

Probador Certificado de ISTQB®

Programa de Estudio

Nivel Avanzado

Gestión de Prueba

Versión ES V00.90

Traducción realizada por

Spanish Software Testing Qualifications Board

Traducción del Programa de Estudio

“ISTQB® Certified Tester – Advanced Level – Test Management, Version V3.0”



Nota sobre Derechos de Propiedad Intelectual

Nota sobre Derechos de Propiedad Intelectual (Copyright) © International Software Testing Qualifications Board (en adelante denominado ISTQB®) ISTQB® es una marca registrada de International Software Testing Qualifications Board.

Nota sobre Derechos de Propiedad Intelectual (Copyright) © 2023, los autores de la actualización 3.0 son Horst Pohlmann (propietario de producto, vicepresidente del AELWG), Tauhida Parveen, Francis Fenner, Laura Albert, Matthias Hamburg, Maud Schlich, Tanja Tremmel, Ralf Bongard, Erik van Veenendaal, Jan Giessen, Bernd Freimut, Andreas Neumeister, Georg Sehl, Rabi Arabi, Therese Kuhfuß, Ecaterina Irina Manole, Veronica Belcher, Kenji Onishi, Pushparajan Balasubramanian, Meile Postuma y Miroslav Renda.

Nota sobre Derechos de Propiedad Intelectual (Copyright) © 2010-2012 de los autores para el subgrupo de trabajo Advanced Level Test Manager: Rex Black (presidente), Judy McKay (vicepresidenta), Graham Bath, Debra Friedenber, Bernard Homès, Kenji Onishi, Mike Smith, Geoff Thompson, Tsuyoshi Yumoto.

Todos los derechos reservados. Por la presente, los autores transfieren los derechos de autor al ISTQB®. Los autores (como actuales titulares de los derechos de autor) y el ISTQB® (como futuro titular de los derechos de autor) han acordado las siguientes condiciones de uso:

- Se podrán copiar extractos de este documento, para uso no comercial, siempre que se cite la fuente. Cualquier Proveedor de Formación Acreditado puede utilizar este programa de estudio como fuente para un curso de formación si los autores y el ISTQB® son reconocidos como fuente y propietarios de los derechos de autor del programa de estudio y siempre que cualquier anuncio de dicho curso de formación pueda mencionar el programa de estudio sólo después de haber recibido la acreditación oficial de los materiales de formación por parte de un Comité Miembro reconocido por el ISTQB®.
- Cualquier persona o grupo de personas puede utilizar este programa de estudio como fuente para artículos y libros, si los autores y el ISTQB® son reconocidos como la fuente y los propietarios de los derechos de autor del programa de estudio.
- Cualquier otro uso de este programa de estudio está prohibido sin la aprobación previa por escrito del ISTQB®.
- Cualquier Comité Miembro reconocido por el ISTQB® puede traducir este programa de estudio siempre y cuando reproduzca el mencionado Nota de Derechos de Propiedad Intelectual (Copyright) en la versión traducida del programa de estudio.

Historial de Revisiones

Versión	Fecha	Observaciones
ISEB v1.1	2001/09/04	ISEB Practitioner Syllabus
ISTQB 1.2E	2003/09	ISTQB Advanced Level Syllabus from EOQ-SG
V2007	2007/10/12	Certified Tester Advanced Level syllabus version 2007
D100626	2010/06/10	Incorporation of changes as accepted in 2009, separation of each chapter for the separate modules
D101227	2010/12/10	Acceptance of changes to format and corrections that have no impact on the meaning of the sentences.
D2011	2011/10/31	Change to split syllabus, re-worked LOs and text changes to match LOs. Addition of BOs.
Alpha 2012	2012/02/09	Incorporation of all comments from NBs received from October release.
Beta 2012	2012/03/26	Incorporation of comments from NBs received on time from Alpha release.
Beta 2012	2012/04/07	Beta version submitted to GA
Beta 2012	2012/06/08	Copy edited version released to NBs
Beta 2012	2012/06/27	EWG and Glossary comments incorporated
RC 2012	2012/08/15	Release candidate version – final NB edits included
Beta v3.0	2023/10/31	Incorporation of all comments from member boards received for all sections (BOIncs) from the Alpha review
POST Beta v3.0	2024/01/31	Incorporation of all comments from member boards received for all sections (BOIncs) from the Beta review
POST Beta v3.0	2024/02/29	Minor modifications in proofreading
RC v3.0	2024/03/28	Release candidate version – latest formal template changes included as proposed by PWG
V.3.0	2024/05/03	Rework after release; only typos and inconsistencies eliminated
ISEB v1.1	2001/09/04	ISEB Practitioner Syllabus
ISTQB 1.2E	2003/09	ISTQB Advanced Level Syllabus from EOQ-SG

Historial de Revisiones – Traducción al idioma español		
Versión	Fecha	Observaciones
0.90	22/11/2024	La traducción del programa de estudio se entrega para publicación.

BORRADOR

Tabla de Contenidos

Nota sobre Derechos de Propiedad Intelectual.....	2
Historial de Revisiones	3
Tabla de Contenidos	5
Agradecimientos	7
Notas de la Versión en Idioma Español	8
0 Introducción.....	9
0.1 Objetivo de este Programa de Estudio	9
0.2 El Probador Certificado Nivel Avanzado Gestión de Prueba	9
0.3 Trayectoria Profesional de los Probadores	9
0.4 Resultados de Negocio.....	10
0.5 Objetivos de Aprendizaje Evaluables y Nivel Cognitivo de Conocimiento	10
0.6 El Examen del Certificado de Nivel Avanzado de Gestión de la Prueba	11
0.7 Acreditación	11
0.8 Tratamiento de los Estándares	11
0.9 Nivel de Detalle.....	11
0.10 Organización del Programa de Estudio	12
0.11 ¿Cuáles son los Supuestos Fundamentales de este Programa de Estudio?	13
1 Gestión de las Actividades de Prueba.....	15
1.1 El Proceso de Prueba	19
1.1.1 Actividades de Planificación de la Prueba	19
1.1.2 Actividades de Monitorización y Control de la Prueba.....	20
1.1.3 Actividades de Compleción de la Prueba	20
1.2 El Contexto de la Prueba	21
1.2.1 Implicados de la Prueba	22
1.2.2 Importancia del Conocimiento de los Implicados en la Gestión de la Prueba	23
1.2.3 Gestión de la Prueba en un Modelo Híbrido de Desarrollo de Software	24
1.2.4 Actividades de Gestión de la Prueba para varios Modelos de Ciclo de Vida del Desarrollo de Software	24
1.2.5 Actividades de Gestión de la Prueba en Distintos Niveles de Prueba	26
1.2.6 Actividades de gestión de la prueba para distintos tipos de prueba	27
1.2.7 Actividades de Gestión de la Prueba para la Planificación, Monitorización y Control	28
1.3 Prueba Basada en el Riesgo	29
1.3.1 La Prueba como Actividad de Mitigación del Riesgo.....	29
1.3.2 Identificación de Riesgos de Calidad.....	30
1.3.3 Evaluación del Riesgo de Calidad	31
1.3.4 Mitigación del riesgo de Calidad Mediante una Prueba Adecuada.....	32
1.3.5 Técnicas para la Prueba Basada en el Riesgo.....	33
1.3.6 Métricas de Éxito y Dificultades Asociadas a la Prueba Basada en el Riesgo	33
1.4 La Estrategia de Prueba del Proyecto	35
1.4.1 Elección de un Enfoque de Prueba	35
1.4.2 Análisis de la Estrategia de Prueba de la Organización, del Contexto del Proyecto y de Otros Aspectos.....	35
1.4.3 Definición de los Objetivos de Prueba.....	37
1.5 Mejora del Proceso de Prueba	39
1.5.1 El Proceso de Mejora de la Prueba (IDEAL)	39
1.5.2 Mejora del Proceso de Prueba Basada en Modelos.....	40
1.5.3 Enfoque de Mejora del Proceso de Prueba Basada en Modelo Analítico	41

1.5.4	Retrospectivas.....	42
1.6	Herramientas de Prueba.....	43
1.6.1	Buenas Prácticas para la Introducción de Herramientas.....	43
1.6.2	Aspectos Técnicos y de Negocio para las Decisiones sobre la Herramienta.....	44
1.6.3	Consideraciones sobre el Proceso de Selección y Evaluación del Retorno de la Inversión.....	45
1.6.4	Ciclo de Vida de las Herramientas.....	46
1.6.5	Métrica de Herramientas.....	46
2	Gestión de Producto.....	48
2.1	Métricas de Prueba.....	50
2.1.1	Métricas para las Actividades de Gestión de la Prueba.....	50
2.1.2	Monitorización, Control y Compleción.....	51
2.1.3	Suministro de información de prueba.....	52
2.2	Estimación de la Prueba.....	54
2.2.1	Estimación de las Actividades que Implicará la Prueba.....	54
2.2.2	Factores que Pueden Influir en el Esfuerzo de Prueba.....	55
2.2.3	Selección de Técnicas de Estimación de Prueba.....	55
2.3	Gestión de Defectos.....	58
2.3.1	Ciclo de Vida de los Defectos.....	58
2.3.2	Gestión de defectos transfuncional.....	60
2.3.3	Particularidades de la Gestión de Defectos en Equipos Ágiles.....	61
2.3.4	Retos de la Gestión de Defectos en el Desarrollo Híbrido de Software.....	61
2.3.5	Información del Informe de Defecto.....	62
2.3.6	Definición de las Acciones de Mejora del Proceso Utilizando la Información de los Informes de Defectos.....	63
3	Gestión de Equipo.....	65
3.1	El Equipo de Prueba.....	67
3.1.1	Habilidades Habituales dentro de las Cuatro Áreas de Competencia.....	67
3.1.2	Analizar las Habilidades Requeridas de los Miembros del Equipo de Prueba.....	68
3.1.3	Evaluación de las Habilidades de los Miembros del Equipo de Prueba.....	69
3.1.4	Desarrollo de Habilidades de los Miembros del Equipo de Prueba.....	69
3.1.5	Habilidades de Gestión Necesarias para Dirigir un Equipo de Prueba.....	70
3.1.6	Factores que Motivan o Desmotivan a un Equipo de Prueba en Determinadas Situaciones.....	71
3.2	Relación con los Implicados.....	73
3.2.1	Coste de la Calidad.....	73
3.2.2	Relación Coste-Beneficio de la Prueba.....	73
4	Referencias.....	76
4.1	Estándares.....	76
4.2	Documentos de ISTQB®.....	76
4.3	Libros.....	76
4.4	Artículos.....	77
4.5	Páginas Web.....	77
5	Anexo A - Objetivos de Aprendizaje/Nivel Cognitivo de Conocimiento.....	78
6	Anexo B - Matriz de Trazabilidad de los Resultados de Negocio con respecto a los Objetivos de Aprendizaje.....	79
7	Anexo C - Traducciones Específicas, Definiciones, Acrónimos, Abreviaturas y Notas.....	87
10.1	Notas.....	88

Agradecimientos

Este documento fue entregado formalmente por la Asamblea General del ISTQB® el 3 de mayo de 2024.

Ha sido elaborado por un equipo de "International Software Testing Qualifications Board": Horst Pohlmann (propietario de producto, vicepresidente del AELWG), Tauhida Parveen, Francis Fenner, Laura Albert, Matthias Hamburg, Maud Schlich, Tanja Tremmel, Ralf Bongard, Erik van Veenendaal, Jan Giessen, Bernd Freimut, Andreas Neumeister, Georg Sehl, Rabi Arabi, Therese Kuhfuß, Ecaterina Irina Manole, Veronica Belcher, Kenji Onishi, Pushparajan Balasubramanian, Meile Postuma y Miroslav Renda.

El equipo da las gracias a Gary Mogyorodi por su revisión técnica (en Beta), a Julia Sabatine por su corrección de pruebas, al equipo revisor y a los "Member Boards" por sus sugerencias y aportaciones.

Las siguientes personas participaron en la revisión, los comentarios y la votación de este programa de estudio:

Revisiones Alfa: Benjamin Timmermans, Mattijs Kemmink, Rik Marselis, Jean-Francois Riverin, Gary Mogyorodi, Ralf Bongard, Ingvar Nordström, Yaron Tsubery, Imre Mészáros, Mattijs Kemmink, Ádám Bíró, Ramit M Kaul, Chinthaka Indikadahena, Darvay Tamás Béla, Beata Karpinska, Young jae Choi, Stuart Reid, Tal Pe'er, Meile Postuma, Daniel van der Zwan, Klaudia Dussa-Zieger, Jörn Münzel, Ralf Bongard, Petr Neugebauer, Derk-Jan de Groot, Rik Kochuyt, Andreas Hetz, Laura Albert, Eszter Sebestyeni, Tamás Szőke, Henriett Braunne Bokor, Ágota Horváth, Péter Sótér, Ferenc Hamori, Darvay Tamás Béla, Paul Weymouth, Lloyd Roden, Kevin Chen, Huang qin, Pushparajan Balasubramanian, Szilard Szell, Tamas Stöckert, Lucjan Stapp, Adam Roman, Anna Miazek, Márton Siska, Erhardt Wunderlich, László Kvintovics, Murian Song, Mette Bruhn-Pedersen, Petra Schneider, Michael Stahl, Ramit M Kaul, Imre Mészáros, Dilhan Jayakody, Francisca Cano Ortiz, Johan Klintin, Liang Ren, Ole Chr. Hansen, Zsolt Hargitai, Tamás Rakamazi, Kenji Onishi, Arnika Hryszko, Rabih Arabi, Veronica Belcher y Vignesh Balasubramanian.

