanto, el siguiente caso de prueba después del TC4 que cubre la mayoría de las declaraciones nuevas aún no cubierto es TC3. Repetimos ahora el análisis anterior contando la cobertura adicional en relación con la cobertura lograda por TC4 y TC3. El cálculo se muestra en la última columna de la Tabla 5.2. TC1 no cubre nada nuevo. TC2, por su parte, cubre una nueva declaración que TC4 y TC3 no cubrieron, a saber, 9. Por lo tanto, logra una

Tabla 5.2 Cálculo de la cobertura adicional lograda mediante pruebas

Prueba	Cobertura	Adicional cobertura después de TC4	Cobertura adicional después de TC4 y TC3
caso Camino ejercido			
TC1 1→3→5→7→8→10	60%	0%	0%
CT2 1→3→5→7→9→10	60%	10% (9)	10% (9)
CT3 1→2→3→4	40%	20% (2, 4)	-
CT4 1→3→5→6→5→7→8→10 70%		-	-

cobertura adicional del 10%. Así, obtuvimos la siguiente priorización según cobertura adicional:

TC4 → TC3 → TC2 → TC1.

Tenga en cuenta que este orden difiere del orden resultante de la variante anterior de el método. Su ventaja es que obtenemos cobertura de todas las declaraciones muy rápidamente (después de sólo las tres primeras pruebas). En la variante anterior (con el orden TC4→TC2→TC1→TC3), las afirmaciones 2 y 4 están cubiertas solo en la última, cuarta prueba.

Otro ejemplo muestra que a veces las dependencias técnicas o lógicas entre pruebas puede alterar el orden deseado de ejecución de las pruebas y obligamos, por Por ejemplo, ejecutar primero una prueba de menor prioridad para “desbloquear” la posibilidad. de ejecutar una prueba de mayor prioridad.

Ejemplo Supongamos que estamos probando la funcionalidad de un sistema para realizar exámenes de conducción de forma electrónica. Supongamos también que al comienzo de cada ejecución de prueba, el La base de datos del sistema está vacía. Se han identificado los siguientes casos de prueba, junto con sus prioridades:

- TC1: Agregar el examinado a la base de datos. Prioridad: baja
- TC2: Realización del examen para el examinado. Prioridad: alta
- TC3: Realización de un examen de revisión. Prioridad: media
- CT4: Tomar una decisión y enviar los resultados del examen al examinado. Prioridad: medio

El caso de prueba con mayor prioridad aquí es TC2 (realizar el examen), pero No podemos ejecutarlo antes de agregar al examinado a la base de datos. Entonces, aunque TC2 tiene un mayor prioridad que TC1, la dependencia lógica que se produce requiere que ejecutemos Primero TC1, que "desbloqueará" la ejecución de la prueba para los otros casos de prueba. la próxima prueba El caso a ejecutar sería TC2, como tarea de mayor prioridad. Al final, podemos ejecutar TC3 y TC4. Tenga en cuenta que para aprobar un examen de revisión, el examinado debe saber que reprobaron el examen original. Por tanto, desde la perspectiva de un usuario particular, tiene sentido ejecutar TC3 sólo después de completar TC4. De este modo, La ejecución de la prueba final es la siguiente:

TC1 → TC2 → TC4 → TC3.

Tabla 5.3 Tareas con sus prioridades y dependencias

Tarea	Prioridad (1 = alta, 5 = baja)	Depende de
12		5
2	4	
3	1	2, 4
4	1	2, 1
53		6
63		7
7	5	

En muchas situaciones, puede suceder que el orden en el que se realizan las acciones depende de varios factores, como la prioridad y las dependencias lógicas y técnicas. Existe un método sencillo para abordar este tipo de tareas, incluso si las dependencias parecen complicado. Consiste en apuntar a la ejecución lo antes posible de tareas de mayor prioridad, pero si están bloqueadas por otras tareas, debemos ejecutar esas tareas. primero. Dentro de estas tareas, también establecemos su orden según prioridades y dependencias. Consideremos este método con un ejemplo concreto.

Ejemplo En la Tabla 5.3 se proporciona una lista de tareas con sus prioridades y dependencias .

Empezamos por establecer un orden que dependa únicamente de prioridades, sin mirando las dependencias, con tareas de igual prioridad agrupadas:

3, 4 → 1 → 5, 6 → 2 → 7.

Dentro del primer grupo, la tarea 3 depende de la tarea 4. Por lo tanto, aclaramos el orden. de las tareas 3 y 4:

4 → 3 → 1 → 5, 6 → 2 → 7.

Comprobamos ahora si podemos realizar las tareas en este orden. Resulta que no podemos: La tarea 4 depende de las tareas 2 y 1, y la tarea 1 tiene mayor prioridad que la tarea 2. Entonces movemos ambas tareas antes de la tarea 4 con sus prioridades preservadas, y después de la tarea 4, mantenemos el orden original de las otras tareas:

1 → 2 → 4 → 3 → 5, 6 → 7.

¿Podemos realizar las tareas en este orden? Todavía no: la tarea 1 depende de la tarea 5, esto en la tarea 6, y ésta a su vez en la tarea 7. Por lo tanto, debemos mover las tareas 5, 6 y 7 delante de tarea 1 con sus dependencias (tenga en cuenta que aquí las prioridades de las tareas 5, 6 y 7 no importa: el orden es forzado por la dependencia):

7 → 6 → 5 → 1 → 2 → 4 → 3.

Tenga en cuenta que en este punto ya podemos realizar la secuencia completa de tareas, 7, 6, 5, 1, 2, 4 y 3, ya que cada tarea en la secuencia anterior no depende de ninguna otra. tarea o depende sólo de las tareas que la preceden.

5.1.6 Pirámide de prueba

La pirámide de pruebas es un modelo que muestra que diferentes pruebas pueden tener diferente "granularidad". El modelo de pirámide de pruebas apoya al equipo en la automatización de pruebas y la asignación del esfuerzo de prueba. Las capas de la pirámide representan grupos de pruebas. Cuanto mayor sea la capa, menor será la granularidad de la prueba, el aislamiento de la prueba, la velocidad de ejecución de la prueba y el costo de calidad.

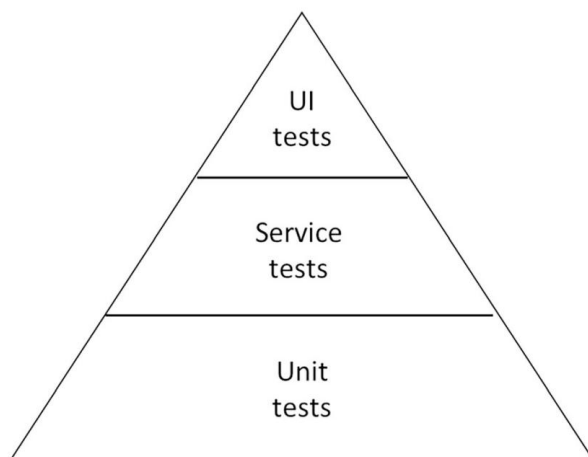
Las pruebas en la capa inferior de la pirámide de pruebas son pequeñas, aisladas y rápidas y verifican una pequeña parte de la funcionalidad, por lo que generalmente se necesitan muchas para lograr una cobertura razonable. La capa superior representa pruebas de extremo a extremo de gran tamaño y de alto nivel. Estas pruebas de alto nivel son más lentas que las pruebas de las capas inferiores y normalmente comprueban una gran parte de la funcionalidad, por lo que normalmente sólo se necesitan algunas para lograr una cobertura razonable.

El número y el nombre de las capas en una pirámide de prueba pueden variar. Por ejemplo, el modelo de pirámide de pruebas original [67] define tres capas: "pruebas unitarias" (llamadas pruebas de componentes en el programa de estudios ISTQB®), "pruebas de servicio" y "pruebas de UI" (ver Fig. 5.9). Otro modelo popular define pruebas unitarias (de componentes), pruebas de integración y pruebas de un extremo a otro.

El costo de la ejecución de pruebas desde una capa inferior suele ser mucho menor en relación con el costo de la ejecución de pruebas desde una capa superior. Sin embargo, como se mencionó anteriormente, una sola prueba de una capa inferior generalmente logra una cobertura mucho menor que una sola prueba de una capa superior. Por lo tanto, en la práctica, en proyectos de desarrollo típicos, el esfuerzo de prueba se refleja bien en el modelo de pirámide de pruebas. El equipo normalmente escribe muchas pruebas de bajo nivel y pocas pruebas de alto nivel. Sin embargo, tenga en cuenta que la cantidad de pruebas de diferentes niveles siempre estará determinada por el contexto amplio del proyecto y producto.

Puede suceder, por ejemplo, que la mayoría de las pruebas que realizará el equipo sean pruebas de integración y aceptación (por ejemplo, en la situación de integrar dos productos comprados a una casa de software; en este caso, no tiene sentido realizar pruebas de componentes en todo).

Fig. 5.9 Pirámide de prueba



Aunque la pirámide de pruebas es ampliamente conocida, todavía hay bastantes organizaciones que le dan la vuelta a esta pirámide, centrándose en las pruebas más complejas (UI). Una variante es la pirámide inclinada, donde pruebas específicas y series de pruebas cortan la pirámide, siguiendo un enfoque funcional.

5.1.7 Prueba de cuadrantes

El modelo de cuadrantes de prueba, definido por Brian Marick [68], [21], alinea los niveles de prueba con tipos de prueba, actividades, técnicas y productos de trabajo relevantes en enfoques de desarrollo ágil. El modelo respalda la gestión de pruebas al garantizar que todos los tipos y niveles de prueba importantes se incluyan en el proceso de desarrollo de software y al comprender que algunos tipos de pruebas están más relacionados con ciertos niveles de prueba que otros. El modelo también proporciona una manera de distinguir y describir los tipos de pruebas para todas las partes interesadas, incluidos desarrolladores, evaluadores y representantes comerciales. El modelo del cuadrante de prueba se muestra en la figura 5.10.

En los cuadrantes de pruebas, las pruebas pueden dirigirse a las necesidades del negocio (usuario, cliente) o de la tecnología (desarrollador, equipo de fabricación). Algunas pruebas validan el comportamiento del software y, por tanto, respaldan el trabajo realizado por el equipo ágil; otros verifican (critican) el producto. Las pruebas pueden ser completamente manuales, completamente automatizadas, una combinación de

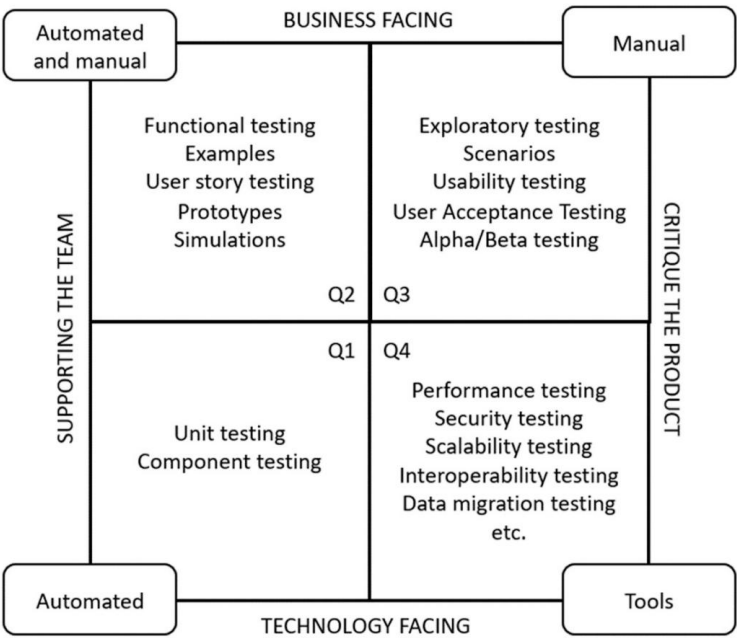


Fig. 5.10 Cuadrantes de prueba

manual y automatizado, o manual pero apoyado en herramientas. Los cuatro cuadrantes son los siguientes:

Cuadrante Q1 EI

cuadrante Q1 describe el nivel de prueba de los componentes, orientado a tecnología y soporte al equipo. Este cuadrante incluye pruebas de componentes. Estas pruebas deben automatizarse tanto como sea posible e integrarse en el proceso de integración continua.

Cuadrante Q2 EI

cuadrante Q2 describe el nivel de prueba del sistema, que se considera orientado al negocio y de apoyo al equipo. Este cuadrante incluye pruebas funcionales, ejemplos (ver Sección 4.5), pruebas de historias de usuario, prototipos y simulaciones. Estas pruebas verifican los criterios de aceptación y pueden ser manuales o automatizadas. A menudo se crean durante el desarrollo de historias de usuario, mejorando así su calidad. Son útiles al crear conjuntos de pruebas de regresión automatizadas.

Cuadrante Q3 EI

cuadrante Q3 describe un nivel de prueba de aceptación orientado al negocio que incluye pruebas de crítica de productos utilizando escenarios y datos realistas. Este cuadrante incluye pruebas exploratorias, pruebas basadas en escenarios (por ejemplo, pruebas basadas en casos de uso) o pruebas de flujo de procesos, pruebas de usabilidad y pruebas de aceptación del usuario, incluidas las pruebas alfa y beta. Estas pruebas suelen ser manuales y están orientadas al usuario.

Cuadrante Q4 EI

cuadrante Q4 describe un nivel de prueba de aceptación orientado a la tecnología que incluye pruebas de "crítica de producto". Este cuadrante contiene la mayoría de las pruebas no funcionales (excepto las pruebas de usabilidad). Estas pruebas suelen estar automatizadas.

Durante cada iteración, es posible que se requieran pruebas de cualquiera o todos los cuadrantes. Los cuadrantes de prueba se refieren a pruebas dinámicas en lugar de pruebas estáticas.

5.2 Gestión de riesgos

FL-5.2.1 (K1) Identificar el nivel de riesgo utilizando la probabilidad del riesgo y el impacto del riesgo

FL-5.2.2 (K2) Distinguir entre riesgos del proyecto y riesgos del producto FL-5.2.3 (K2)

Explicar cómo el análisis de riesgos del producto puede influir en la exhaustividad y alcance de las pruebas


FL-5.2.4 (K2) Explicar

qué medidas se pueden tomar en respuesta a los riesgos del producto analizados

Las organizaciones enfrentan muchos factores internos y externos que hacen que sea incierto si lograrán sus objetivos y cuándo [7]. La gestión de riesgos permite a las organizaciones aumentar la probabilidad de lograr sus objetivos, mejorar la calidad del producto y aumentar la confianza de las partes interesadas.

El nivel de riesgo generalmente se define por la probabilidad del riesgo y el impacto del riesgo (ver Sección 5.2.1). Los riesgos, desde el punto de vista del evaluador, se pueden dividir en dos grupos principales: riesgos del proyecto y riesgos del producto (ver Sección 5.2.2). Las principales actividades de gestión de riesgos incluyen:

- Análisis de riesgos (que consiste en la identificación y evaluación de riesgos; consulte la Sección 5.2.3)
- Control de riesgos (que consiste en mitigación de riesgos y seguimiento de riesgos; consulte la Sección 5.2.4)

Un enfoque de las pruebas en el que las actividades de prueba se gestionan, seleccionan y priorizado en función del análisis de riesgos y el control de riesgos se denomina prueba basada en riesgos .

Uno de los muchos desafíos en las pruebas es la selección y priorización adecuadas de las condiciones de prueba. El riesgo se utiliza para enfocar y asignar adecuadamente el esfuerzo requerido durante la prueba de estas condiciones de prueba. Se utiliza para decidir dónde y cuándo comenzar las pruebas e identificar áreas que necesitan más atención. Las pruebas se utilizan para reducir el riesgo, es decir, para reducir la probabilidad de un evento adverso o para reducir el impacto de un evento adverso.

Varias formas de pruebas son una de las actividades típicas de mitigación de riesgos en proyectos de desarrollo de software (ver Sección 5.2.4). Proporcionan retroalimentación sobre los riesgos identificados y restantes (no resueltos). El análisis temprano de riesgos del producto (ver Sección 5.2.3) contribuye al éxito del proyecto.

Las pruebas basadas en riesgos contribuyen a reducir los niveles de riesgo del producto y proporcionan información completa para ayudar a decidir si un producto está listo para su lanzamiento (este es uno de los principales objetivos de las pruebas; consulte la Sección 1.1.1). Normalmente, la relación entre el riesgo del producto y las pruebas se controla mediante un mecanismo de trazabilidad bidireccional.

Los principales beneficios de las pruebas basadas en riesgos son:

- Aumentar la probabilidad de descubrir defectos en orden de prioridad realizando pruebas en orden relacionado con la priorización de riesgos.
- Minimizar el riesgo residual del producto después del lanzamiento asignando el esfuerzo de prueba de acuerdo a riesgos
- Informar el riesgo residual midiendo los resultados de las pruebas en términos de niveles de riesgos relacionados.
- Contrarrestar los efectos de la presión del tiempo en las pruebas y permitir que el período de prueba se acorte con el menor aumento posible en el riesgo del producto asociado con una reducción en el tiempo asignado para las pruebas.

Para garantizar que se minimice la probabilidad de falla del producto, las actividades de prueba basadas en riesgos brindan un enfoque disciplinado para:

- Analizar (y reevaluar periódicamente) qué puede salir mal (riesgo)
- Determinar qué riesgos son lo suficientemente importantes como para abordarlos
- Implementar medidas para mitigar estos riesgos
- Diseñar planes de contingencia para hacer frente a los riesgos cuando se produzcan

Además, las pruebas pueden identificar nuevos riesgos, ayudar a determinar qué riesgos deben mitigarse y reducir la incertidumbre relacionada con los riesgos.