Revisiones Beta: Maria-Therese Teichmann, Dominik Weber, Thomas Puffler, Peter Kunit, Martin Klonk, Michaël Pilaeten, Wim Decoutere, Arda Ender Torçuk, Piet de Roo, Rik Marselis, Jakub Platek, Ding Guofu, Zheng Dandan, Liang Ren, Yifan Chen, Hallur Helmsdal, Ole Chr. Hansen, Klaus Skafte, Gitte Ottosen, Tanzeela Gulzar, Arne Becher, Klaudia Dussa-Zieger, Jan Giesen, Florian Fieber, Carsten Weise, Arnd Pehl, Matthias Hamburg, Stephanie Ulrich, Jürgen Beniermann, Márton Siska, Sterbinszky Ádám, Ágnes Srancsik, Marton Matyas, Tamas Stöckert, Csilla Varga, Zsolt Hargitai, Bíró Ádám, Horváth Ágota, Sebestyeni Eszter, Szilárd Széll, Péter Sótér, Giancarlo Tomasig, Nicola de Rosa, Kaiwalya Katyarmal, Pradeep Tiwari, Sreeja Padmakumari, Seunghee Choi, Stuart Reid, Dmitrij Nikolajev, Mantas Aniulis, Monika Stoecklein-Olsen, Adam Roman, Mahmoud Khalaili, Ingvar Nordström, Beata Karpinska, Armin Born, Ferdinand Gramsamer, Mergole Kuate, Thomas Letzkus, Nishan Portoyan, Ainsley Rood, Lloyd Roden, Sarah Ireton.

Notas de la Versión en Idioma Español

Este “Programa de Estudio de Probador Certificado del ISTQB®” de Nivel Avanzado, Gestión de Prueba, Versión 3.0” ha sido traducido por Spanish Software Testing Qualifications Board (SSTQB).

El equipo de traducción y revisión para este programa de estudio es el siguiente (por orden alfabético):

Responsable de la traducción: Gustavo Márquez Sosa (España)

Responsable de la revisión: Luisa Morales Gómez Tejedor (España)

El Comité Ejecutivo del SSTQB agradece toda aportación que permita mejorar esta traducción del programa de estudio.

En una siguiente versión se podrán incorporar aportaciones adicionales. El SSTQB considera conveniente mantener abierta la posibilidad de realizar cambios en el “Programa de Estudio”.

Madrid, 22 de noviembre de 2024

0 Introducción

0.1 Objetivo de este Programa de Estudio

Este programa de estudio constituye la base para la formación como Probador Certificado del ISTQB® de Nivel Avanzado, Gestión de Prueba. El ISTQB proporciona este programa de estudio en los siguientes términos:

1. A los comités miembro, para traducir a su idioma local y para acreditar a los proveedores de formación. Los Comités Miembro pueden adaptar el programa de estudio a sus necesidades lingüísticas particulares y añadir referencias para adaptarlo a sus publicaciones locales.
2. A los organismos de certificación, para elaborar las preguntas del examen en su lengua local adaptadas a los objetivos de aprendizaje de este programa de estudio.
3. A los proveedores de formación, para desarrollar material didáctico y determinar los métodos de enseñanza adecuados.
4. A los candidatos a la certificación, para que preparen el examen de certificación (ya sea como parte de un curso de formación o de forma independiente).
5. A la comunidad internacional de ingeniería de software y sistemas, para avanzar en la profesión de prueba del software y sistemas, y como fuente de libros y artículos.

0.2 El Probador Certificado Nivel Avanzado Gestión de Prueba

Esta cualificación de nivel avanzado está dirigida a cualquier persona implicada en la gestión de la prueba del software. Esto incluye a personas en roles como probadores, consultores de pruebas, jefes de prueba, probadores de aceptación de usuario, scrum masters, jefes de proyecto o propietarios de producto. Esta cualificación de nivel avanzado en gestión de la prueba también es adecuada para cualquier persona que desee una comprensión avanzada de la prueba de software, como directores de proyecto, directores de calidad, directores de desarrollo de software, analistas de negocio, directores de TI y consultores de gestión. Los poseedores del "Certificado de gestión de la prueba de nivel avanzado "podrán acceder a la cualificación en pruebas de software "ISTQB® Expert Level". El certificado "ISTQB® Certified Tester Advanced Level - Test Manager or Test Management" es válido de por vida y no requiere renovación. El certificado está reconocido internacionalmente y demuestra la competencia profesional de los candidatos y su credibilidad en la gestión de la prueba.

0.3 Trayectoria Profesional de los Probadores

El esquema ISTQB® proporciona apoyo a los profesionales de pruebas en todas las etapas de sus carreras. Las personas que obtienen la "ISTQB® Certified Tester Advanced Level Test Management Certification" también pueden estar interesadas en las otras "Core Advanced Level certifications (i.e., Test Analyst y Technical Test Analyst) y posteriormente en las certificaciones ISTQB® Expert Level (i.e., Test Management o Improving the Test Process)". Cualquiera que busque desarrollar habilidades en pruebas dentro del desarrollo ágil de software puede considerar las certificaciones "Agile Technical Tester" o "Agile Test Leadership at Scale". La corriente de Especialista ofrece una inmersión profunda en áreas que tienen enfoques de prueba y actividades de prueba específicos (por ejemplo, en "Test Automation", "AI Testing", o "Mobile App Testing"), o conocimientos de pruebas de agrupamiento para ciertos dominios de la industria (por ejemplo, "Automotive" o "Gaming"). Visite www.istqb.org para obtener la información más reciente del "Esquema de probador certificado de ISTQB®".

0.4 Resultados de Negocio

Esta sección enumera los once resultados de negocio que se esperan de un candidato que haya obtenido la certificación de gestión de la prueba de nivel avanzado.

Un probador certificado en gestión de la prueba de nivel avanzado puede...

- TM_01** Gestionar la prueba en varios proyectos de desarrollo de software mediante la aplicación de procesos de gestión de la prueba establecidos para el equipo de proyecto o la organización de pruebas.
- TM_02** Identificar a los implicados en las pruebas y los modelos de ciclo de vida de desarrollo del software que sean relevantes en un contexto determinado.
- TM_03** Organizar sesiones de identificación del riesgo y de evaluación del riesgo dentro de cualquier ciclo de vida de desarrollo de software y utilizar los resultados para guiar las pruebas con el fin de alcanzar los objetivos de prueba.
- TM_04** Definir una estrategia de prueba del proyecto consistente con la estrategia de prueba organizativa y el contexto del proyecto.
- TM_05** Monitorizar y controlar continuamente las pruebas para lograr los objetivos del proyecto.
- TM_06** Evaluar e informar del avance de las pruebas a los implicados en el proyecto.
- TM_07** Identificar las habilidades necesarias y desarrollarlas en su equipo.
- TM_08** Preparar y presentar un caso de negocio para las pruebas en distintos contextos que describa los costes y los beneficios esperados.
- TM_09** Dirigir las actividades de mejora del proceso de prueba en proyectos o flujos de productos de desarrollo de software y contribuir a las iniciativas organizativas de mejora del proceso de prueba.
- TM_10** Planificar las actividades de prueba, incluida la infraestructura de la prueba necesaria y estimar el esfuerzo requerido para la prueba.
- TM_11** Crear informes de defectos y un flujo de trabajo de defectos adecuado para un ciclo de vida de desarrollo del software.

0.5 Objetivos de Aprendizaje Evaluables y Nivel Cognitivo de Conocimiento

Los objetivos de aprendizaje respaldan los resultados de negocio y se utilizan para crear los exámenes de Certified Tester Advanced Test Management.

En general, todos los contenidos de este programa de estudio son evaluables a nivel K1, excepto la Introducción, las Referencias, el Epílogo y los Anexos. Es decir, se puede pedir al candidato que reconozca, recuerde o rememore una palabra clave o un concepto mencionado en cualquiera de los tres capítulos de este programa de estudio. Los objetivos de aprendizaje específicos y los niveles correspondientes se muestran al principio de cada capítulo y se clasifican de la siguiente manera:

- K2: Comprender
- K3: Aplicar

- K4: Analizar

En el Anexo A se ofrecen más detalles y ejemplos de los objetivos de aprendizaje.

Deberán recordarse todos los términos que figuran como palabras clave justo debajo de los títulos de los capítulos (K1), aunque no se mencionen explícitamente en los objetivos de aprendizaje.

0.6 El Examen del Certificado de Nivel Avanzado de Gestión de la Prueba

El examen del Certificado de nivel avanzado de Gestión de la Prueba se basa en este programa de estudio. Las respuestas a las preguntas del examen pueden requerir el uso de material basado en más de una sección de este programa de estudio. Todas las secciones del programa de estudio son evaluables, excepto la Introducción, los Anexos y las Referencias. Las normas y los libros se incluyen como referencias (capítulo 5), pero su contenido, más allá de lo que se resume en el propio programa de estudio a partir de dichas normas y libros, no es evaluable.

Para más información, consulte el documento “Estructuras y normas de examen 1.1 Compatibilidad con los programas de estudio de los niveles básico y avanzado y los módulos de especialización”.

- El criterio de acceso para presentarse al examen de certificación de gestión de la prueba de nivel avanzado es:-
 - Los candidatos deben poseer el Certificado de Nivel Básico ISTQB® antes de presentarse al examen de certificación de gestión de la prueba de Nivel Avanzado. Sin embargo, se recomienda encarecidamente que el candidato tenga al menos una formación mínima en desarrollo de software o en prueba de software, como por ejemplo seis meses de experiencia como probador o como desarrollador de software.

0.7 Acreditación

Un Comité Miembro de ISTQB® puede acreditar a los proveedores de formación cuyo material de curso siga este programa de estudio. Los proveedores de formación deberán obtener las directrices de acreditación del Comité Miembro u organismo que realice la acreditación. Un curso acreditado es reconocido como conforme a este programa de estudio y se le permite tener un examen ISTQB® como parte del curso.

Las directrices de acreditación para este programa de estudio son las Directrices de Acreditación genéricas publicadas por el Grupo de Trabajo de Gestión de Procesos y Conformidad del ISTQB®.

0.8 Tratamiento de los Estándares

Hay estándares a los que se hace referencia en el Programa de Estudio de Gestión de la Prueba de Nivel Avanzado (por ejemplo, ISO, IEC e IEEE). El propósito de estas referencias es proporcionar un marco de trabajo o proporcionar una fuente de información adicional si así lo desea el lector. Tenga en cuenta que el programa de estudio utiliza estos estándares como referencias. No está previsto que los estándares sean objeto de examen. Consulte el capítulo 5 Referencias para obtener más información sobre los estándares.

0.9 Nivel de Detalle

El nivel de detalle de este programa de estudio permite que los cursos y los exámenes sean consistentes a escala internacional. Para lograr este objetivo, el programa de estudio consta de:

- Objetivos generales de instrucción que describen la intención del programa de estudio de gestión de la prueba de nivel avanzado.
- Una lista de palabras clave que los alumnos deben ser capaces de recordar.
- Objetivos de aprendizaje para cada área de conocimiento, que describen el resultado cognitivo de aprendizaje que debe alcanzarse.
- Una descripción de los conceptos clave, incluyendo referencias a fuentes como la bibliografía aceptada o los estándares.

El contenido del programa de estudio no es una descripción de toda el área de conocimiento de las pruebas; refleja el nivel de detalle que se cubrirá en los cursos de formación en gestión de la prueba de nivel avanzado. Se concentra en conceptos y técnicas de prueba que pueden aplicarse a todos los proyectos de software.

0.10 Organización del Programa de Estudio

Hay tres capítulos con contenido evaluable. El encabezamiento superior de cada capítulo especifica el tiempo mínimo requerido para que los cursos de formación acreditados cubran el contenido del capítulo; no se proporciona cronología por debajo del nivel del capítulo. Para los cursos de formación acreditados, el programa de estudio exige un mínimo de 22,75 horas de instrucción, distribuidas en los tres capítulos del siguiente modo:

- Capítulo 1: Gestión de las Actividades de Prueba (750 minutos)
 - El participante aprende a explicar las actividades de gestión de la prueba (es decir, planificación de la prueba, monitorización de la prueba, control de la prueba y compleción).
 - El participante aprende a definir una estrategia de prueba del proyecto que incluya los objetivos de prueba y a seleccionar el enfoque de prueba adecuado y consistente con la estrategia de prueba organizativa y el contexto del proyecto.
 - El participante aprende a gestionar proyectos en diversos contextos.
 - El participante aprende a aplicar pruebas basadas en el riesgo para concentrar las pruebas en los riesgos identificados.
 - El participante aprende a dirigir las actividades de mejora del proceso de prueba mediante la realización de una retrospectiva del proyecto o de la iteración.
 - El participante aprende a mejorar el soporte de herramientas para las pruebas teniendo en cuenta los riesgos, costes y beneficios del soporte de herramientas.
- Capítulo 2: Gestión del producto (390 minutos)
 - El participante aprende a monitorizar y controlar las pruebas para cumplir los objetivos de la prueba utilizando métricas de prueba y a evaluar e informar del avance de la prueba.
 - El participante aprende a seleccionar técnicas adecuadas de estimación de la prueba en diferentes filiales siguiendo diferentes modelos de desarrollo de software.
 - El participante aprende a definir un flujo de trabajo de defectos dentro de la gestión de defectos para ajustarse al modelo de desarrollo secuencial, ágil e híbrido de software.
- Capítulo 3: Gestión del equipo (225 minutos)
 - El participante aprende a analizar el contexto de un proyecto determinado para identificar las habilidades que necesita el equipo de prueba.
 - El participante aprende a gestionar un equipo según el enfoque de equipo completo.
 - El participante aprende a definir un caso de negocio para las actividades de prueba en el proyecto.

Nota: Para cada objetivo de aprendizaje existe en el programa de estudio una subsección correspondiente con contenido (por ejemplo, para LO-1.2.3 existe una subsección 1.2.3).

0.11 ¿Cuáles son los Supuestos Fundamentales de este Programa de Estudio?