5.2.1 Definición de riesgo y atributos de riesgo

Según el Glosario de términos de prueba de ISTQB® [42], el riesgo es un factor que puede tener consecuencias negativas en el futuro; generalmente se describe por impacto y probabilidad (probabilidad). Por lo tanto, el nivel de riesgo está determinado por la probabilidad de que ocurra un evento adverso y el impacto o consecuencias (el daño resultante del evento). Esto suele definirse mediante la ecuación:

$$\text{Nivel de riesgo} = \text{Probabilidad de riesgo} \times \text{Impacto del riesgo}$$

Esta ecuación puede entenderse simbólicamente (como se describió anteriormente) o de manera bastante literal. Nos ocupamos del último caso en el llamado enfoque cuantitativo del riesgo, donde la probabilidad y el impacto se expresan numéricamente: la probabilidad como un número en el intervalo (0, 1), mientras que el impacto, en dinero. El impacto representa la cantidad de pérdida que sufriremos cuando el riesgo se materialice.

Ejemplo Una organización define un nivel de riesgo como el producto de su probabilidad e impacto. Se ha identificado el riesgo de que un informe clave se genere demasiado lentamente cuando hay una gran cantidad de usuarios conectados simultáneamente.

Dado que en este caso el requisito de un alto rendimiento es fundamental, se estimó que el impacto de este riesgo sería muy alto. Teniendo en cuenta la cantidad de usuarios que podrían experimentar retrasos en la generación de informes y las consecuencias de estos retrasos, el impacto del riesgo se estimó en 800.000 dólares.

Por otro lado, se estimó que la probabilidad de un rendimiento deficiente del programa era muy baja, debido a un proceso de prueba bastante bien definido, revisiones de arquitectura programadas y la amplia experiencia del equipo en pruebas de rendimiento. La probabilidad de riesgo se estimó en un 5%.

Finalmente, el nivel de riesgo según la ecuación anterior se fijó en:

$$5\% \times \$800\,000 = 0.05 \times \$800\,000 = \$40\,000$$

5.2.2 Riesgos del proyecto y riesgos del producto

Hay dos tipos de riesgos en las pruebas de software: riesgos del proyecto y riesgos del producto.

Riesgos del


proyecto Los riesgos del proyecto están relacionados con la gestión y el control del proyecto. Los riesgos del proyecto son riesgos que afectan el éxito del proyecto (la capacidad del proyecto para lograr sus objetivos).

Los riesgos del proyecto incluyen:

- **Problemas organizacionales** (p. ej., retrasos en la entrega de productos de trabajo, estimaciones inexactas, reducción de costos, procesos mal implementados, administrados y mantenidos, incluido el proceso de **ingeniería** de requisitos o control de calidad) • **Problemas humanos** (p. ej., **habilidades** insuficientes, conflictos, comunicación problemas, escasez de personal, falta de expertos en la materia disponibles) • **Problemas técnicos** (p. ej., **variación del alcance**; soporte deficiente de herramientas; estimación inexacta; cambios de último momento; clarificación insuficiente de los requisitos; incapacidad para cumplir con los requisitos debido a limitaciones de tiempo; falta de hacer que el entorno de prueba esté disponible a tiempo; programación tardía de la conversión de datos; migración o provisión de herramientas necesarias para esto; defectos en el proceso de fabricación, incluidas especificaciones de requisitos o casos de prueba u otras deudas técnicas;)
- **Problemas relacionados con proveedores** (por ejemplo, fallas en la entrega de terceros, quiebra del empresa de soporte, retraso en la entrega, problemas contractuales)

Los riesgos del proyecto, cuando ocurren, pueden afectar el cronograma, el presupuesto o el alcance del proyecto, lo que a su vez afecta la capacidad del proyecto para lograr sus objetivos. La consecuencia directa más común de los riesgos del proyecto es que el proyecto se retrasa, lo que trae consigo problemas adicionales, como mayores costos debido a la necesidad de asignar más tiempo a ciertas actividades o la necesidad de pagar sanciones contractuales por la entrega tardía de un producto.

Riesgos del

producto Los  riesgos del producto están relacionados con las características de calidad de un producto, como funcionalidad, confiabilidad, rendimiento, usabilidad y otras características descritas, por ejemplo, por el modelo de calidad ISO/IEC 25010 [5]. Los riesgos del producto ocurren dondequiera que un producto de trabajo (por ejemplo, una especificación, componente, sistema o prueba) pueda no satisfacer las necesidades de los usuarios y/o partes interesadas. Los riesgos del producto se definen por áreas de posible falla en el producto bajo prueba, ya que amenazan la calidad del producto de varias maneras. Ejemplos de riesgos del producto incluyen:

- Funcionalidad faltante o inadecuada • Cálculos incorrectos • Fallos durante el funcionamiento del software (p. ej., la aplicación se cuelga o se cierra inactivo)
- Arquitectura deficiente
- Algoritmos ineficientes •
- Tiempo de respuesta inadecuado
- Mala experiencia de usuario • Vulnerabilidades de seguridad

El riesgo del producto, cuando ocurre, puede tener varias consecuencias negativas, que incluyen:

- Insatisfacción del usuario final
- Pérdida de ingresos •
- Daños causados a terceros • Altos costos de mantenimiento

- Sobrecarga del servicio de asistencia técnica
- Pérdida de imagen
- Pérdida de confianza en el producto
- Sanciones penales

En casos extremos, la aparición de riesgos en el producto puede provocar daños físicos, lesiones e incluso la muerte.

5.2.3 Análisis de riesgos del producto

El propósito del análisis de riesgos es brindar conciencia sobre los riesgos para enfocar el esfuerzo de prueba de tal manera que se minimice el nivel de riesgo residual del producto. Idealmente, el análisis de riesgos del producto comienza temprano en el ciclo de desarrollo. El análisis de riesgos consta de dos fases principales: identificación de riesgos y evaluación de riesgos. La información de riesgo del producto obtenida en la fase de análisis de riesgo se puede utilizar para:

- Planificación de pruebas
- Especificación, preparación y ejecución de casos de prueba
- Monitoreo y control de pruebas

El análisis temprano de riesgos del producto contribuye al éxito de todo el proyecto, porque el análisis de riesgos del producto permite:

- Identificar niveles de prueba específicos y tipos de pruebas a realizar
- Definir el alcance de las pruebas a realizar
- Priorizar las pruebas (para detectar defectos críticos lo antes posible)
- Seleccionar técnicas de prueba apropiadas que lograrán de manera más efectiva la cobertura establecida y detectarán defectos asociados con los riesgos identificados
- Estimar para cada tarea la cantidad de esfuerzo puesto en las pruebas
- Determinar si se pueden utilizar otras medidas además de las pruebas para reducir (mitigar) riesgos
- Establecer otras actividades no relacionadas con las pruebas

Identificación de riesgos

La identificación de riesgos implica generar una lista completa de riesgos. Las partes interesadas pueden identificar riesgos utilizando diversas técnicas y herramientas, tales como:

- Lluvia de ideas (para aumentar la creatividad en la búsqueda de riesgos)
- Talleres de riesgos (para realizar conjuntamente la identificación de riesgos por parte de varias partes interesadas)
- Método Delphi o evaluación de expertos (para obtener el acuerdo o desacuerdo de los expertos en riesgo)
- Entrevistas (para preguntar a las partes interesadas sobre los riesgos y recopilar sus opiniones)
- Listas de verificación (para abordar los riesgos típicos, conocidos y que ocurren comúnmente)
- Bases de datos de proyectos pasados, retrospectivas y lecciones aprendidas (para abordar los riesgos pasados experiencia)

- Diagramas de causa y efecto (para descubrir riesgos mediante la realización de análisis de causa raíz) • Plantillas de riesgo (para determinar quién puede verse afectado por los riesgos y cómo)

Evaluación de

riesgos La evaluación de riesgos implica categorizar los riesgos identificados; determinar su probabilidad, impacto y nivel de riesgo; priorizarlos; y proponer formas de afrontarlos. La categorización ayuda a asignar acciones de mitigación, ya que normalmente los riesgos de la misma categoría se pueden mitigar utilizando un enfoque similar.

La evaluación de riesgos puede utilizar un enfoque cuantitativo o cualitativo o una combinación de ambos. En un enfoque cuantitativo, el nivel de riesgo se calcula como el producto de la probabilidad y el impacto. En un enfoque cualitativo, el nivel de riesgo se puede calcular utilizando una matriz de riesgo.

En la figura 5.11 se muestra un ejemplo de matriz de riesgos . La matriz de riesgo es una especie de "tabla de multiplicar", en la que el nivel de riesgo se define como el producto de ciertas categorías de probabilidad e impacto.

En la matriz de riesgo de la figura 5.11, hemos definido cuatro categorías de probabilidad: baja, media, alta y muy alta, y tres categorías de impacto: bajo, medio y alto. En la intersección de las categorías definidas de probabilidad e impacto, se define un nivel de riesgo. En esta matriz de riesgo se definen seis categorías de niveles de riesgo: muy bajo, bajo, medio, alto, muy alto y extremo. Por ejemplo, un riesgo con probabilidad media y alto impacto tiene un nivel de riesgo alto, porque

$$\text{probabilidad media impacto alto} = \text{nivel de riesgo alto:}$$

Hay otros métodos de riesgo en los que, como en el método de análisis modal de fallas y efectos (FMEA), por ejemplo, la probabilidad y el impacto se definen en una escala numérica (por ejemplo, del 1 al 5, donde 1 es el más bajo y 5 es el más bajo). categoría más alta) y el nivel de riesgo se calcula multiplicando los dos números. Por ejemplo, para una probabilidad de 2 y un impacto de 4, el nivel de riesgo se define como $2 * 4 = 8$.

En tales casos, sin embargo, se debe tener cuidado. La multiplicación es conmutativa, por lo que el nivel de riesgo de una probabilidad de 1 y un impacto de 5 es 5 y es el mismo que para un riesgo con una probabilidad de 5 y un impacto de 1, porque $1 * 5 = 5 * 1$. Sin embargo, en la práctica , el impacto es

RISK LEVEL		RISK LIKELIHOOD			
		low	medium	high	very high
RISK IMPACT	low	very low	low	medium	medium
	medium	low	medium	high	high
	high	medium	high	very high	extreme

Fig. 5.11 Matriz de riesgos

un factor mucho más importante para nosotros. Un riesgo con una probabilidad de 5 y un impacto de 1 quizás ocurra muy a menudo, pero prácticamente no causará problemas debido a su impacto insignificante. Por el contrario, un riesgo con una probabilidad de 1 y un impacto de 5 ciertamente tiene una probabilidad muy pequeña de ocurrir, pero cuando ocurre, sus consecuencias pueden ser catastróficas.

Además de los enfoques cuantitativos y cualitativos para la evaluación de riesgos, también existe un enfoque mixto. En este enfoque, la probabilidad y el impacto no se definen mediante valores numéricos específicos, sino que se representan mediante rangos de valores que expresan cierta incertidumbre sobre la estimación de estos parámetros. Por ejemplo, la probabilidad se puede definir como el intervalo [0,2, 0,4] y el impacto, como el intervalo [\$1000, \$3000]. Esto significa que esperamos que la probabilidad de riesgo esté entre el 20% y el 40%, mientras que se espera que su impacto esté entre \$1000 y \$3000. El nivel de riesgo en este enfoque también se puede especificar como un rango, según la siguiente fórmula de multiplicación:

$$\frac{1}{2} a, b \frac{1}{2} c, d = \frac{1}{2} ac, bd$$

es decir, en nuestro caso, el nivel de riesgo será

$$\frac{1}{2} 0,2, 0,4 \quad \frac{1}{2} \$1000, \$3000 = \frac{1}{2} \$200, \$1200 :$$

Por lo tanto, el nivel de riesgo está entre \$200 y \$1200.

El enfoque mixto es una buena solución si nos resulta difícil estimar los valores exactos de las probabilidades y los impactos de los riesgos individuales (enfoque cuantitativo), pero al mismo tiempo queremos resultados de evaluación de riesgos más precisos que los expresados en una escala ordinal. (enfoque cualitativo).

Ejemplo Una organización define un nivel de riesgo como el producto de su probabilidad e impacto. Se definen los siguientes riesgos, junto con estimaciones de la probabilidad y el impacto de cada riesgo:

Riesgo 1: probabilidad del 20%, impacto de 40.000 dólares

Riesgo 2: probabilidad del 10%, impacto de 100.000 dólares

Riesgo 3: probabilidad del 5%, impacto de 20.000 dólares

Para calcular el nivel de riesgo total sumamos los respectivos productos: Nivel de

$$\text{riesgo total} = 0,2 * \$40.000 + 0,1 * \$100.000 + 0,05 * \$20.000 = \$8000 + \$10.000 + \$1000 = \$19.000.$$

Esto significa que si no tomamos ninguna medida para minimizar (mitigar) estos riesgos, la pérdida promedio (esperada) en la que incurriremos como resultado de su posible ocurrencia será de aproximadamente \$19.000.


Tenga en cuenta que el análisis de riesgo cuantitativo para un solo riesgo no tiene mucho sentido, porque los riesgos (desde el punto de vista del evaluador o de un observador externo) son fenómenos aleatorios. No sabemos cuándo ocurrirán o si ocurrirán y, de ser así, qué tipo de daño causarán realmente. Un análisis de un único fenómeno aleatorio con probabilidad P e impacto X no tiene sentido, porque ocurrirá o no, por lo que la pérdida será 0 o X (asumiendo que nuestra estimación del impacto fue correcta) y

nunca seas P*X. Sin embargo, un análisis realizado para múltiples riesgos, como en el ejemplo anterior, tiene mucho más sentido. Esto se debe a que la suma de los niveles de riesgo para todos los riesgos identificados puede tratarse como el valor esperado de la pérdida en la que incurriremos debido a la ocurrencia de algunos de estos riesgos.

5.2.4 Control de riesgos del producto

El control de riesgos del producto juega un papel clave en la gestión de riesgos, ya que incluye todas las medidas que se toman en respuesta a los riesgos del producto identificados y evaluados. El control de riesgos se define como las medidas tomadas para mitigar y monitorear los riesgos a lo largo del ciclo de desarrollo. El control de riesgos es un proceso que incluye dos tareas principales: mitigación de riesgos y seguimiento de riesgos.

Mitigación de riesgos

Una vez analizados los riesgos, existen varias opciones para mitigarlos o posibles respuestas al riesgo [71,  72]:

- Aceptación de riesgos •
- Transferencia de
- riesgos • Plan de contingencia
- Mitigación de riesgos mediante pruebas

Aceptar (ignorar) riesgos puede ser una buena idea cuando se trata de riesgos de bajo nivel. Estos riesgos suelen tener la prioridad más baja y abordarse al final. A menudo, es posible que simplemente se nos acabe el tiempo para ocuparnos de ellos, pero esto no nos molesta especialmente. Esto se debe a que, incluso cuando se producen tales riesgos, el daño que provocan es menor o incluso insignificante. Por lo tanto, no tiene sentido perder el tiempo ocupándose de ellos cuando tenemos otras cuestiones mucho más importantes y serias que considerar (por ejemplo, riesgos de un nivel mucho más alto).

Un ejemplo de transferencia de riesgo podría ser, por ejemplo, la contratación de una póliza de seguro. A cambio de pagar la prima del seguro, se transfiere el riesgo de incurrir en el costo del riesgo a un tercero, en este caso, al asegurador.

Se desarrollan planes de contingencia para ciertos tipos de riesgos. Estos planes se desarrollan con el fin de evitar el caos y el retraso en la respuesta a los riesgos cuando ocurre una situación indeseable. Gracias a la existencia de este tipo de planes, cuando se produce un riesgo, todo el mundo sabe exactamente qué hacer. Por lo tanto, la respuesta al riesgo es rápida y bien pensada de antemano y, por tanto, eficaz.

En el programa de estudios de nivel básico, de los métodos de respuesta al riesgo anteriores, solo se presenta en detalle la mitigación de riesgos mediante pruebas, ya que el programa de estudios trata sobre pruebas. Se pueden diseñar pruebas para comprobar si realmente existe un riesgo. Si la prueba falla (provoca una falla, por lo que muestra un defecto), el equipo es consciente del problema. En consecuencia, esto mitiga el riesgo, que ya está identificado y no es desconocido. Si la prueba pasa, el equipo gana más confianza en la calidad del sistema. La mitigación de riesgos puede reducir la probabilidad o el impacto de un riesgo. En general, las pruebas contribuyen a reducir el riesgo general.

nivel de riesgo en el sistema. Cada prueba superada se puede interpretar como una situación en la que hemos demostrado que un riesgo específico (al que está asociada la prueba) no existe porque se supera la prueba. Esto reduce el riesgo residual total por el valor del nivel de riesgo de ese riesgo particular.

Las acciones que pueden tomar los evaluadores para mitigar el riesgo del producto son las siguientes:


- Seleccionar evaluadores con el nivel correcto de experiencia, apropiado para el tipo de riesgo
- Aplicar un nivel apropiado de independencia de las pruebas
- Realizar revisiones
- Realizar análisis estáticos
- Seleccionar técnicas de diseño de pruebas y niveles de cobertura apropiados
- Priorizar las pruebas según el nivel de riesgo
- Determinar el rango apropiado de pruebas de regresión

Algunas de las actividades de mitigación abordan las pruebas preventivas tempranas aplicándolas antes de que comiencen las pruebas dinámicas (por ejemplo, revisiones). En las pruebas basadas en riesgos, las actividades de mitigación de riesgos deben realizarse en todo el SDLC.