Este programa de estudio está dirigido a cualquier persona que desee alcanzar un nivel avanzado de competencia en la gestión de la prueba, como directores de prueba, analistas de prueba, ingenieros de prueba, consultores de prueba, coordinadores de prueba, jefes de prueba y directores de proyecto. El programa de estudio está alineado con el Nivel Básico de ISTQB® (“ISTQB® Foundation Level Syllabus”) V.4, que proporciona los conocimientos básicos y la comprensión de la prueba de software.

Este programa de estudio cubre dos roles principales en la prueba: el rol de gestión de la prueba y el rol de prueba. El rol de gestión de la prueba también se conoce como jefe de prueba en el contexto del modelo de desarrollo secuencial, en el que el jefe de prueba suele ser un rol independiente del jefe de proyecto o del propietario de producto. El rol de gestión de la prueba es responsable del proceso general de prueba, del equipo de prueba y de la gestión de la prueba. Esto incluye la definición de la estrategia de la prueba, la planificación de las actividades de la prueba, la monitorización y el control del avance de la prueba, la elaboración de informes de los resultados de la prueba y la gestión de los riesgos y dificultades de la prueba. El rol de gestión de la prueba también asegura que los objetivos de la prueba estén alineados con las necesidades del negocio y de los implicados, y que las actividades de la prueba estén coordinadas con otros implicados del proyecto.

El rol de prueba también realiza actividades de evaluación de la prueba, gestión de defectos y cierre de la prueba. El rol de prueba utiliza varias técnicas de prueba para asegurar la calidad y fiabilidad de los artefactos de prueba y del sistema sujeto a prueba. El rol de prueba también utiliza herramientas de prueba y automatización para apoyar el proceso de prueba y mejorar la eficiencia y la efectividad de la prueba. Las actividades y tareas asignadas a estos roles pueden variar en función del contexto, como el proyecto, el producto, las habilidades y la organización. (véase el programa de estudio de nivel básico de ISTQB® V.4).

El término miembro de un equipo de prueba se utiliza en este programa de estudio para referirse a cualquier persona que desempeñe un rol de gestión de la prueba o de prueba que realice pruebas, independientemente del contexto organizativo y de otros roles. Los equipos de prueba están formados por personas con distintas habilidades y competencias. Los miembros del equipo también pueden tener distintos niveles de experiencia y certificación, como nivel básico, avanzado o experto. Los miembros del equipo de prueba también pueden tener diferentes roles y responsabilidades en función del enfoque de prueba y del modelo de proceso de prueba utilizado, como prueba Ágil, prueba basada en modelos, prueba basada en el riesgo, etc.

Un punto importante sobre la perspectiva que ofrece el programa de estudio es el hecho de que se concentra en la gestión de la prueba a nivel de proyecto y no a nivel de organización. Por lo tanto, este programa de estudio se ajusta y contiene información que puede utilizarse para las actividades de gestión de la prueba a nivel de proyecto, pero no tanto para la gestión de la prueba a nivel organizativo.

El desarrollo híbrido de software se utiliza en este programa de estudio para referirse a un enfoque de desarrollo de software que combina elementos de diferentes modelos de ciclo de vida del software, como el modelo V, el modelo iterativo, el modelo incremental y el modelo ágil. El desarrollo híbrido de software pretende aprovechar los puntos fuertes y mitigar los puntos débiles de cada modelo, en función del contexto y las necesidades del proyecto. Por ejemplo, un enfoque de desarrollo híbrido de software puede utilizar un modelo V para las fases iniciales de planificación y análisis de requisitos, seguido por un modelo ágil para las fases de diseño, desarrollo y prueba. Alternativamente, un enfoque de desarrollo híbrido de software puede utilizar un modelo iterativo para la gestión general del proyecto, aplicando al mismo tiempo un modelo incremental para cada iteración y un modelo ágil para cada incremento. El desarrollo híbrido de software requiere un elevado grado de flexibilidad, comunicación y colaboración entre las partes implicadas, así como una clara comprensión de los objetivos, riesgos y restricciones de cada fase y modelo.

Según este programa de estudio y el Glosario, una estrategia de prueba es una descripción de cómo se llevará a cabo la prueba para alcanzar los objetivos de la prueba en unas circunstancias determinadas.

Una estrategia de prueba define el alcance general, el enfoque y los recursos para probar un sistema o un producto. Suele documentarse en un plan de prueba o como parte de otros documentos, en función del contexto de la prueba. Una estrategia de prueba está influida por la estrategia de prueba organizativa, que es una estrategia de prueba de alto nivel que describe cómo se realizan las pruebas en una organización. Una estrategia de prueba también puede existir para un único nivel de prueba o un tipo de prueba, que son aspectos específicos de la prueba que se concentran en diferentes objetivos, metas y criterios. El término genérico “estrategia de prueba” puede utilizarse en cualquier contexto (proyecto, organización, producto). Un enfoque de prueba es la forma en que se implementan las tareas de prueba, especialmente la selección y combinación de niveles de prueba, tipos de prueba y técnicas de prueba para pruebas estáticas y dinámicas, así como otras prácticas de prueba como las pruebas mediante guion, las pruebas manuales, las pruebas back-to-back, etc. El enfoque de prueba elegido por el rol de gestión de la prueba es una decisión clave a la hora de formular una estrategia de prueba apropiada para un contexto determinado.

1 Gestión de las Actividades de Prueba

Duración: 750 minutos

Palabras Clave¹

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	prueba basada en la experiencia	experience-based testing
ISTQB	prueba funcional	functional testing
ESPDOM	métrica de pregunta objetivo (MPO)	goal question metric (GQM)
ISTQB	modelo híbrido de desarrollo de software	hybrid software development model
ESPDOM	IDEAL	IDEAL
ISTQB	mejora	improvement
ISTQB	modelo de desarrollo incremental	incremental development model
ESPDOM	indicador	indicator
ISTQB	modelo de desarrollo iterativo	iterative development model
ESPDOM	medida	measure
ESPDOM	métrica	metric
ISTQB	prueba no funcional	non-functional testing
ISTQB	riesgo de producto	product risk
ISTQB	riesgo de calidad	quality risk
ISTQB	retrospectiva	retrospective
ISTQB	análisis del riesgo	risk analysis
ISTQB	evaluación del riesgo	risk assessment
ISTQB	identificación del riesgo	risk identification
ISTQB	impacto del riesgo	risk impact
ISTQB	nivel de riesgo	risk level
ISTQB	probabilidad del riesgo	risk likelihood
ISTQB	gestión del riesgo	risk management
ISTQB	mitigación del riesgo	risk mitigation

¹ Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Tipo Palabra Clave	Español	Inglés
ISTQB	monitorización del riesgo	risk monitoring
ISTQB	prueba basada en el riesgo	risk-based testing
ISTQB	metodología por objetivos S.M.A.R.T.	S.M.A.R.T. goal methodology
ISTQB	modelo de desarrollo secuencial	sequential development model
ISTQB	ciclo de vida de desarrollo del software	software development lifecycle
ISTQB	compleción de la prueba	test completion
ISTQB	control de prueba	test control
ISTQB	nivel de prueba	test level
ISTQB	Test Maturity Model integration	Test Maturity Model integration
ISTQB	monitorización de la prueba	test monitoring
ISTQB	objetivo de prueba	test objective
ISTQB	plan de prueba	test plan
ISTQB	planificación de la prueba	test planning
ISTQB	proceso de prueba	test process
ISTQB	estrategia de prueba	test strategy
ISTQB	tipo de prueba	test type
ISTQB	TPI NEXT	TPI NEXT

Objetivos de Aprendizaje para “Capítulo 1”

1.1 El proceso de Prueba

- TM-1.1.1 (K2)** Resumir la planificación de la prueba.
- TM-1.1.2 (K2)** Resumir la monitorización de la prueba.
- TM-1.1.3 (K2)** Resumir la compleción de la prueba.

1.2 El Contexto de la Prueba

- TM-1.2.1 (K2)** Comparar por qué los distintos implicados están interesados en la prueba.
- TM-1.2.2 (K2)** Explicar por qué es importante el conocimiento de los implicados en la gestión de la prueba.
- TM-1.2.3 (K2)** Explicar la prueba en un modelo híbrido de desarrollo de software.
- TM-1.2.4 (K2)** Resumir las actividades de gestión de la prueba para varios ciclos de vida de desarrollo del software.
- TM-1.2.5 (K2)** Comparar las actividades de gestión de la prueba en varios niveles de prueba.
- TM-1.2.6 (K2)** Comparar las actividades de gestión de la prueba para varios tipos de prueba.
- TM-1.2.7 (K4)** **TM-1.2.7 (K4)** Analizar un proyecto dado y determinar las actividades de gestión de la prueba que enfatizan la planificación de la prueba, la monitorización de la prueba y el control de la prueba.

1.3 Prueba Basada en el Riesgo

- TM-1.3.1 (K2)** Explicar las distintas medidas que se toman en la prueba basada en el riesgo para responder a riesgos.
- TM-1.3.2 (K2)** Aportar ejemplos de las distintas técnicas que puede utilizar un jefe de prueba para identificar riesgos relacionados con la calidad del producto.
- TM-1.3.3 (K2)** Resumir los factores que determinan los niveles de riesgo relacionados con la calidad del producto.
- TM-1.3.4 (K4)** Seleccionar las actividades de prueba adecuadas para mitigar los riesgos en función de su nivel de riesgo en un contexto determinado.
- TM-1.3.5 (K2)** Diferenciar entre ejemplos pesados y ligeros de técnicas de prueba basadas en el riesgo.
- TM-1.3.6 (K2)** Aportar ejemplos de métricas de éxito y dificultades asociadas a la prueba basada en el riesgo.

1.4 La Estrategia de Prueba del Proyecto

- TM-1.4.1 (K2)** Explicar las opciones típicas de un enfoque de prueba.
- TM-1.4.2 (K4)** Analizar una estrategia de prueba organizativa y el contexto del proyecto para seleccionar el enfoque de prueba adecuado.
- TM-1.4.3 (K3)** Utilizar la metodología de objetivos S.M.A.R.T. para definir objetivos de prueba y criterios de salida medibles.

1.5. Mejora del Proceso de Prueba

- TM-1.5.1 (K2)** Explicar cómo utilizar el modelo de IDEAL para la mejora del proceso de prueba en un proyecto determinado.
- TM-1.5.2 (K2)** Resumir el enfoque de mejora de proceso basado en modelos y comprender cómo aplicarlo en el contexto de un proyecto.
- TM-1.5.3 (K2)** Resumir el enfoque de mejora del proceso basado en el análisis y comprender cómo aplicarlo en el contexto de un proyecto.
- TM-1.5.4 (K3)** Aplicar una retrospectiva de proyecto o de iteración para evaluar los procesos de prueba y descubrir áreas de prueba que puedan ser objeto de mejora.

1.6 Herramientas de prueba

- TM-1.6.1 (K2)** Resumir las buenas prácticas para la introducción de herramientas.
- TM-1.6.2 (K2)** Explicar el impacto de los diferentes aspectos técnicos y de negocio en la decisión sobre un tipo de herramienta.
- TM-1.6.3 (K4)** Analizar una situación dada para crear un plan para la selección de una herramienta que cubra riesgos, costes y beneficios.
- TM-1.6.4 (K2)** Diferenciar entre las etapas del ciclo de vida de una herramienta.
- TM-1.6.5 (K2)** Aportar ejemplos de recopilación y evaluación de métricas mediante el uso de herramientas.

1.1 El Proceso de Prueba

Introducción

El programa de estudio de Probador Certificado de ISTQB® Nivel Básico V4.0 describe un proceso de prueba que incluye las siguientes actividades: planificación de la prueba, monitorización y control de la prueba, análisis de la prueba, diseño de la prueba, implementación de la prueba, ejecución de la prueba y completación de la prueba.

El programa de estudio de Probador Certificado de ISTQB® Nivel Básico V4.0 establece que estas actividades del proceso de prueba a menudo se implementan de forma iterativa o en paralelo, dependiendo del modelo del ciclo de vida de desarrollo del software (CVDS) y del contexto del proyecto. Normalmente es necesario adaptar estas actividades dentro del contexto del producto y del proyecto.

En este programa de estudio, la atención se centra en las siguientes actividades clave de gestión de la prueba:

- **Planificación de la prueba:** definir los objetivos de la prueba, el enfoque de la prueba, el alcance de la prueba, los recursos de la prueba, el calendario de la prueba, los entregables de la prueba y los participantes en la prueba (implicados de la prueba).
- **Monitorización y control de la prueba:** seguimiento del avance, los resultados y las desviaciones de la prueba; adopción de medidas correctivas cuando sea necesario; suministro de información sobre el estado y los resultados de la prueba a los implicados relevantes.
- **Completación de la prueba:** finalización y archivo de los artefactos de la prueba, evaluación del proceso de prueba y del producto de la prueba, identificación de las acciones de mejora del proceso de prueba y comunicación del cierre de la prueba a las partes interesadas pertinentes.

La norma ISO/IEC/IEEE 29119-2 define los procesos de gestión de la prueba que cubren estas actividades de gestión de la prueba. Estos procesos de gestión de la prueba pueden aplicarse a diferentes niveles de prueba, como proyecto, programa o cartera. Cada nivel de prueba puede tener su propio plan de prueba que se alinee con el plan de prueba de nivel superior.

1.1.1 Actividades de Planificación de la Prueba

Esta sección se centra en las actividades de planificación de la prueba para distintos alcances, como todo el proyecto, un nivel de prueba, un tipo de prueba o una entrega / iteración / esprint en Ágil. Dependiendo del alcance, la planificación de la prueba puede comenzar y terminar en diferentes puntos del proceso de desarrollo. La planificación de la prueba es una actividad que consiste en identificar las actividades y los recursos necesarios para alcanzar los objetivos de prueba identificados en la política de prueba. La planificación de la prueba debe iniciarse lo antes posible en el proceso de desarrollo, preferiblemente antes de que se identifiquen los requisitos, y debe actualizarse a medida que avanza el proyecto. La planificación de la prueba suele ser un proceso iterativo que requiere una nueva planificación durante el proyecto para adaptarse a los cambios y a la retroalimentación.