El paradigma de desarrollo ágil de software requiere la autoorganización del equipo (consulte la Sección [1.4.2](#) sobre roles de prueba y la Sección [1.5.2](#) sobre el enfoque de “todo el equipo”) al tiempo que proporciona prácticas de mitigación de riesgos que pueden verse como un sistema holístico de mitigación de riesgos. Una de sus fortalezas es su capacidad para identificar riesgos y proporcionar buenas prácticas de mitigación. Ejemplos de riesgos reducidos por estas prácticas son [69]:

- El riesgo de insatisfacción del cliente. En un enfoque ágil, el cliente o el representante del cliente ve el producto de forma continua, lo que, con una buena ejecución del proyecto, retroalimentación frecuente y comunicación intensiva, mitiga este riesgo.
- El riesgo de no completar todas las funciones. El lanzamiento frecuente y la planificación y priorización de productos reducen este riesgo al garantizar que los incrementos relacionados con altas prioridades comerciales se entreguen primero.
- El riesgo de una estimación y planificación inadecuadas. Las actualizaciones de los productos de trabajo se rastrean diariamente para garantizar el control de la gestión y oportunidades frecuentes de corrección.
- El riesgo de no resolver los problemas rápidamente. Un equipo autoorganizado, una gestión adecuada y los informes diarios de trabajo brindan la oportunidad de informar y resolver problemas diariamente.
- El riesgo de no completar el ciclo de desarrollo. En un ciclo determinado (cuanto más corto, mejor), el enfoque ágil entrega software funcional (o, más precisamente, una parte específica del mismo) de modo que no haya problemas importantes de desarrollo.
- El riesgo de asumir demasiado trabajo y cambiar las expectativas. La gestión del trabajo pendiente del producto, la planificación de iteraciones y los lanzamientos de trabajo evitan el riesgo de cambios inadvertidos que impactan negativamente en la calidad del producto al obligar a los equipos a enfrentar y resolver los problemas tempranamente.

Monitoreo de riesgos

El monitoreo de riesgos  es una tarea de gestión de riesgos que se ocupa de las actividades relacionadas con la verificación del estado de los riesgos del producto. El monitoreo de riesgos permite a los evaluadores medir las actividades de mitigación de riesgos implementadas (acciones de mitigación) para garantizar que estén logrando los efectos previstos e identificar eventos o circunstancias que crean riesgos nuevos o mayores. El seguimiento del riesgo se suele realizar a través de informes que comparan el estado real con lo esperado. Idealmente, el monitoreo de riesgos se lleva a cabo en todo el SDLC.

Las actividades típicas de monitoreo de riesgos implican revisar e informar sobre los riesgos del producto. Hay varios beneficios del monitoreo de riesgos, por ejemplo:


- Conocer el estado exacto y actual de cada riesgo del producto
- La capacidad de informar el progreso en la reducción del riesgo residual del producto
- Centrarse en los resultados positivos de la mitigación del riesgo
- Descubrir riesgos que no han sido identificados previamente
- Captar nuevos riesgos a medida que surgen
- Factores de seguimiento que afectan los costos de gestión de riesgos

Algunos enfoques de gestión de riesgos tienen incorporados métodos de seguimiento de riesgos. Por ejemplo, en el método AMEF antes mencionado, la estimación de la probabilidad y el impacto de un riesgo se realiza dos veces: primero en la fase de evaluación del riesgo y luego, nuevamente, después de implementar acciones para mitigar ese riesgo.

5.3 Monitoreo de pruebas, control de pruebas y finalización de pruebas


FL-5.3.1 (K1) Recordar las métricas utilizadas para las pruebas

FL-5.3.2 (K2) Resumir los propósitos, el contenido y las audiencias de los informes de prueba FL-5.3.3 (K2) Ejemplificar cómo comunicar el estado de las pruebas

El objetivo principal del monitoreo de pruebas es recopilar  y compartir información para obtener información sobre las actividades de prueba y visualizar el proceso de prueba. La información monitoreada se puede recopilar de forma manual o automática, utilizando herramientas de gestión de pruebas. Más bien sugerimos un enfoque automático, utilizando el registro de pruebas y la información de la herramienta de informe de defectos, respaldado por un enfoque manual (control directo del trabajo del evaluador; esto permite una evaluación más objetiva del estado actual de las pruebas). Sin embargo, no olvide recopilar información manualmente, por ejemplo, durante las reuniones diarias de Scrum (ver Sección 2.1.1).

Los resultados obtenidos se aplican a:

- Medición del cumplimiento de los criterios de salida, por ejemplo, logro de la cobertura de riesgo asumido del producto, requisitos y criterios de aceptación.
- Evaluación del progreso del trabajo en comparación con el cronograma y el presupuesto.

Actividades realizadas como parte del control de pruebas.  son principalmente los siguientes:

- Tomar decisiones basadas en la información obtenida del monitoreo de pruebas • Volver a priorizar las pruebas cuando se materialicen los riesgos identificados (por ejemplo, no entregar el software a tiempo) • Realizar cambios en el cronograma de ejecución de pruebas • Evaluar la disponibilidad o indisponibilidad del entorno de prueba u otros recursos

5.3.1 Métricas utilizadas en las pruebas

Durante y después de un nivel de prueba determinado, se pueden (y deben) recopilar métricas que permitan estimar:

- El progreso de la implementación del cronograma y presupuesto • El nivel de calidad actual del objeto de prueba • La idoneidad del enfoque de prueba elegido • La efectividad de las actividades de prueba desde el punto de vista de lograr el objetivos

El monitoreo de pruebas recopila varias métricas para respaldar las actividades de control de pruebas. Típico Las métricas de prueba incluyen:

- Métricas del proyecto (por ejemplo, finalización de tareas, utilización de recursos, esfuerzo de prueba, porcentaje de finalización del trabajo de preparación del entorno de prueba planificado, fechas de hitos)
- Métricas de casos de prueba (p. ej., progreso de implementación de casos de prueba, progreso de preparación del entorno de prueba, número de casos de prueba ejecutados/no ejecutados, aprobados/fallidos, tiempo de ejecución de la prueba)
- Métricas de calidad del producto (por ejemplo, disponibilidad, tiempo de respuesta, tiempo medio hasta la falla) • Métricas de defectos (por ejemplo, número de defectos encontrados/repuestos, prioridades de defectos, densidad de defectos (número de defectos por unidad de volumen, por ejemplo, 1000 líneas de código) , frecuencia de defectos (número de defectos por unidad de tiempo), porcentaje de defectos detectados, porcentaje de pruebas de confirmación exitosas) • Métricas de riesgo (p. ej., nivel de riesgo residual, prioridad de riesgo) • Métricas de cobertura (p. ej., cobertura de requisitos, cobertura de historias de usuario, aceptación cobertura de criterios de prueba, cobertura de condiciones de prueba, cobertura de código, cobertura de riesgos) • Métricas de costos (por ejemplo, costo de pruebas, costo organizacional de calidad, costo promedio de ejecución de pruebas, costo promedio de reparación de defectos)

Tenga en cuenta que diferentes métricas tienen diferentes propósitos. Por ejemplo, desde un punto de vista puramente de gestión, un gerente estará interesado en saber cuántas ejecuciones de pruebas planificadas se realizaron o si no se superó el presupuesto planificado.

Sin embargo, desde el punto de vista de la calidad del software, serán más importantes cuestiones como, por ejemplo, el número de defectos encontrados según su grado de criticidad, el tiempo medio entre fallos, etc. Ser capaz de elegir el conjunto adecuado de métricas para medir Es un arte, ya que normalmente es imposible medirlo todo. Además, cuantos más datos haya en el

informes, menos legibles se vuelven. Debes elegir un pequeño conjunto de métricas a medir, que al mismo tiempo te permitirán responder todas las preguntas sobre los aspectos del proyecto que te interesan. La técnica Goal-Question-Metric (GQM), por ejemplo, puede ayudar a desarrollar dicho plan de medición, pero no lo presentamos aquí, ya que está más allá del alcance del programa de estudios.

5.3.2 Propósito, contenido y destinatario de los informes de prueba

Los informes de prueba se utilizan para resumir y proporcionar información sobre las actividades de prueba (por ejemplo, pruebas en un nivel determinado) tanto durante como después de su ejecución. Un informe de prueba elaborado durante la ejecución de una actividad de prueba se denomina informe de progreso de la prueba, y el informe elaborado al finalizar dicha actividad, informe de finalización de la prueba.

Ejemplos de estos informes se muestran en las Figs. 5.12, 5.13 y 5.14.

Según la norma ISO/IEC/IEEE 29119-3 [3], un informe de prueba típico debe incluir:

- Resumen de las pruebas realizadas • Descripción de lo que sucedió durante el período de prueba • Información sobre desviaciones del plan • Información sobre el estado de las pruebas y la calidad del producto, incluida información sobre cumplir con los criterios de salida (o Definición de Hecho)
- Información sobre los factores que bloquean las pruebas.

Summary Status Test Report for: New subscription system (NSS) Vers.: Iteration 3

Covers: Complete NSS iteration 3 results.

Progress against Test Plan: Test has been done in the iteration on the 5 user stories for this iteration. For the one high risk story 92 % statement coverage was achieved, and for the others 68 % statement coverage was achieved on average.

There are no outstanding defects of severity 1 and 2, but the showcase showed that the product has 16 defects of severity 3.

Factors blocking progress: None

Test measures: 6 new test procedures have been developed, and 2 of the other test procedures have been changed.

The testing in the iteration has taken up approx. 30 % of the time. The test took about 2½ hours.

New and changed risks: The risks for the stories have been mitigated satisfactorily. New risks are not identified yet.

Planned testing: As per test plan.

Backlog added: 16 defects (severity 3)

Fig. 5.12 Informe de progreso de pruebas en una organización ágil (según ISO/IEC/IEEE 29119-3)

- Medidas relacionadas con defectos, casos de prueba, cobertura de prueba, progreso del trabajo y recursos. utilización
- Información sobre riesgos residuales •

Información sobre productos de trabajo para su reutilización

Un informe de progreso de prueba típico incluye además información sobre:

- Estado de las actividades de prueba y progreso del plan de prueba. •

Pruebas programadas para el próximo período de informe.

Los informes de prueba deben adaptarse tanto al contexto del proyecto como a las necesidades del público objetivo. Deben incluir:

- Información detallada sobre tipos de defectos y tendencias relacionadas, para audiencia técnica. •

Resumen del estado de los defectos por prioridad, presupuesto y cronograma, así como aprobados.

falló y bloqueó casos de prueba, para las partes interesadas del negocio

Los principales destinatarios del informe de progreso de las pruebas son aquellos que pueden realizar cambios en la forma en que se realizan las pruebas. Estos cambios pueden incluir agregar más

Test report for: New subscription system (NSS) **Vers.:** Iteration 3361

Covers: NSS final iteration result, including result of previous iterations, in preparation for a major customer delivery (for use).

Risks: The live data risk was retired by creation of a simulated database using historic live data "cleaned" by the test team and customer.

Test Results: Customer accepted this release of the product based on:

16 user stories were successful, including one added after the last status report.

100% statement coverage was achieved in technology-facing testing with the one high risk story, and for the others 72% statement coverage was achieved on average.

Team accepted the backlog on 4 defects of severity 3.

Showcase was accepted by the customer with no added findings. Showcase demo iteration features interfaced with "live" data.

Performance of the iteration features was found to be acceptable by team and customer.

New, changed, and residual risks: Security of the system could become an issue in future releases, assuming a follow work activity is received from the customer.

Notes for future work from retrospective:

Iteration team feels a new member could be needed given possible new risk since no one has knowledge in this area.

Severity 3 defects that move on to backlog should be addressed in next release to reduce technical debt.

The modified live data worked well and should be maintained.

Test automation and exploratory testing is working, but additional test design techniques should be considered, e.g. security and combinatorial testing.

Fig. 5.13 Informe resumido de pruebas en una organización ágil (según ISO/IEC/IEEE 29119-3)

Project PC-part of the UV/TRT-14 33a product**System Test Completion Report, V 1.0, 29/11/2019****Written by:** Carlo Titlefsen (Test Manager)**Approved by:** Benedicte Rytter (Project Manager)**Summary of testing performed:**

- The test specification was produced; it included 600 test procedures.
- The test environment was established according to the plan.
- Test execution and recording were performed according to the plan.
- Based on the outcomes of testing, it is recommended that UAT commence.

Deviations from planned testing: The Requirement Specification was updated during the test to v5.6. This entailed rewriting of a few test cases, but this did not have an impact on the schedule.

Test completion evaluation: All test procedures were executed and passed testing, except 3. There are no remaining severity 1 or 2 defects, and all open severity 3 and 4 defects have been signed off by the business. The remaining 3 failed test cases will be rerun during UAT, once remaining severity 3 and 4 defects have been fixed. The requirements for these 3 failed tests are low risk.

Factors that blocked progress: Deployment was delayed on 5 occasions, preventing testing from starting on time. This equated to an increase in testing duration by 25 person days.

Test measures:

Three test procedures out of 605 were not passed due to defects. All test procedures that were run had passed at the end of the planned 3 weeks of testing.

During test execution, 83 defects were detected and 80 were closed.

Duration of test design and execution:

- 12,080 working hours were spent on the production of test cases and test procedures
- 10 working hours were spent on the establishment of the test environment
- 15,040 working hours were spent on test execution and defect reporting
- One hour was spent on the production of this report.

Residual risks: All the risks listed in the test plan have been eliminated, except the one with the lowest exposure, Risk # 19. This risk has been deemed acceptable by the Business Representative.

Test deliverables: All deliverables specified in the plan have been delivered to the common CM-system according to the procedure.

Reusable test assets: The test specification and the related test data and test environment requirements can be reused for maintenance testing, if and when this is needed.

Lessons learned: The test executors should have been given a general introduction to the system; it sometimes delayed them finding out how to perform a test case, but fortunately the test analyst was available throughout the test period.

Fig. 5.14 Informe resumido de prueba en una organización tradicional (según ISO/IEC/IEEE 29119)

probar recursos o incluso realizar cambios en el plan de prueba. Por lo tanto, los miembros típicos de este grupo son directores de proyectos y propietarios de productos. También puede ser útil incluir en este grupo a personas responsables de los factores que inhiben el progreso de la prueba, para que quede claro que son conscientes del problema. Finalmente, el equipo de prueba también debería

ser incluidos en este grupo, ya que esto les permitirá ver que su trabajo es apreciado y les ayudará a comprender cómo su trabajo contribuye al progreso general.

La audiencia del informe de finalización de la prueba se puede dividir en dos grupos principales: los responsables de tomar decisiones (qué hacer a continuación) y los que realizarán pruebas en el futuro utilizando la información del informe. Quienes toman las decisiones variarán según las pruebas que se informen (por ejemplo, nivel de prueba, tipo de prueba o proyecto) y pueden decidir qué pruebas deben incluirse en la siguiente iteración o si implementar un objeto de prueba dado el riesgo residual informado. El segundo grupo son aquellos que son responsables de las pruebas futuras del objeto de prueba (por ejemplo, como parte de las pruebas de mantenimiento) o aquellos que de otro modo pueden decidir sobre la reutilización de los productos a partir de las pruebas informadas (por ejemplo, datos de prueba limpiados del cliente en un proyecto separado). . Además, todas las personas que aparecen en la lista de distribución original del plan de prueba también deben recibir una copia del informe de finalización de la prueba.

5.3.3 Comunicación del estado de las pruebas

Los medios para comunicar el estado de las pruebas varían, dependiendo de las preocupaciones, expectativas y visión de la gestión de pruebas, las estrategias de prueba de la organización, los estándares regulatorios o, en el caso de equipos autoorganizados (ver Sección 1.5.2), el equipo mismo. Las opciones incluyen, en particular:

- Comunicación verbal con los miembros del equipo y otras partes interesadas.
- Paneles de control, como paneles de CI/CD, tableros de tareas y gráficos de evolución •
- Canales de comunicación electrónicos (p. ej., correos electrónicos, chats)
- Documentación en línea •

Informes de prueba formales (consulte la Sección 5.3.2)

Se pueden utilizar una o más de estas opciones. Una comunicación más formal puede ser más apropiada para equipos distribuidos, donde la comunicación directa cara a cara no siempre es posible debido a diferencias geográficas o horarias.