Las siguientes tareas forman parte de la planificación de la prueba (similares a las que se encuentran en la norma ISO/IEC/IEEE 29119-2):

- **Comprender el contexto y organizar la planificación de la prueba.**
Comprender el contexto de la organización (por ejemplo, la política de prueba y la estrategia de prueba de la organización), el alcance de la prueba y el elemento de prueba (es decir, el producto de trabajo que se está probando) es fundamental para la planificación de la prueba (véase el apartado 1.2., El contexto de la prueba). También implica todas las actividades necesarias para organizar el desarrollo del plan de prueba y obtener la aprobación de esas actividades y del calendario por parte de los implicados (por ejemplo, el propietario del producto, el jefe de proyecto o el director del equipo de desarrollo).
- **Identificar y analizar los riesgos de producto.**

El análisis del riesgo implica identificar y evaluar el impacto potencial y la probabilidad de los riesgos de producto como parte de la planificación de la prueba. Para obtener más información sobre riesgos de producto, consulte la sección 1.3 de este programa de estudio, Prueba Basada en el Riesgo.

- **Identificar los enfoques de tratamiento de riesgos.**

Se seleccionan los enfoques de tratamiento del riesgo adecuados en función del análisis del riesgo y se documentan en el plan de prueba. Estos pueden incluir acciones preventivas, correctivas o paliativas para abordar los riesgos identificados

- **Definir el enfoque de prueba, estimar y asignar los recursos de prueba.**

Se define el enfoque de prueba para el alcance actual de la prueba en función de la estrategia de prueba de la organización, las normas reglamentarias, las limitaciones impuestas por el proyecto y los enfoques de tratamiento de riesgos (véase el apartado 1.4, Estrategia de prueba del proyecto). Cuando se define un enfoque de prueba, es importante estimar los recursos de prueba necesarios, como el personal de prueba, las herramientas de prueba, los entornos de prueba y los datos de prueba, y asignar esos recursos a las actividades de prueba.

- **Establecer el plan de prueba.**

El plan de prueba debe ser aceptado por todos los implicados y, por lo tanto, se deberían resolver los desacuerdos entre ellos.

1.1.2 Actividades de Monitorización y Control de la Prueba

Para que la gestión de la prueba proporcione un control eficiente de la misma, debe establecerse un calendario de prueba y un marco de monitorización que permita el seguimiento del estado y el avance de la prueba. Este marco debe incluir las medidas detalladas y los objetivos necesarios para relacionar el estado de los productos de trabajo y los recursos de la prueba con el plan y los objetivos estratégicos.

Para los proyectos pequeños y menos complejos, puede ser relativamente fácil relacionar los productos del trabajo de prueba y las actividades con el plan y los objetivos estratégicos, pero por lo general deben definirse objetivos más detallados para lograrlo. Esto puede incluir la definición de medidas y metas necesarias para cumplir los objetivos de prueba y la cobertura de la base de prueba.

La necesidad de relacionar el estado de los productos del trabajo de prueba y las actividades de una manera que sea comprensible y relevante para el proyecto y los implicados del negocio es un aspecto de particular importancia.

La monitorización y el control de la prueba son actividades continuas.

El control de prueba compara el avance real con el plan de prueba e implementa las acciones correctivas necesarias. Guía la prueba para cumplir las estrategias y los objetivos de la prueba (véase la sección 1.4, La estrategia de prueba del proyecto), y vuelve a examinar las actividades de planificación de la prueba cuando es necesario. Los datos de control requieren información detallada sobre la planificación de la prueba para poder reaccionar adecuadamente. Esta actividad implica:

- Gestionar las desviaciones de la prueba planificada.
- Tratar los riesgos recientemente identificados y modificados.
- Establecer la preparación para comenzar la prueba.
- Conceder y obtener la aprobación para la compleción de la prueba basándose en los criterios de salida.

La monitorización de la prueba implica recopilar y registrar los resultados de la prueba, identificar las desviaciones de la prueba planificada, identificar y analizar los nuevos riesgos que requieren pruebas y monitorizar los cambios para los riesgos identificados.

1.1.3 Actividades de Compleción de la Prueba

La compleción de la prueba tiene lugar normalmente en hitos del proyecto (por ejemplo, una entrega, el final de una iteración o la compleción del nivel de prueba). Se crean solicitudes de cambio o elementos de

la lista de trabajo acumulado del producto para cualquier defecto no resuelto. Véase el programa de estudio de nivel básico de ISTQB® V.4. Una vez cumplidos los criterios de salida, se deberían capturar, archivar y proporcionar las salidas clave a los implicados relevantes. La compleción de la prueba requiere las siguientes tareas:

- **Crear y aprobar el informe de compleción de la prueba.**

Esta tarea asegura que se ha realizado toda la prueba y que se han cumplido todos los objetivos de prueba. Esta tarea implica la recopilación de información relevante de diversos productos de prueba, como planes de prueba, resultados de prueba, informes de avance de prueba, informes de compleción de prueba e informes de defectos. La información recopilada se evalúa y se resume en el informe de compleción de la prueba. El informe de compleción se aprueba y se comunica a los implicados relevantes.

- **Archivar los productos de prueba.**

Esta tarea identifica los productos de prueba que pueden ser útiles en el futuro o que se espera que sean reutilizados, que normalmente son casos de prueba. Los hace accesibles y fáciles de ser entendidos para su futura reutilización. Además, los resultados de la prueba, los registros de prueba, los informes de prueba y otro material de prueba deben archivar temporalmente en el sistema de gestión de la configuración.

- **Entrega del producto de prueba.**

Esta tarea entrega productos de trabajo valiosos a quienes los necesitan. Por ejemplo, los defectos conocidos aplazados o aceptados deben comunicarse a quienes vayan a utilizar o apoyar el uso del producto de prueba.

- **Realizar todas las tareas necesarias para limpiar el entorno de prueba y restaurarlo a un estado predefinido.**

Esta tarea asegura que el entorno de prueba esté listo para el siguiente ciclo de prueba o proyecto. Consiste en eliminar todos los datos de prueba, herramientas de prueba, controladores de prueba, stubs de prueba y guiones de prueba del entorno de prueba. También implica restaurar el entorno de prueba a su estado original o deseado.

- **Realizar/recopilar/documentar las lecciones aprendidas.**

Esta tarea se realiza en retrospectivas en las que se discuten y documentan las lecciones importantes aprendidas durante el proceso de prueba. Esto puede incluir hallazgos para todo el ciclo de vida de desarrollo de software (SDLC). Las lecciones aprendidas pueden utilizarse para la mejora del proceso de prueba, como se describe en la sección 1.5. de este programa de estudio, Mejora del proceso de prueba.

1.2 El Contexto de la Prueba

El contexto de la prueba engloba las condiciones y restricciones únicas que influyen en el proceso de prueba, dando forma a las decisiones y estrategias de planificación, diseño y ejecución de las pruebas. Es vital que los jefes de prueba comprendan este contexto para alinear la prueba con las necesidades y objetivos específicos del proyecto de desarrollo de software. Este contexto puede diferir en función del tipo de producto, la industria, los requisitos normativos y, sobre todo, el ciclo de vida de desarrollo de software (CVDS) que se esté empleando.

Los jefes de prueba se encargan de aplicar las estrategias de prueba establecidas y de elegir las técnicas de prueba en lugar de desarrollarlas. Desempeñan un rol clave en la formulación de planes de prueba adaptados al contexto del proyecto. Al comprender y tener en cuenta estos diversos factores, los jefes de prueba pueden asegurar que la prueba sea pertinente, eficaz y eficiente a la hora de cumplir los objetivos de la prueba.

1.2.1 Implicados de la Prueba

Los implicados de la prueba son personas o grupos con un interés directo o indirecto en la calidad del producto. A continuación, se presenta una lista de implicados potenciales típica, modificada para reflejar sus diversos intereses en el proceso de prueba:

- **Desarrolladores, líderes de desarrollo y directores de desarrollo:** Además de implementar el sistema sometido a prueba y actuar en función de los resultados de la prueba, estos implicados también participan en la prueba unitaria y contribuyen al proceso de prueba.
- **Probadores, líderes de prueba y jefes de prueba:** Estas personas preparan el software de prueba, lo que incluye el desarrollo de planes de prueba y contribuir al proceso de prueba a través de actividades como el análisis de requisitos, diseño de prueba, ejecución de prueba, seguimiento e informe de defectos, automatización de la prueba e informe del avance de la prueba.
- **Jefes de proyecto, propietarios de producto y usuarios de negocio:** Especifican los requisitos, definen el nivel de calidad exigido y recomiendan la cobertura necesaria en función de los riesgos percibidos. También revisan los productos del trabajo, participan en la prueba de aceptación de usuario (PAU) y toman decisiones basadas en los resultados de las pruebas.
- **Equipo de operaciones:** Participan en las pruebas de aceptación operativa, garantizan que el sistema está listo para la producción y contribuyen a definir los requisitos no funcionales.
- **Clientes y usuarios:** Los clientes compran el producto, mientras que los usuarios lo utilizan directamente. Ambos son clave en la definición de los requisitos y deben participar en la prueba de aceptación de usuario (PAU) para validar que el producto satisface sus necesidades.

Esta lista no incluye a todos los implicados potenciales. Los jefes de prueba deben realizar un análisis de los implicados como parte de la creación de la estrategia de prueba y del plan de prueba, teniendo en cuenta el alcance de la prueba en la identificación de los implicados específicos para su proyecto.

1.2.2 Importancia del Conocimiento de los Implicados en la Gestión de la Prueba

En la gestión de la prueba, es crucial tener en cuenta las perspectivas y la influencia de los distintos implicados. La matriz de implicados, a menudo denominada matriz de poder-interés, orienta a los jefes de prueba a la hora de priorizar el compromiso de los implicados y gestionar las expectativas de forma eficaz. La matriz de implicados es una herramienta estratégica en la gestión de la prueba que:

- Utiliza la experiencia de los implicados, con usuarios finales y equipos técnicos que aportan retroalimentación y puntos de vista sobre el rendimiento y la seguridad.
- Apoya la gestión del riesgo destacando los intereses y la influencia de los implicados, fomentando los esfuerzos proactivos de mitigación.
- Valora las diversas perspectivas con una retroalimentación de valor.

La matriz de implicados se compone de cuatro cuadrantes:

- Promotores (Alto nivel de influencia, Alto nivel de interés): colaboradores clave con alto nivel de influencia e interés, vitales para dar forma a la estrategia y el plan de prueba.
- Latentes (Alto nivel de influencia, Bajo nivel de interés): Aunque puede que no tengan un gran interés en las tareas cotidianas, sus decisiones son fundamentales para la asignación de recursos y la dirección del proyecto a alto nivel.
- Defensores (Bajo nivel de influencia, alto nivel de interés): Suelen aportar retroalimentación cualitativa y se les puede mantener comprometidos mediante actualizaciones periódicas y la participación en debates específicos.
- Apáticos (Bajo nivel de influencia, Bajo nivel de interés): Aunque no estén estrechamente implicados, ponerles al día sobre hitos significativos y solicitar su entrada sobre dificultades concretas puede aportar una perspectiva única.

El rol del jefe de prueba incluye la compilación de una lista detallada de implicados y la comprensión de la conexión de cada uno de ellos con las actividades de prueba, utilizando la matriz de implicados para mejorar la efectividad de las prácticas de gestión de la prueba.



Imagen 1 - Diferentes tipos de implicados

1.2.3 Gestión de la Prueba en un Modelo Híbrido de Desarrollo de Software

Los modelos híbridos de desarrollo de software integran elementos tanto de los enfoques secuenciales tradicionales como de las prácticas ágiles para adaptarse a las necesidades específicas de un proyecto o a las transiciones organizativas. Las siguientes son razones comunes para utilizar un modelo de desarrollo de software híbrido, aunque dependiendo de la organización y del proyecto, también puede haber otras razones:

- **Híbrido como transición a Ágil**
 - La transición de las metodologías tradicionales a Ágil puede suponer un reto debido a los cambios fundamentales en el flujo de trabajo, la cultura y la dinámica del equipo. Los modelos híbridos ofrecen un enfoque equilibrado que facilita esta transición al combinar la estructura de los métodos tradicionales con la flexibilidad de las prácticas ágiles.
- **Híbrido como adecuado al propósito**
 - Algunas organizaciones o proyectos pueden no ser capaces de pasar a Ágil. Los proyectos de alto riesgo pueden requerir tareas secuenciales para algunas cosas y prácticas ágiles para otras. Pueden utilizar un modelo híbrido según se ajuste a su propósito.

En un entorno híbrido, las actividades de gestión de la prueba pueden incluir:

- La evaluación de la comprensión y la capacidad del equipo para realizar una transición fluida entre las metodologías tradicional y ágil.
- Identificar los puntos fuertes y débiles en la adaptación a un enfoque híbrido.
- Asegurar que el equipo es experto (competente) en combinar procesos estructurados con flexibilidad ágil.
- Mejorar la colaboración entre el equipo de prueba y los implicados para gestionar mejor las pruebas dentro de los sprints y las fases de prueba tradicionales.
- Participar en esfuerzos coordinados, como scrum-of-scrums para probadores, para mantener el foco en la prueba a la vez que se contribuye a los objetivos generales de desarrollo.
- Hacer un seguimiento y revisar los esfuerzos y casos de prueba dentro de los sprint para asegurar que se alinean con las prácticas ágiles.

Se puede encontrar más información en (Fowler, 2010).

1.2.4 Actividades de Gestión de la Prueba para varios Modelos de Ciclo de Vida del Desarrollo de Software

Para alinear adecuadamente la prueba dentro del modelo de ciclo de vida de desarrollo de software (CVDS), un jefe de prueba debe comprender los distintos modelos de CVDS utilizados en su organización y utilizar ese conocimiento para alinear adecuadamente la prueba con las actividades de desarrollo.

La tabla siguiente muestra una comparación entre varias actividades de gestión de la prueba basadas en diferentes modelos de ciclo de vida de desarrollo de software (CVDS):

Aspecto	Modelo de desarrollo secuencial p. ej., V-Model	Modelo de desarrollo iterativo p. ej., SCRUM
Estimación	Estimación detallada temprana para cada nivel de prueba.	Estimación iterativa, parte de la planificación de historias por iteración.
Producto de prueba	Incluye estrategia, plan, casos, calendario e informes.	Se concentra en los criterios de aceptación y la definición de hecho; documentación mínima.
Roles	El jefe de prueba supervisa las decisiones y la gestión de la prueba.	Los roles están integrados; el facilitador o entrenador sustituye al jefe de pruebas tradicional.
Herramientas	Predominan las herramientas de gestión de la prueba adaptadas a las pruebas por fases.	Las herramientas para la integración continua/entrega continua (IC/EC) y la automatización son fundamentales.
Enfoque de Prueba	Planificadas con antelación, atendiendo a las fases del proyecto.	Integradas en las iteraciones, concentrándose en la adaptabilidad y la retroalimentación.
Automatización de la prueba	Implementada de forma estratégica, puede tener lugar en varias etapas.	Integrada desde el principio, con énfasis en la regresión automatizada en la integración continua/entrega continua (IC/EC)
Monitorización y suministro de información	Suministro de información basado en hitos, con cuadros de mando automatizados opcionales.	Suministro continuo de información con cuadros de mando en tiempo real y actualizaciones diarias de estado.
Métricas	Se concentra en las métricas tradicionales de las pruebas y en la gestión de defectos. (por ejemplo, ejecución de pruebas, tasas de defectos).	Incluye métricas ágiles para el seguimiento de iteraciones, además de las métricas tradicionales. (por ejemplo, velocidad del equipo, gráficos de quemado).

Tabla 1 - Actividades de gestión de la prueba para varios modelos de ciclo de vida de desarrollo de software (CVDS)

1.2.5 Actividades de Gestión de la Prueba en Distintos Niveles de Prueba

- **Prueba de componente:**
 - Definir el alcance, los objetivos y los criterios de completación de la prueba de componente.
 - Involucrar a los probadores en actividades que vayan más allá de los roles de prueba tradicionales, como las revisiones de código, en las que sus competencias analíticas añaden valor.
 - Coordinar con el equipo de desarrollo la resolución de las dificultades y la contribución a las pruebas unitarias.
- **Prueba de integración de componentes:**
 - Determinar las secuencias de integración y las combinaciones de pruebas en colaboración con el equipo de desarrollo, teniendo en cuenta el modelo de vida de desarrollo de software (CVDS), las herramientas y los procesos.
 - Supervisar el avance para asegurar que se alinea con las estrategias de pruebas de aceptación y de sistema.
 - Gestionar esta fase de forma cooperativa con los desarrolladores, teniendo en cuenta también las pruebas de integración de componentes.
- **Prueba de integración de sistemas:**
 - Asegurar que el alcance y los objetivos de la prueba de integración de sistemas son claros y están en sintonía con la evaluación del riesgo y los objetivos de calidad.
 - Mantener la supervisión del avance, los resultados y la gestión de las dificultades durante la prueba de integración de sistemas.
- **Prueba de sistema:**
 - Adaptar la planificación al modelo de ciclo de vida de desarrollo de software (CVDS), con una cuidadosa asignación de recursos, selección de herramientas y establecimiento de calendarios.
 - Para los proyectos ágiles, integrar la prueba de sistema con la prueba de historia iterativa, evitando fases de prueba distintas y asegurar que la prueba es continua e integrada. Mientras que, en los modelos secuenciales, las pruebas pueden seguir etapas planificadas.
- **Prueba de aceptación:**
 - Colaborar con los implicados para revisar y confirmar el cumplimiento de los criterios de aceptación y planificar las actividades de prueba, incluida la gestión de la prueba de usuario en prueba de aceptación de usuario (PAU).
 - Coordinar la logística de las pruebas de aceptación, facilitando las pruebas en las instalaciones del cliente, para asegurar que el producto satisface las necesidades del negocio y los estándares de calidad fuera del entorno de desarrollo.
 - Facilitar la resolución de cualquier dificultad de la prueba de aceptación de usuario (PAU) y guiar a los implicados en el proceso de aprobación del producto una vez cumplidos los criterios de aceptación.

1.2.6 Actividades de gestión de la prueba para distintos tipos de prueba

Una gestión de la prueba eficaz requiere un enfoque integrado que tenga en cuenta las demandas únicas de las pruebas funcionales, no funcionales, de caja negra y de caja blanca. Para los jefes de prueba que se ocupan de la prueba funcional, la atención se concentra en asegurar que todas las funcionalidades se prueben en profundidad y cumplan los requisitos definidos. La gestión de la prueba no funcional gira en torno a la verificación de atributos del sistema como el rendimiento y la seguridad. La gestión de la prueba de caja negra implica asegurar que las pruebas se concentran en el usuario y que se cubren todas las interacciones externas posibles. La gestión de la prueba de caja blanca hace hincapié en la comprensión de la estructura del código y en asegurar que las pruebas cubran a fondo la lógica interna.

- **Gestión de la Prueba Funcional:**
 - **Planificación Estratégica y Seguimiento del Avance:** elaboración de una estrategia de prueba detallada que se alinee con requisitos funcionales y los objetivos del proyecto, así como monitorizar el avance.
 - **Coordinación de Recursos:** Asignación eficaz de los recursos humanos y técnicos para cubrir todos los aspectos funcionales del sistema.
- **Gestión de la Prueba No Funcional:**
 - **Comparativa de rendimiento:** Establecimiento de puntos de referencia de rendimiento y gestión de las actividades de prueba que evalúan el sistema en función de estos criterios.
 - **Verificación de la conformidad:** Supervisión de las pruebas que aseguran que el sistema cumple estándares no funcionales como la seguridad, la usabilidad y la fiabilidad.
- **Gestión de la Prueba de Caja Negra:**
 - **Análisis de Cobertura de la Prueba:** Asegurar que las pruebas de caja negra cubren todos los escenarios de usuario y requisitos de negocio.
 - **Incorporación de retroalimentación:** Gestionar el proceso de recopilación de retroalimentación de los implicados para perfeccionar los enfoques de prueba de caja negra y la corrección de defectos.
- **Gestión de la Prueba de Caja Blanca:**
 - **Optimización de la Cobertura de Código:** Supervisión del uso de herramientas de cobertura de código para identificar vacíos en la prueba de caja blanca y dirección de los recursos para abordar estas áreas.
 - **Integración de conocimientos técnicos:** Gestionar la incorporación de conocimientos técnicos en el proceso de planificación de la prueba, asegurando que las pruebas se diseñan comprendiendo el funcionamiento interno de la aplicación.

1.2.7 Actividades de Gestión de la Prueba para la Planificación, Monitorización y Control

La gestión eficaz de la prueba es la piedra angular del éxito de cualquier esfuerzo de prueba, que abarca una amplia gama de actividades que requieren una planificación cuidadosa, un seguimiento atento y un control estratégico. Los jefes de prueba desempeñan un papel fundamental a la hora de garantizar que el proceso de prueba no sólo sea eficaz y eficiente, sino que también se adapte a las exigencias específicas del proyecto en cuestión.

- **Planificación de la Prueba:**

- **Definición integral del alcance:** Un plan de prueba debe elaborarse meticulosamente, incorporando una definición rigurosa del alcance. Esto incluye la identificación de todos los requisitos funcionales y no funcionales para asegurar una cobertura completa de la prueba. También implica tener en cuenta las implicaciones de las metodologías de prueba de caja negra y de caja blanca, asegurando que los casos de prueba desarrollados sean capaces de validar el sistema a prueba desde todos los ángulos.
- **Evaluación del Riesgo y Plan de Mitigación:** Una parte integral del plan de prueba es un marco sólido de gestión del riesgo. Los jefes de prueba deben llevar a cabo un análisis detallado de los riesgos, señalando las posibles vulnerabilidades y desafíos que podrían afectar tanto al flujo de trabajo del proyecto como al producto final. El desarrollo de estrategias de mitigación es crucial, e implica una planificación preventiva para eludir o minimizar estos riesgos de forma eficaz.
- **Estrategia de Asignación de Recursos:** La planificación de los recursos es otro elemento crítico. Esto va más allá de la mera asignación para definir la estructura del equipo, delimitar los roles y establecer protocolos de comunicación. En los entornos en los que los equipos están distribuidos, como ocurre con los modelos en sitio/fuera de sitio, esto adquiere especial importancia para mantener la sincronía y asegurar una colaboración sin fisuras.

- **Monitorización de la Prueba:**

- **Supervisión de la Ejecución:** La monitorización desempeña un papel central en el proceso de gestión de la prueba. Implica una revisión continua de la ejecución de la prueba con respecto al plan establecido, el seguimiento del avance de los casos de prueba y la gestión de cualquier defecto que pueda surgir. Ajustar las prioridades de las pruebas en función de la evaluación del riesgo y de la evolución en tiempo real asegura que la prueba se mantiene concentrada y alineada con las áreas más críticas.
- **Optimización de Herramientas y Entornos:** La selección y el uso con criterio de las herramientas de prueba y los entornos son cruciales para apoyar la estrategia de prueba. La monitorización continua garantiza que se integren eficazmente en la canalización de integración continua/entrega continua (IC/EC), lo que facilita las pruebas continuas y los bucles de retroalimentación inmediata que son vitales para el proceso de desarrollo ágil.
- **Colaboración con Desarrollo:** Mantener una estrecha relación de trabajo con el equipo de desarrollo es esencial para obtener resultados de la prueba satisfactorios. Esta colaboración debe respaldar un enfoque integral de la prueba, aprovechando las perspectivas tanto de caja blanca como de caja negra para abordar de forma preventiva las posibles dificultades.

- **Control de prueba:**

- **Gestión Adaptativa de Procesos:** El control de la prueba consiste en ajustar dinámicamente el proceso de prueba en respuesta a las nuevas perspectivas, los retos y la evolución de la dinámica del proyecto. Requiere que el jefe de prueba sea receptivo y flexible, capaz de implementar cambios en el enfoque de prueba que reflejen el estado actual del proyecto.
- **Gestión de Puertas de Calidad:** Un enfoque estructurado de la gestión de la puerta de calidad es fundamental. Esto incluye definir qué constituye una puerta de calidad dentro del ciclo de vida de la prueba y tomar decisiones informadas sobre el avance de la fase de prueba, que es fundamental para mantener la integridad del producto.

Al centrarse en estas actividades específicas dentro de la planificación de las pruebas, la supervisión de las pruebas y el control de las pruebas, los jefes de prueba pueden garantizar que el proceso de prueba esté bien definido, sea adaptable a los cambios del proyecto y dé como resultado un producto que cumpla tanto los requisitos del proyecto como las expectativas de las partes implicadas.

1.3 Prueba Basada en el Riesgo

Introducción

La prueba basada en el riesgo implica la identificación, evaluación, monitorización y mitigación del riesgo para impulsar la prueba. Estos riesgos son identificados por diversos implicados y pueden utilizarse para seleccionar y priorizar las pruebas. Cuanto más alto sea el nivel de riesgo, antes debería comenzar la prueba y más intenso y prolongado debería ser el esfuerzo de prueba.

1.3.1 La Prueba como Actividad de Mitigación del Riesgo

Un riesgo de producto es una situación potencial en la que pueden existir problemas de calidad en un producto. Cuando una prueba revela defectos, la prueba ha ayudado a mitigar el riesgo de producto al proporcionar el conocimiento de los defectos y las oportunidades para tratarlos antes de la entrega. Cuando la prueba no encuentra defectos, la prueba indica que el nivel de riesgo de producto es inferior al esperado.

Entre otras cosas, el jefe de prueba es responsable de ofrecer una evaluación correcta y fiable de la calidad del producto. Para ello, debe participar activamente en la gestión de riesgos del proyecto centrándose en los riesgos del proyecto relacionados con el aseguramiento de la calidad (por ejemplo, requisitos ambiguos que provoquen problemas importantes en la validación tardía, entornos de prueba insuficientes que obstruyan la ejecución de las pruebas).

La prueba basada en el riesgo concentra la prueba en los riesgos de calidad. Sigue el proceso genérico de gestión del riesgo, que consta de las siguientes actividades principales:

- Análisis del riesgo, consistente en la identificación del riesgo y la evaluación del riesgo.
- Control del riesgo, consistente en la monitorización del riesgo y la mitigación del riesgo.

Para concentrar la prueba en los riesgos de calidad, deben identificarse y evaluarse los riesgos de calidad. Para ser más eficaz, el análisis del riesgo debe incluir a diversos implicados. Al ser uno de los principales implicados en el análisis del riesgo de calidad, el jefe de prueba debe comprender y monitorizar estas actividades y ser capaz de moderarlas.

La monitorización de la prueba debe incluir la monitorización del riesgo. Además de monitorizar la evolución de los riesgos de calidad conocidos, debe incluir el análisis de cualquier riesgo de calidad nuevo y el ajuste del registro de riesgos.

El jefe de la prueba es una de las varias personas que impulsan la mitigación del riesgo de calidad. La mitigación del riesgo se distribuye entre varias actividades de prueba. Por ejemplo, los resultados del análisis del riesgo de calidad se utilizan en la planificación de la prueba para concentrar la prueba en las áreas correctas utilizando las técnicas correctas. En el análisis del riesgo, los niveles de riesgo guían la selección de las condiciones de prueba que deben cubrirse. En la ejecución de pruebas, la priorización basada en los riesgos rige la secuencia de ejecución de las pruebas.