La Figura 5.15 muestra un ejemplo de un tablero típico para comunicar información sobre el proceso de integración y entrega continua. Desde este panel podrás obtener muy rápidamente la información básica más importante sobre el estado del proceso, por ejemplo:

- El número de lanzamientos por estado de lanzamiento (exitoso, no creado, abortado, etc.) • Las fechas y el estado de los últimos diez lanzamientos •
- Información sobre las últimas diez confirmaciones •
- Frecuencia de los lanzamientos
- Tiempo promedio para crear un lanzamiento como función de tiempo

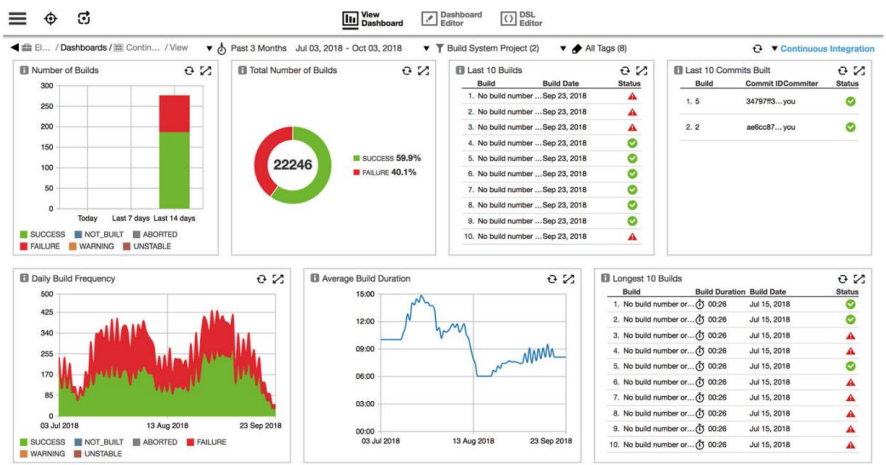


Fig. 5.15 Ejemplo de un panel de CI/CD (fuente: <https://devops.com/electric-cloud-extends-continuity-delivery-platform/>)

5.4 Gestión de la configuración

FL-5.4.1 (K2) Resumir cómo la gestión de la configuración respalda las pruebas

El objetivo principal de la gestión de la configuración es garantizar y mantener la integridad del componente/ sistema y el software de prueba y las interrelaciones entre ellos durante todo el ciclo de vida del proyecto y del producto. La gestión de la configuración tiene como objetivo garantizar que:

- Todos los elementos de configuración, incluidos los objetos de prueba, los elementos de prueba (partes individuales de los objetos de prueba) y otro software de prueba, han sido identificados, controlados en versión, rastreados para detectar cambios y vinculados entre sí de una manera que mantiene la trazabilidad en todas las etapas del proceso. proceso de prueba • Toda la documentación y los elementos de software identificados se mencionan explícitamente en la documentación de prueba.

Hoy en día, se utilizan herramientas apropiadas para la gestión de la configuración, pero es importante tener en cuenta que los procedimientos de gestión de la configuración junto con la infraestructura (herramientas) necesaria deben identificarse e implementarse en la etapa de planificación, ya que el proceso cubre todos los productos de trabajo que ocurren dentro de el proceso de desarrollo.

La Figura 5.16 muestra una representación gráfica de un determinado repositorio almacenado en el sistema de versiones de código git. Los vértices C0, C1, ..., C6 indican las llamadas instantáneas, es decir, versiones sucesivas del código fuente. Las flechas indican en función de qué versión se creó la siguiente versión.

Por ejemplo, la confirmación C1 se creó sobre la base de la confirmación C0 (por ejemplo, un desarrollador descargó el código C0 a su computadora local, le hizo algunos cambios y

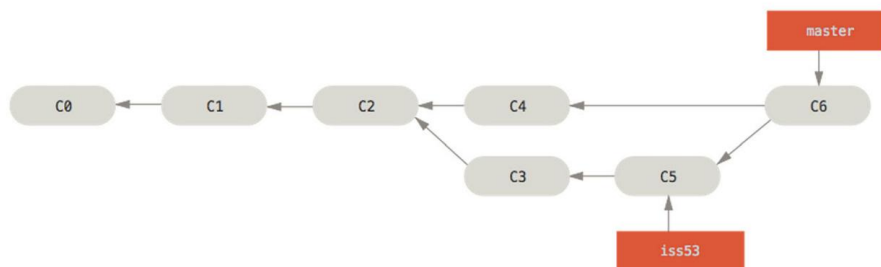


Fig. 5.16 Representación gráfica de un repositorio de código en el sistema Git (fuente: <https://git-scm.com/docs/gittutorial>)

lo comprometí en el repositorio). El repositorio distingue entre estas dos versiones del código. Es posible que otros dos desarrolladores hayan creado las confirmaciones C3 y C4 de forma independiente, basándose en la misma confirmación C2. El rectángulo "maestro" representa la rama maestra. La confirmación C5, por otro lado, representa la rama "iss53", donde se creó el código como resultado de corregir un error detectado en la versión C3 del código. La confirmación C6 se creó como resultado de una combinación de cambios realizados de forma independiente en las confirmaciones C5 (reparación de defectos) y C4 (por ejemplo, agregar una nueva característica basada en la versión C2 del código).

Al utilizar un sistema de control de versiones como git, los desarrolladores tienen control total sobre su código. Pueden rastrear, guardar y, si es necesario, recrear cualquier versión histórica del código. Si se comete un error o falla en alguna compilación, siempre es posible deshacer los cambios y volver a la versión anterior que funciona.

Hoy en día, el uso de herramientas de control de versiones de código es el estándar de facto. Sin estas herramientas, el proyecto se convertiría en un caos muy rápidamente.

Ejemplo Consideremos un ejemplo muy simplificado de la aplicación de la gestión de configuración en la práctica. Supongamos que el sistema que estamos produciendo consta de tres componentes: A, B y C. El sistema es utilizado por dos de nuestros clientes: C1 y C2.

La siguiente secuencia de eventos muestra el historial de cambios en el repositorio de código y el historial de lanzamientos de software para los clientes. Suponemos que la creación de una determinada versión de la aplicación siempre tiene en cuenta las últimas versiones de los componentes incluidos en la misma.

1. Cargando la versión 1.01 del componente A al repositorio
2. Cargando la versión 1.01 del componente B al repositorio
3. Cargando la versión 1.02 del componente A (después de la eliminación del defecto) al repositorio
4. Cargando la versión 1.01 del componente C al repositorio
5. Crear la versión 1.0 de la aplicación para su lanzamiento y enviarla a C1
6. Cargar la versión 1.02 del componente C al repositorio (agregar nuevo.

funcionalidad)

7. Crear la versión 1.01 de la aplicación para su lanzamiento y enviarla a C1
8. Cargar la versión 1.03 del componente A al repositorio (agregar nuevos funcionalidad)
9. Crear la versión 1.02 de la aplicación para su lanzamiento y enviarla al C2.

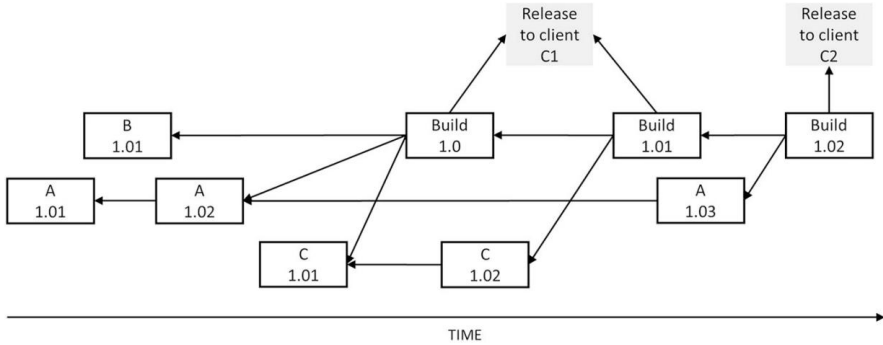


Fig. 5.17 Historia de la creación de versiones sucesivas de componentes y lanzamientos de software.

Este proceso se representa gráficamente en la figura 5.17.

Ahora supongamos que en este punto (después del paso 9), el cliente C1 informó una falla. Si no tuviéramos establecido un proceso de gestión de configuración, no sabríamos en qué versiones de los componentes A, B y C buscar defectos potenciales. Supongamos que el cliente nos envió información de que el problema se observó en la versión 1.01 del software. La herramienta de gestión de configuración, basándose en esta información y en los datos anteriores, puede reconstruir los componentes de software individuales que se incluyeron en la versión de software 1.01 entregada al cliente C1 en el paso 7. Además, la herramienta también puede reproducir versiones adecuadas de los casos de prueba utilizados para el software en la versión 1.01, las versiones adecuadas de los componentes del entorno, bases de datos, etc. que se utilizaron para crear la versión 1.01 del software. En particular, analizando la secuencia de eventos anterior, vemos que la versión de software 1.01 consta de:

- Componente A versión 1.02 •
- Componente B versión 1.01 •
- Componente C versión 1.02

Esto significa que el defecto potencial está en uno (o más) de estos tres componentes.

Tenga en cuenta que si se encuentra que el defecto está en el componente B, después de solucionarlo y crear la versión 1.02, se debe enviar una nueva versión del software (1.03) no solo al cliente C1 sino también al cliente C2, ya que la versión actual del software utilizado por C2 utiliza el componente B defectuoso en la versión 1.01.


5.5 Gestión de defectos

FL-5.5.1 (K3) Preparar un informe de defectos

Uno de los objetivos de las pruebas es encontrar defectos y, por lo tanto, todos los defectos encontrados deben registrarse. La forma en que se informa un defecto depende del contexto de prueba del componente o sistema, el nivel de prueba y el modelo SDLC elegido. Cada defecto identificado debe

ser investigado y rastreado desde el momento en que se detecta y clasifica hasta que se resuelve el problema.