1.3.2 Identificación de Riesgos de Calidad

La tarea del jefe de la prueba consiste en recopilar los riesgos de los implicados. Los implicados pueden identificar los riesgos de calidad mediante una o varias de las siguientes técnicas:

- Entrevistas a expertos
- Evaluaciones independientes
- Retrospectivas
- Talleres sobre riesgos
- Tormentas de ideas
- Listas de comprobación
- Remitirse a experiencias pasadas

Al involucrar a la muestra más amplia posible de implicados, el proceso de identificación del riesgo suele identificar la mayoría de los riesgos de producto importantes. La identificación de los que deben participar en esta fase es muy importante. Es esencial asegurarse de que la lista de implicados participantes sea completa y acordada con el jefe de proyecto. Una identificación del riesgo que omita a implicados clave puede ser muy problemática. La clave está en asegurar que todos los implicados relevantes tengan la oportunidad de participar. Si no pueden participar, al menos deberían tener la oportunidad de delegar la tarea. Cuando los implicados clave no estén representados, se puede recurrir a una reunión de inicio para determinar si faltan.

En la prueba basada en el riesgo, es importante comprender que el riesgo no se distribuye uniformemente dentro de los objetos de prueba. Por ejemplo, los componentes orientados al cliente de una aplicación de autoservicio pueden tener riesgos de usabilidad muy diferentes de los componentes de administración. Identificar los riesgos individuales de los distintos elementos de prueba es una tarea importante en la planificación de la prueba.

La identificación del riesgo a menudo produce subproductos, (es decir, la identificación de dificultades que no son riesgos de producto). Algunos ejemplos son las preguntas o dificultades generales sobre el producto o el proyecto, o los problemas en los documentos de referencia, como los requisitos y las especificaciones de diseño. Los riesgos de proyecto también suelen identificarse como un subproducto de la identificación del riesgo de calidad, pero no son el foco de la prueba basada en el riesgo. A menudo, el jefe de prueba puede desempeñar un papel importante a la hora de poner de relieve estos subproductos y dejar claro que la calidad es un asunto de interés para todos. Los requisitos deficientes o ausentes suelen ser un indicador de un problema más fundamental en la planificación y la preparación, ya que el aseguramiento de la calidad interviene en todo el ciclo de vida de desarrollo de software (CVDS).

1.3.3 Evaluación del Riesgo de Calidad

Una vez identificados los riesgos, se pueden evaluar. La evaluación del riesgo de calidad incluye la categorización de los riesgos por tipo (riesgo de producto o riesgo de proyecto) y por características de calidad afectadas.

La determinación del nivel de riesgo suele implicar la evaluación, para cada elemento de riesgo, de la probabilidad del riesgo y del impacto del riesgo.

Entre los factores que influyen en la probabilidad del riesgo para los riesgos de calidad se incluyen:

- Complejidad de la tecnología, las herramientas o la arquitectura del sistema.
- Madurez de la organización.
- Las dificultades del personal en cuanto a competencias, disponibilidad, motivación o trabajo autónomo, incluido el conocimiento del ciclo de vida de desarrollo de software (CVDS) en uso.
- Conflictos dentro del equipo.
- Problemas contractuales con los proveedores.
- Equipos distribuidos geográficamente.
- Débil liderazgo directivo o técnico.
- Presión de tiempo, recursos, presupuesto y gestión.
- Falta de actividades tempranas de aseguramiento de la calidad.
- Tasas de cambio elevadas de la base de prueba, el producto o el personal.

Factores que influyen en el impacto del riesgo:

- Frecuencia de uso de la prestación afectada.
- Criticidad de la prestación afectada.
- Criticidad del objetivo de negocio afectado.
- Daño a la reputación.
- Pérdida de ingresos del negocio.
- Posibles pérdidas financieras, ecológicas o sociales, o responsabilidad civil.
- Sanciones legales civiles o penales.
- Problemas de interconexión e integración.
- Carencia de soluciones provisionales razonables.
- Necesidades de protección.

El jefe de prueba combina la probabilidad del riesgo y el impacto del riesgo para determinar el nivel de riesgo.

Si el análisis del riesgo se basa en datos de riesgo amplios y estadísticamente válidos, lo apropiado es una evaluación cuantitativa. Por ejemplo, la probabilidad del riesgo puede expresarse como un porcentaje y el impacto del riesgo como una cantidad. En tal caso, el nivel de riesgo puede calcularse como el producto de estos dos factores. Sin embargo, normalmente, la probabilidad del riesgo y el impacto del riesgo sólo pueden determinarse cualitativamente en escalas ordinales, por ejemplo, como muy alto, alto, medio, bajo o muy bajo. Los valores de probabilidad del riesgo y de impacto del riesgo se combinan entonces a través de una matriz de riesgo para crear un nivel de riesgo agregado. Este nivel de riesgo agregado debe interpretarse como una calificación cualitativa y relativa sobre una escala ordinal.

A menos que el análisis del riesgo se base en datos de riesgo extensos y estadísticamente válidos, el análisis del riesgo será cualitativo, basado en las percepciones subjetivas de los implicados sobre la probabilidad del riesgo y su impacto.

1.3.4 Mitigación del riesgo de Calidad Mediante una Prueba Adecuada

En el desarrollo de software, la prueba es la actividad más importante de mitigación del riesgo de calidad y permite reducir la probabilidad de fallos. Otras posibles medidas de mitigación del riesgo son un plan de contingencia (por ejemplo, proporcionando soluciones provisionales), la transferencia del riesgo a un tercero (por ejemplo, el proveedor de un componente) o la aceptación del riesgo.

En la planificación de la prueba, el tiempo y el esfuerzo asociados al desarrollo y la ejecución de una prueba deben ser proporcionales al nivel de riesgo: Las pruebas para niveles de riesgo más elevados deben comenzar pronto y utilizar técnicas de prueba más rigurosas, mientras que las pruebas para niveles de riesgo más bajos pueden comenzar más tarde y deben utilizar técnicas de prueba menos rigurosas. Para mitigar mejor el riesgo global a través de la prueba, el jefe de prueba debe analizar los siguientes factores contextuales y seleccionar un enfoque de prueba que resulte adecuado:

- **Los elementos de prueba:** Diferentes elementos de prueba dentro de un objeto de prueba pueden tener diferentes niveles del mismo tipo de riesgo, por lo que no es necesario probar un objeto de prueba con un rigor uniforme.
- **Las características de calidad:** Los riesgos que afectan a características de calidad específicas deben mitigarse mediante tipos de prueba asociados que necesitan un esfuerzo de prueba, un entorno de prueba y unas competencias de prueba específicos.
- **Los niveles de prueba y los tipos de prueba:** Algunos riesgos sólo pueden probarse dinámicamente en determinados niveles de prueba; otros, mediante pruebas estáticas (por ejemplo, análisis estáticos y revisiones de código para la mantenibilidad), o mediante una combinación de ambas (por ejemplo, mediante una revisión de la arquitectura y pruebas dinámicas del sistema integrado para detectar vulnerabilidades de seguridad). Probar cada elemento de prueba lo antes posible mitiga el riesgo de encontrar defectos críticos en una fase tardía del ciclo de vida, lo que provocaría mayores costes de fallo interno y retrasos.
- **El ciclo de vida de desarrollo de software (CVDS):** Las actividades de prueba tienen sus propios criterios de entrada específicos. Los distintos CVDS los cumplen en diferentes momentos.
- **El equipo de prueba:** Las personas más cualificadas deberían probar los elementos de prueba con los niveles de riesgo más elevados.
- **Los requisitos normativos:** Algunos estándares relacionados con la seguridad (por ejemplo, el estándar IEC 61508), prescriben las técnicas de prueba y la cobertura requerida en función del nivel de integridad. El jefe de la prueba debe asegurar que se siguen estos estándares.

Además, el nivel de riesgo debe influir en las decisiones de control de calidad, como el uso de revisiones de productos de trabajo como los casos de prueba, el nivel de independencia de la prueba respecto al desarrollo y el alcance de las pruebas de regresión realizadas.

Durante **la monitorización de la prueba y el control de la prueba**, la prueba basada en el riesgo permite informar sobre el avance de la prueba en términos del nivel de riesgo residual en cualquier momento. Esto ayuda al equipo de desarrollo y a los implicados a monitorizar y controlar el desarrollo del software, incluida la toma de decisiones de entrega, basándose en el nivel de riesgo residual. Para ello es necesario informar de los resultados de prueba en términos de riesgos de una manera que las partes implicadas puedan entender.

Durante la implementación de pruebas, la priorización de las pruebas se basa en los niveles de riesgo. Durante la ejecución de pruebas, esto asegura la cobertura temprana de las áreas más críticas y la mitigación de los riesgos de mayor nivel.

- En algunos casos, las pruebas se priorizan para su ejecución en estricto orden descendente de los niveles de riesgo que cubren, empezando por el más alto. Este enfoque se denomina profundidad primero y es apropiado cuando es importante mitigar los riesgos de más alto nivel lo antes posible.
- Alternativamente, se asigna la máxima prioridad al menos a una prueba para cada riesgo. Todas las demás pruebas se priorizan en función de sus niveles de riesgo cubiertos. Este enfoque se denomina primero amplitud (“breadth-first”) y es apropiado cuando los implicados desean tener una visión global

de la calidad del producto lo antes posible. En la práctica, las pruebas suelen comenzar con el enfoque de profundidad primero, pero a medida que el tiempo se vuelve más limitado, se pasa al enfoque de amplitud primero, probando todos los elementos de riesgo restantes al menos una vez.

Tanto si la prueba basada en el riesgo se realiza primero en profundidad, primero en amplitud o de forma combinada, el tiempo asignado a la prueba puede consumirse sin que se realicen todas las pruebas previstas. Llegados a este punto, las pruebas basadas en el riesgo facilitan la provisión de una recomendación justificada a la dirección sobre si ampliar las pruebas o aceptar el riesgo restante.

1.3.5 Técnicas para la Prueba Basada en el Riesgo

Hay una serie de técnicas específicas con diversos grados de formalidad para la implementación de la prueba basada en el riesgo. La adecuación de una técnica depende de consideraciones relacionadas con el proyecto, el proceso y el producto. Existen dos tipos básicos de técnicas: pesadas o ligeras. En los sistemas de seguridad crítica, las técnicas de peso pesado se utilizan con mucha frecuencia. En aplicaciones no críticas para la seguridad, suelen emplearse técnicas ligeras.

Las técnicas de peso pesado son formales, utilizan procedimientos definidos y documentación detallada. Involucran a amplios grupos de implicados. La evaluación del riesgo dentro de las técnicas de peso pesado utiliza factores detallados de probabilidad del riesgo e impacto del riesgo, y fórmulas matemáticas para calcular la probabilidad del riesgo y el impacto del riesgo a partir de esos factores. Ejemplos de técnicas de peso pesado son:

- **Análisis del peligro:** Amplía el proceso analítico en sentido ascendente intentando identificar los peligros que subyacen a cada riesgo.
- **Coste de exposición:** Determina para cada elemento de riesgo de calidad la probabilidad de un fallo, el coste de una pérdida asociada a un fallo típico y el coste de la prueba para dichos fallos.
- **Análisis de Modos de Fallo y sus Efectos (AMFE) y sus variantes:** Identifica los riesgos de calidad, sus causas potenciales y sus efectos probables y, a continuación, asigna severidad, prioridad e índices de detección.
- **Análisis de árbol de defectos:** Relaciona los fallos potenciales con los defectos que pueden causar el fallo, después con los errores que pueden causar esos defectos, continuando hasta que se identifican las diversas causas raíz.

Por el contrario, las técnicas ligeras son menos profundas y requieren menos esfuerzo por parte del equipo de prueba y de los implicados. Al igual que las técnicas de peso pesado, también se basan en involucrar a los implicados, utilizando los resultados del análisis del riesgo como base para la planificación de la prueba y las condiciones de prueba. Sin embargo, el grupo de implicados puede no ser tan amplio y los factores de riesgo suelen reducirse a impacto del riesgo y probabilidad del riesgo en una escala ordinal. Algunas de estas técnicas, como la Prueba Sistemática de Software (SST) (Craig & Jaskiel, 2002), sólo pueden utilizarse cuando se proporcionan especificaciones de requisitos. Otras técnicas, como el Análisis y Gestión del Riesgo Pragmáticos (PRAM) (Black, 2009) y la Gestión del Riesgo de Producto (PRISMA) (van Veenendaal, 2012), utilizan los requisitos y/u otras especificaciones como entrada para el análisis del riesgo, pero pueden funcionar totalmente basadas en la entrada de los implicados..

1.3.6 Métricas de Éxito y Dificultades Asociadas a la Prueba Basada en el Riesgo

En una retrospectiva, el equipo de prueba debe medir hasta qué punto ha obtenido los beneficios de la prueba basada en el riesgo. En muchos casos, esto implica responder a algunas o a todas las preguntas siguientes mediante el uso de métricas y consultas:

- ¿Participaron o estuvieron representados los implicados relevantes en el análisis del riesgo?
- ¿Fue adecuada la participación de los implicados en el análisis del riesgo?
- Si se han producido incidentes críticos en producción que indiquen que se han escapado defectos críticos, ¿han sido resueltos?
- ¿Se encontraron la mayoría de los defectos de alta prioridad en una fase temprana de la ejecución de prueba?

- ¿El equipo de prueba ha sido capaz de explicar los resultados de la prueba a los implicados en términos de riesgo?
- ¿Las pruebas omitidas tenían un nivel de riesgo asociado inferior al de las ejecutadas?

En la mayoría de los casos, el éxito de la prueba basada en el riesgo da como resultado una respuesta afirmativa a todas estas preguntas. A largo plazo, deben fijarse objetivos de mejora del proceso en cuanto a métricas de éxito, además de esforzarse por mejorar la eficiencia del proceso de análisis del riesgo de calidad.

La gestión del riesgo suele encontrar dificultades inesperadas debido a complejidades que, a menudo, se ignoran.

- **Dificultad para evaluar el nivel de riesgo:** Estimar el impacto del riesgo y la probabilidad del riesgo puede resultar muy difícil. Solución: utilizar datos históricos y pedir su evaluación a los principales implicados en el proyecto.
- **Comienzo entusiasta:** La creación y el mantenimiento de un enfoque de prueba basado en el riesgo adecuado suelen descuidarse ante la gran presión por alcanzar el éxito a corto plazo. Solución: monitorización regular y suministro de información del riesgo a los implicados.
- **Déjà vu:** en cada proyecto se plantea el mismo conjunto de riesgos, lo que conduce a la complacencia respecto al riesgo. Solución: implicar a las personas adecuadas en la identificación del riesgo y mitigar sólo los riesgos que se consideren importantes.
- **Se están omitiendo riesgos clave:** La causa raíz de esta dificultad suele deberse a la implicación de personas inexpertas o inadecuadas en el proceso. Solución: involucrar a las personas adecuadas y formarlas.
- **Rotación de implicados:** Los implicados pueden cambiar con el tiempo y también pueden surgir nuevos riesgos, por lo que el análisis del riesgo es una actividad continua e iterativa y no debe realizarse una sola vez al principio.

1.4 La Estrategia de Prueba del Proyecto

Introducción

A lo largo de este programa de estudio, la estrategia de prueba de la organización se da por supuesta. El desarrollo y mantenimiento de una estrategia de prueba de la organización se considera en el contexto de la norma ISO/IEC/IEEE 29119-3 (donde se hace referencia como una “práctica de prueba de la organización”) y los programas de estudio para el ISTQB® Expert Level -Test Management, y el ISTQB® Certified Tester Agile Test Leadership at Scale.

Si no existe una estrategia de prueba de la organización o no cubre los aspectos requeridos, la gestión de la prueba debe tratar de aclarar los detalles que faltan con los implicados relevantes.

En el contexto de esta sección, la definición de una estrategia de prueba de proyecto es un ejemplo de cualquier tipo de estrategia de prueba detallada para un proyecto, una entrega, un producto o cualquier otro tipo de iniciativa de desarrollo o adquisición de sistemas. Una estrategia de prueba de proyecto (denominada “estrategia de prueba” en la norma ISO/IEC/IEEE 29119-3) describe el enfoque de prueba en un contexto específico para que puedan cumplirse los objetivos de la organización, en particular los relacionados con la calidad del producto y las actividades de prueba. También puede existir una estrategia de prueba para un único nivel de prueba o un tipo de prueba.

La estrategia de prueba de proyecto es el principal resultado de la planificación de la prueba de un proyecto y suele documentarse en un plan de prueba o como parte de otros documentos. Se recomienda documentar la estrategia de prueba, pero no necesariamente en forma de plan de prueba formal. La necesidad de documentación depende del contexto de la prueba (véase la sección 1.2, El contexto de la prueba). Cuando un proyecto sigue un modelo de desarrollo secuencial, la estrategia de prueba de proyecto suele documentarse, preferiblemente en el plan de prueba (véase ISO/IEC/IEEE 29119-3). La documentación también suele ser exigida por contratos, acuerdos, organismos reguladores o leyes.

1.4.1 Elección de un Enfoque de Prueba

La estrategia de prueba del proyecto guía todas las actividades de prueba dentro de un proyecto y detalla objetivos, recursos, calendarios y responsabilidades. Esta estrategia debe adaptarse a los requisitos exclusivos del proyecto. Las decisiones clave incluyen la selección de los niveles de prueba, los tipos de prueba y las técnicas de prueba para las pruebas estáticas y dinámicas y otras prácticas de prueba (por ejemplo, pruebas mediante guion, pruebas manuales, prueba contigua).

En teoría, todos los tipos de prueba pueden realizarse en cualquier nivel de prueba, y cualquier técnica de prueba puede aplicarse a cualquier tipo de prueba en cualquier nivel de prueba. En la práctica, la selección y combinación adecuadas de estas opciones tienen un impacto significativo en la efectividad y eficiencia de la prueba. Por ejemplo, la mantenibilidad del código puede evaluarse a menudo de forma más eficaz y eficiente mediante el análisis estático de código o la revisión de código. Por otro lado, la eficiencia de desempeño puede evaluarse mejor mediante pruebas de sistema mediante guion debido a la interacción de los componentes internos, o la utilidad de la funcionalidad puede validarse mejor con los usuarios mediante pruebas de aceptación manuales desarrolladas en colaboración. Elegir el mejor enfoque para una estrategia de prueba puede ser un proceso complejo en el que pueden influir la estrategia de prueba de la organización, el contexto del proyecto y otros aspectos.

Por lo tanto, seleccionar y combinar los niveles de prueba, los tipos de prueba y las técnicas de prueba es fundamental para que la estrategia de prueba de un proyecto sea eficaz, ya que influye significativamente en la eficiencia y la efectividad de la prueba.

1.4.2 Análisis de la Estrategia de Prueba de la Organización, del Contexto del Proyecto y de Otros Aspectos

La estrategia de prueba de la organización, el contexto del proyecto y los factores o restricciones adicionales relacionados con la prueba deben comprenderse en su totalidad para habilitar el desarrollo de una estrategia de prueba del proyecto.

Para elegir el enfoque de prueba adecuado, normalmente deben analizarse los siguientes factores:

- **Dominio:** El dominio para el que se creará o modificará el producto. Cualquier normativa, estándar o práctica específica del dominio puede cambiar el rigor de la prueba, la documentación requerida, así como su nivel de detalle. Por ejemplo, en los productos farmacéuticos y la medicina, el enfoque de prueba suele enfatizar una prueba de aceptación del usuario intensiva centrada en los riesgos para la salud del paciente, utilizando casos de prueba basados en los requisitos funcionales del usuario, mientras que la prueba de aceptación del usuario para las aplicaciones de seguros basadas en la web podría centrarse en la facilidad de uso y en aumentar la probabilidad de nuevos contratos de seguros mediante pruebas A/B.
- **Objetivos de la organización y características de calidad generales:** Los objetivos de la organización pueden incluir la necesidad de demostrar el valor de la prueba y de aumentar el grado de automatización de la prueba o las características de calidad del proceso de prueba, como el nivel de madurez de la prueba o la eficiencia de la detección de defectos. Éstos pueden determinar los niveles de prueba y los tipos de prueba que se necesitan.
- **Los objetivos del proyecto y el tipo de proyecto:** Los objetivos del proyecto (por ejemplo, los relativos al presupuesto, el tiempo y la calidad) y el tipo de proyecto (es decir, el desarrollo de un producto específico para el cliente frente al orientado al mercado) suelen contener restricciones y riesgos, así como oportunidades que afectan a la prueba. Por ejemplo, un presupuesto ajustado y limitaciones de tiempo pueden requerir el uso riguroso de prueba basada en el riesgo para establecer las prioridades de los casos de prueba para la ejecución de la prueba, mientras que el desarrollo de un producto específicamente para un cliente puede requerir pruebas que cubran los criterios de aceptación contractual predefinidos.
- **Recursos de prueba:** Debe tenerse en cuenta cualquier restricción relativa a la disponibilidad de los recursos de prueba, incluidas las herramientas de prueba, la infraestructura de la prueba, la tecnología y el entorno de desarrollo utilizados en el proyecto, así como el personal de prueba disponible y sus competencias. (véase la sección 3.1, El equipo de prueba). Recursos de prueba Por ejemplo, las pruebas basadas en la experiencia requieren probadores con un buen conocimiento del dominio; las aplicaciones móviles suelen tener que probarse en un número normalmente limitado de dispositivos diferentes; el uso de herramientas de prueba puede estar limitado por el número de licencias disponibles.
- El modelo de ciclo de vida de desarrollo del software utilizado para el proyecto: Para determinar los niveles de prueba adecuados, el esfuerzo de prueba, los criterios de entrada apropiados y los criterios de salida, consulte el programa de estudio de nivel básico V.4, Sección 2.2 y 5.1. Un ciclo de vida del software con integración continua requiere más pruebas automatizadas que el desarrollo único mediante un modelo en cascada y, por lo tanto, pueden utilizarse diferentes tipos de prueba y técnicas de prueba.
- **Interfaces con otros sistemas:** En un sistema de sistemas, es esencial alinear la prueba con otros equipos o proyectos y seleccionar los niveles de prueba adecuados, especialmente para la prueba de integración de sistemas. Por ejemplo, las pruebas basadas en el riesgo ayudan a priorizar y escalar las pruebas de integración de sistemas.
- **Disponibilidad de los datos de prueba:** Se deben tener en cuenta las restricciones en la disponibilidad de los datos de prueba, como la necesidad de datos de prueba anonimizados procedentes de producción, o la creación de datos de prueba específicos que pueden ser difíciles de proporcionar y necesitan ser validados, como los datos para las pruebas de IA. Por ejemplo, las pruebas basadas en modelos pueden apoyar la creación de datos de prueba y la gestión de datos de prueba.

El jefe de prueba debe determinar qué combinación de técnicas de prueba, niveles de prueba y tipos de prueba debe utilizarse como el mejor enfoque para satisfacer la estrategia de prueba de la organización, el contexto del proyecto y los factores o restricciones adicionales relacionados con la prueba.

1.4.3 Definición de los Objetivos de Prueba

Debe definirse un plan de prueba para cada proyecto de prueba que contenga, entre otros aspectos, el alcance de la prueba, los objetivos de la prueba y los criterios de salida. El plan de prueba puede establecerse a nivel de entrega, como plan de prueba de proyecto (también conocido como plan de prueba maestro) y, si es necesario, como plan de prueba de nivel para los distintos niveles de prueba. Además, pueden definirse planes de prueba para las distintas características de calidad, como un plan de prueba de seguridad o un plan de prueba de rendimiento. En los proyectos de desarrollo ágil de software y de desarrollo híbrido de software, se puede acordar un plan de pruebas de iteración. Para cada entrega e iteración, el alcance de las prestaciones funcionales y sus características no funcionales que deben entregarse se define en el plan de prueba y se acuerda con los implicados.

Asociados a las prestaciones entregadas para su prueba en un proyecto, deben definirse los objetivos de prueba del proyecto y los criterios de salida. Para ello se puede utilizar la metodología de objetivos S.M.A.R.T.:

- **S** specific → específico → El objetivo de prueba y el criterio de salida de un proyecto deben ser claros e inequívocos.
- **M** measurable → medible → Debe ser cuantificable y tener criterios específicos para medir el avance y determinar si se ha alcanzado.
- **A** achievable → alcanzable → Debe ser factible teniendo en cuenta los recursos disponibles, el calendario y las capacidades.
- **R** relevant → relevante → Debe estar en consonancia con los objetivos generales del proyecto.
- **T** timely → oportuno → Debe tener un marco temporal específico y un plazo definido para su compleción.

Los objetivos de prueba del proyecto deben abordar todos los aspectos de calidad y cantidad, siempre que sean medibles o evaluables. Algunos ejemplos de objetivos de prueba del proyecto son:

- Alcanzar los criterios de salida especificados dentro del plazo definido.
- Cumplir los objetivos de calidad de la organización (por ejemplo, medido como un indicador clave de rendimiento para el número de reclamaciones de los clientes por un producto).
- Cumplir las normas y reglamentos de la industria específica.
- Asegurar la disponibilidad de los datos sólo para los usuarios autorizados (por ejemplo, mediante derechos de acceso).
- Comprobar la completitud funcional, el rendimiento, la eficiencia, la portabilidad y la seguridad de la migración de datos.
- Mejorar el nivel de automatización de la prueba o de rendimiento en un porcentaje definido.
- Refactorizar el código con éxito y demostrar que no ha introducido nuevos defectos (por ejemplo, para eliminar el código fuente mal estructurado o la deuda técnica manteniendo la funcionalidad existente, probado mediante una prueba de regresión).
- Demostrar la seguridad de las interfaces (por ejemplo, validando los mensajes XML (Extensible Markup Language) con su definición de esquema XML para asegurar el rechazo de datos maliciosos).

Además del recuento y la medición de los objetivos de prueba del proyecto, debe tenerse en cuenta la evaluación del nivel de calidad por parte de los expertos del dominio y los implicados.

Dependiendo del contexto del proyecto y de los objetivos de prueba, a veces pueden ser necesarios varios entornos de prueba con los recursos y/o herramientas de prueba disponibles. Es posible que no todos los entornos de prueba estén disponibles al mismo tiempo. Esto debe tenerse en cuenta al formular los objetivos de prueba alcanzables y los criterios de salida.

Dependiendo del contexto del proyecto, habrá que tener en cuenta otros factores a la hora de definir los objetivos de prueba y el alcance de las pruebas del proyecto, como se describe en la sección 1.2 de este programa de estudio, El contexto de la Prueba.

BORRADOR

1.5 Mejora del Proceso de Prueba

Introducción

La prueba es una parte importante del desarrollo de software y a menudo supone al menos el 30-40% de los costes totales del proyecto. Junto a los muchos retos (técnicos) a los que se enfrentan los proyectos de software (por ejemplo, el aumento de la complejidad y el tamaño, las nuevas tecnologías, la gran variedad de dispositivos y sistemas operativos y las vulnerabilidades de seguridad), existe la necesidad de optimizar la efectividad y la eficiencia de la prueba, así como la necesidad de mejorar los procesos de prueba en consecuencia. Aprender de las buenas prácticas existentes y de los errores propios permite mejorar el proceso de prueba y hacer que los proyectos tengan más éxito.

Un proceso de mejora a nivel organizativo suele ser más útil que un proceso de mejora a nivel de proyecto o de equipo. Sin embargo, también es posible y beneficioso aplicar la mejora del proceso a nivel de proyecto o de equipo, pero debe adaptarse a las necesidades del proyecto o del equipo. Una mejora de la prueba puede iniciarse, por ejemplo, por insatisfacción con los resultados de la prueba actual, defectos inesperados, circunstancias cambiantes, un resultado de prueba comparativa o falta de comunicación. Existen diferentes técnicas para mejorar las pruebas (Bath & van Veenendaal, 2014). A continuación, se describen algunas de estas técnicas. Las técnicas descritas en este programa de estudio pueden aplicarse tanto a modelos de desarrollo secuencial como a modelos de desarrollo ágil de software/modelos de desarrollo incremental. El programa de estudio ISTQB® Expert Level Improving the Test Process Syllabus ofrece una visión más profunda.