La organización debe implementar un proceso de gestión de defectos, cuya formalización puede variar desde un enfoque informal hasta uno muy formal. Es posible que algunos informes describan situaciones de falsos positivos en lugar de fallas reales causadas por defectos. Los evaluadores deben intentar minimizar la cantidad de resultados falsos positivos que se informan como defectos; sin embargo, en proyectos reales, un cierto número de defectos reportados son falsos positivos.

Los objetivos principales del informe de defectos.  son:

- Comunicación:
 - Proporcionar a los fabricantes información sobre la incidencia para que puedan identificar, aislar y reparar el defecto si fuera necesario.
 - Proporcionar a la gestión de pruebas información actualizada sobre la calidad de las sistema bajo prueba y el progreso de las pruebas.
- Permitir la recopilación de información sobre el estado del producto bajo prueba:
 - Como base para la toma de decisiones.
 - Para identificar los riesgos tempranamente
 - Como base para el análisis post-proyecto y la mejora del desarrollo.
procedimientos

En el estándar ISO/IEC/IEEE 29119-3 [3], los informes de defectos se denominan informes de incidentes. Se trata de una nomenclatura un poco más precisa, porque no toda anomalía observada tiene por qué significar inmediatamente un problema a resolver: un evaluador puede cometer un error y considerar algo que es perfectamente correcto como un defecto que debe corregirse. Estas situaciones generalmente se analizan durante el ciclo de vida del defecto y, por lo general, dichos informes se rechazan, con un estado de "no es un defecto" o similar.

En la figura 5.18 se muestra un ejemplo de ciclo de vida de un defecto . Desde el punto de vista del evaluador, sólo los estados Rechazado, Modificado, Aplazado y Cerrado significan que se ha completado el análisis de defectos; el estado Corregido significa que el desarrollador ha solucionado el defecto, pero solo se puede cerrar después de volver a realizar la prueba.

El informe de defectos para pruebas dinámicas debe incluir:

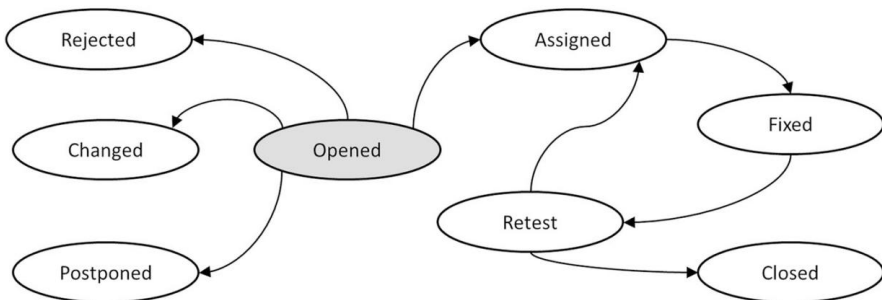


Fig. 5.18 Un ejemplo de ciclo de vida de defectos y estados de defectos

• Identificador único • Título

y un breve resumen de la anomalía reportada • Fecha del informe (la fecha en que se descubrió la anomalía) • Información sobre el autor del informe de defecto

• Identificación del elemento bajo prueba • Fase del ciclo de desarrollo de software en la que se se observó

anomalía • Descripción de la anomalía para permitir su reproducción y eliminación, incluyendo cualquier

tipo de registros, volcados de bases de datos, capturas de pantalla o

grabaciones • Resultados reales y

esperados • Prioridad de eliminación de defectos (si

se conoce) • Estado (p. ej., abierto, aplazado, duplicado, reabierto, etc.) •

Descripción de la no conformidad para ayudar a determinar la causa del problema • Urgencia de la

solución (si se conoce) • Identificación de un elemento

de configuración del sistema o software • Conclusiones y recomendaciones •

Historial de cambios • Referencias a otros elementos,

incluido el caso de prueba a

través del cual se resolvió el problema

fue revelado

En la figura 5.19 se muestra un ejemplo de informe de defecto .

Las partes interesadas en los informes de defectos incluyen evaluadores, desarrolladores, gerentes de pruebas y gerentes de proyectos. En muchas organizaciones, el cliente también obtiene acceso al sistema de gestión de defectos, especialmente en el periodo inmediatamente posterior a la liberación.

Sin embargo, se debe tener en cuenta que el cliente típico no está familiarizado con las reglas de notificación de defectos. Además, es típico que desde el punto de vista del cliente, casi todos los defectos sean críticos, porque impiden el trabajo del cliente.

Preguntas de muestra

Pregunta 5.1

(FL-5.1.1, K2)

¿Cuál de los siguientes NO es parte del plan de prueba?

- A. Estrategia de prueba.
- B. Restricciones presupuestarias.
- C. Alcance de las pruebas.
- D. Registro de riesgos.

Elija una respuesta.

Pregunta 5.2

(FL-5.1.2, K2)

¿Cuál de las siguientes acciones realiza el evaluador durante la planificación del lanzamiento?

- A. Realizar análisis de riesgos detallados para historias de usuarios.
- B. Identificar aspectos no funcionales del sistema a probar.

Incident Registration Form			
Number	278		
Short Title	Information truncated		
Software product	Project PC-part of the UV/TIT-14 33a product		
Version (n.m)	5.2		
Status = Created			
Registration created by	Heather Small	Date & time	14 th May
Anomaly observed by	Heather Small	Date & time	14 th May
Comprehensive description	<i>The text in field "Summary" is truncated after 54 characters; it should be able to show 75 characters.</i>		
Observed during	Walk through / Review / Inspection / Code & Build / Test / Use		
Observed in	Requirement / Design / Implementation / Test / Operation		
Symptom	Oper. system crash / Program hang-up / Program crash / Input / Output / Total product failure / System error / Other.		
User impact	High / Medium / Low		
User urgency	Urgent / High / Medium / Low / None		

Fig. 5.19 Ejemplo de informe de defectos según ISO/IEC/IEEE 29119-3

C. Estimar el esfuerzo de prueba para nuevas funciones planificadas en una iteración determinada.

D. Definición de historias de usuario y sus criterios de aceptación.

Elija una respuesta.

Pregunta 5.3

(FL-5.1.3, K2)

Considere los siguientes criterios de entrada y salida:

i. Disponibilidad de probadores. ii.

Sin defectos críticos abiertos. III.

Cobertura del 70% del estado de cuenta lograda en las pruebas de componentes.

IV. Todas las pruebas de humo realizadas antes de la prueba del sistema han pasado.

¿Cuáles de los anteriores son criterios de entrada y cuáles son criterios de salida para las pruebas?

A. (i), (iii) y (iv) son los criterios de entrada; (ii) es el criterio de salida.

B. (ii) y (iii) son los criterios de entrada; (i) y (iv) son los criterios de salida.

- C. (i) y (ii) son los criterios de entrada; (iii) y (iv) son los criterios de salida.
- D. (i) y (iv) son los criterios de entrada; (ii) y (iii) son los criterios de salida.

Elija una respuesta.

Pregunta 5.4

(FL-5.1.4, K3)

El equipo utiliza el siguiente modelo de extrapolación para la estimación del esfuerzo de prueba:

$$E_{n+1} = \frac{E_n \cdot p - 1 E_{n-1} \cdot p - 2 E_{n-2} \cdot p - 3}{3},$$

donde E(n) es el esfuerzo en la enésima iteración. El esfuerzo en la enésima iteración es, por tanto, el promedio del esfuerzo de las últimas tres iteraciones.

En las primeras tres iteraciones, el esfuerzo real fue de 12, 15 y 18 días-persona, respectivamente. El equipo acaba de completar la tercera iteración y quiere utilizar el modelo anterior para estimar el esfuerzo de prueba necesario en la QUINTA iteración. ¿Cuál debería ser la estimación hecha por el equipo?

- A. 24 días-persona.
- B. 16 días-persona.
- C. 15 días-persona.
- D. 21 días-persona.

Elija una respuesta.

Pregunta 5.5

(FL-5.1.5, K3)

Quiere priorizar los casos de prueba para lograr un orden de ejecución de prueba óptimo. Utiliza un método de priorización basado en cobertura adicional. La cobertura de funciones se utiliza como métrica de cobertura. Hay siete funciones en el sistema, denominadas A, B, C, D, E, F y G. La tabla 5.4 muestra qué funciones cubre cada caso de prueba: ¿Qué caso de prueba se ejecutará TERCERO en orden?

- A.TC5.
- B. TC2.
- C.TC4.
- D.TC3.

Elija una respuesta.

Tabla 5.4 Cobertura de funciones por casos de prueba

Caso de prueba	Funciones cubiertas
TC1	A, B, C, F
TC2	D
TC3	A, F, G
TC4	mi
TC5	re, g