1.5.1 El Proceso de Mejora de la Prueba (IDEAL)

Una vez que se ha acordado que los procesos de prueba deben mejorarse, las actividades de implementación de la mejora de la prueba que deben adoptarse para esta actividad pueden definirse como en el modelo IDEAL, que se basa en ideas similares a las del conocido ciclo planificar-hacer-comprobar-actuar². IDEAL es un acrónimo de los términos en inglés “Initiating, Diagnosing, Establishing, Acting and Learning”. IDEAL significa iniciar, diagnosticar, establecer, actuar y aprender.

Aunque IDEAL se definió originalmente para apoyar las actividades de mejora a nivel organizativo, también puede aplicarse a nivel de proyecto o de equipo de desarrollo Ágil de Software. En el contexto de un proyecto, los objetivos de las actividades (véase a continuación) están por alcanzar. La principal diferencia es probablemente la fase de iniciación, que es mucho menor a nivel de proyecto o de equipo que a nivel organizativo. Diagnosticar a través de una retrospectiva y establecer un plan serán probablemente mucho menores que para un nivel organizativo. La actuación y el aprendizaje también serán relevantes a nivel de proyecto o de equipo.

- **Iniciar** (“initiating”) el Proceso de Mejora
 - Al inicio del proceso de mejora, los implicados acuerdan los objetivos y el alcance de las mejoras del proceso.
- **Diagnosticar** (“Diagnosing”) la Situación Actual
 - Se evalúa el proceso de prueba actual para identificar posibles mejoras. La evaluación suele realizarse en función de un marco de trabajo estándar en el caso de la mejora del proceso de pruebas basado en modelos (véase el apartado 1.5.2, Mejora del proceso de pruebas basado en modelos) o puede basarse en un análisis de métricas específicas en el caso de la mejora del proceso de pruebas basado en análisis (véase el apartado 1.5.3, Enfoque de mejora del proceso de pruebas basado en análisis).
- Establecer (“Diagnosing”) un Plan de Mejora del Proceso de Prueba
 - Un plan de mejora del proceso de prueba puede ser un documento formal que enumere todas las acciones detalladas que deben realizarse para lograr mejoras. Dependiendo del contexto,

² Consultar Anexo C

el plan puede ser muy informal y muy ligero. La lista de posibles mejoras del proceso debe priorizarse. La priorización puede basarse en el retorno de la inversión (ROI), los riesgos, la alineación con las estrategias del proyecto o del equipo, y/o los beneficios cuantitativos o cualitativos medibles que deben lograrse.

- Actuar (“acting”) para Implementar la Mejora del Proceso de Prueba
 - Se implementa el plan de mejora del proceso de prueba para la entrega de las mejoras. Esto suele incluir la formación y la realización de pilotos de los procesos modificados y su despliegue completo en el proyecto o equipo.
- Aprender (“learning”) del Programa de Mejora
 - Una vez desplegadas plenamente las mejoras del proceso, es esencial verificar qué beneficios, previstos o imprevistos, se han recibido. Una vez aprendido lo que funcionó y lo que no funcionó, debemos actuar en función de esta información y sólo a partir de entonces podrá iniciarse el siguiente ciclo de mejora.

1.5.2 Mejora del Proceso de Prueba Basada en Modelos

Una premisa tanto para la mejora del proceso de prueba basado en modelos como para la mejora basada en análisis es la suposición de que la calidad de los procesos que se aplican y utilizan influye en gran medida en la calidad del producto. Cuando se aplica la mejora del proceso de prueba basada en modelos, se utiliza un modelo de mejora de la prueba. Los modelos de mejora de la prueba se basan en buenas prácticas y organizan la mejora de la prueba de forma escalonada.

Han surgido varios modelos de proceso recomendados que apoyan la mejora del proceso de prueba. Entre ellos se encuentran Test Maturity Model integration (TMMi®) y TPI NEXT®.

La mejora basada en modelos también puede aplicarse a nivel de proyecto. En tales casos, la evaluación y el proceso de mejora se centran específicamente en los procesos de prueba o en las áreas clave definidas en el modelo que se relacionan con las actividades a nivel de proyecto (por ejemplo, la planificación de la prueba y el diseño de la prueba) y, a menudo, omiten en gran medida las que se encuentran a nivel de organización (por ejemplo, la política de prueba y la organización de la prueba). Alternativamente, también se pueden adaptar adecuadamente al contexto del proyecto las prácticas que abordan el nivel organizativo.

Para más información sobre la mejora del proceso de prueba basado en modelos, consulte el programa de estudio ISTQB® Expert Level Improving the Test Process.

- **Test Maturity Model integration**
 - El TMMi® (van Veenendaal & Cannegieter, 2011) (van Veenendaal, 2020) se compone de cinco niveles de madurez. Cada nivel de madurez, excepto el nivel 1 del TMMi®, contiene áreas del proceso de prueba y objetivos de mejora. Además, para facilitar y apoyar su implementación, TMMi® contiene prácticas, subprácticas y ejemplos. El TMMi® se desarrolló inicialmente para complementar el Capability Maturity Model Integration (CMMI®), pero hoy en día se utiliza ampliamente con independencia del CMMI®.
 - Para facilitar y apoyar la actualización del TMMi® en el desarrollo ágil de software, se ha desarrollado una directriz específica que explica cómo se puede utilizar y aplicar el TMMi® de forma beneficiosa en el desarrollo ágil de software.
 - Para más información sobre TMMi®, consulte www.tmmi.org.
- **TPI NEXT®**
 - El modelo TPI NEXT® (van Ewijk, 2013) define 16 áreas clave, cada una de las cuales cubre un aspecto específico del proceso de prueba (por ejemplo, estrategia de prueba, métrica de prueba, herramientas de prueba y entorno de prueba). En el modelo se definen cuatro niveles de madurez para cada una de las 16 áreas clave.
 - Se definen puntos de control específicos para evaluar cada área clave en cada uno de los niveles de madurez. Los resultados de la evaluación se resumen y visualizan mediante una matriz de madurez que abarca todas las áreas clave.

- Para más información sobre TPI NEXT®, consulte www.tmap.net

1.5.3 Enfoque de Mejora del Proceso de Prueba Basada en Modelo Analítico

Mediante un enfoque de mejora basado en modelos, como el descrito en el apartado anterior, se introducen mejoras comparando el enfoque de prueba de un proyecto o equipo con las buenas prácticas externas. Los enfoques analíticos identifican los problemas basándose en los datos del propio proyecto o equipo. Del análisis de estos problemas se pueden obtener las mejoras adecuadas. Los enfoques analíticos pueden utilizarse junto con un enfoque basado en modelos para verificar los resultados y aportar diversidad.

Los problemas pueden identificarse utilizando datos cuantitativos y cualitativos. La sección 1.5.3 de este programa de estudios, Enfoque de mejora del proceso de prueba, presenta enfoques analíticos que utilizan principalmente datos cuantitativos del proceso de prueba y datos de defectos para evaluar el enfoque actual. La sección 1.5.4 de este programa de estudio, Retrospectivas, presenta las retrospectivas, en las que se recopilan datos cualitativos sobre lo que funciona bien y lo que no entre los miembros del equipo de desarrollo y de prueba.

El análisis de datos es importante para la mejora del proceso objetivo de prueba y un valioso apoyo a las evaluaciones puramente cualitativas, que de otro modo podrían dar lugar a recomendaciones imprecisas que no están respaldadas por datos. La aplicación de un enfoque analítico a la mejora suele implicar un análisis cuantitativo del proceso de prueba para identificar áreas problemáticas y establecer objetivos específicos del proyecto. Es necesario definir y medir los parámetros clave para evaluar el proceso de prueba y valorar si las mejoras han tenido éxito.

Algunos ejemplos de enfoques analíticos son:

- Análisis de la causa raíz.
- Análisis mediante medidas, métricas e indicadores
- El enfoque GQM (Goal-Question-Metric)

El análisis de la causa raíz es el estudio de los problemas para identificar sus causas raíz. Esto permite identificar soluciones que eliminen las causas de los problemas en lugar de limitarse a abordar los síntomas evidentes inmediatos. Un posible procedimiento de análisis consistiría en seleccionar un conjunto adecuado de defectos, identificar agrupamientos en estos datos y utilizar diagramas causa-efecto (también llamados diagramas de Ishikawa o de espina de pescado) para identificar las causas raíz de los agrupamientos de defectos importantes. A continuación, se obtienen mejoras para evitar que se produzcan defectos similares.

Las medidas, métricas e indicadores se utilizan de forma cuantitativa para evaluar lo bien que se realiza el proceso de prueba en el proyecto o equipo. Los atributos clave del proceso de prueba que hay que tener en cuenta son la efectividad, la eficiencia y la previsibilidad. Para cada uno de estos atributos se pueden seleccionar una o varias métricas. Mediante la recopilación y el análisis de los datos correspondientes, se pueden identificar las áreas clave que requieren mejoras.

El enfoque GQM (Basili, et al., 2014) (van Solingen & Berghout, 1999) proporciona un marco para definir y analizar métricas adaptadas a las necesidades de información de los implicados relevantes del proyecto. Los objetivos de medición definen un aspecto de calidad de un objeto que necesita ser medido para un propósito, una perspectiva y un contexto concretos. Estos objetivos se refinan en preguntas que definen el aspecto de calidad desde el punto de vista de los implicados. A continuación, se seleccionan las métricas que proporcionan la información necesaria para responder a la pregunta. Los datos recogidos para las métricas responden a las preguntas, para evaluar el objetivo de medición y satisfacer las necesidades de información de los implicados.

Para más información sobre estos enfoques de mejora del proceso de prueba basados en el análisis, consultar el programa de estudio ISTQB® Expert Level Improving the Test Process Syllabus.

1.5.4 Retrospectivas

Las retrospectivas son reuniones en las que un equipo revisa sus métodos y su colaboración, recopila las lecciones aprendidas (buenas y malas) y decide los cambios y las acciones para lograr mejoras (tanto para las cuestiones relacionadas con las pruebas como para las que no lo están). Las retrospectivas abordan temas como el proceso

. En este contexto, las retrospectivas tienen como objetivo generar lecciones aprendidas para gestionar mejor los proyectos futuros. En el Desarrollo Ágil de Software las retrospectivas se suelen realizar al final de cada iteración para discutir qué ha tenido éxito y qué necesita mejorarse, y cómo pueden incorporarse esas mejoras en la siguiente iteración. Las retrospectivas son realizadas por todo el equipo, por lo que apoyan el enfoque de equipo completo y fomentan la mejora continua. Tenga en cuenta que a veces se requieren retrospectivas específicas para abordar dificultades de la prueba.

Una retrospectiva típica consta de los siguientes pasos:

- **Introducción:** Se revisan el objetivo y la agenda de la retrospectiva, y se crea una atmósfera de confianza mutua para poder discutir los problemas sin culpar ni juzgar.
- **Recopilar datos:** Se recogen datos sobre lo ocurrido durante la iteración o el proyecto. Es posible recopilar datos cualitativos, como una cronología de los acontecimientos clave en la que se identifiquen las dificultades y se enumere cómo se siente cada miembro del equipo al respecto. Además, se pueden presentar datos cuantitativos procedentes de métricas; por ejemplo, los datos sobre el avance de la prueba, la detección de defectos, la efectividad de la prueba, la eficiencia de la prueba y la previsibilidad pueden proporcionar una visión objetiva de la prueba del proyecto o de la iteración.
- **Derivar mejoras:** Los datos recopilados se analizan para comprender la situación actual y generar ideas de mejora. Por ejemplo, se puede aplicar el análisis de la causa raíz para identificar las causas de los problemas detectados y celebrar una tormenta de ideas para generar ideas sobre cómo resolver las causas raíz.
- **Decidir las acciones de mejora:** Se obtienen y se priorizan las acciones para implementar las ideas de mejora. Se definen un plan de mejora y las responsabilidades. Se pueden definir objetivos y métricas asociadas para evaluar el impacto de las acciones en los problemas identificados. Implementar demasiadas mejoras a la vez es difícil de gestionar con pasos verificables.
- **Cerrar la retrospectiva:** En este último paso, se revisa la propia retrospectiva para identificar los puntos fuertes y las mejoras del proceso retrospectivo. La retrospectiva se realiza con regularidad, especialmente en el desarrollo ágil de software. La mejora continua también se aplica a la propia retrospectiva.

Es importante documentar adecuadamente los resultados de una retrospectiva. En un modelo de desarrollo secuencial, los hallazgos, las conclusiones y las recomendaciones deben distribuirse y comunicarse de forma comprensible a los miembros de la organización. En el Desarrollo Ágil de Software, los problemas y las acciones también deben documentarse para permitir la revisión de las acciones y su posible impacto en los problemas de la siguiente iteración.

Los probadores, al formar parte del equipo (del proyecto), aportan su perspectiva única. Pueden plantear problemas relacionados con la prueba (y otros) y estimular al equipo a pensar en posibles mejoras.

Para más información, consultar (Derby & Larsen, 2006).

1.6 Herramientas de Prueba

Introducción

Existen tres tipos de herramientas de negocio:

- Herramientas comerciales.
- Herramientas de código abierto.
- Herramientas a medida.

Cuando se selecciona una herramienta de negocio, deben tenerse en cuenta todos los requisitos y normativas de la organización y de los implicados del negocio.

También existen herramientas técnicas como las herramientas de automatización de la prueba, las herramientas de gestión de la prueba y muchas más.

En el programa de estudio de nivel básico V.4 de ISTQB® encontrará ejemplos de uso de herramientas de prueba.

1.6.1 Buenas Prácticas para la Introducción de Herramientas

Esta sección contiene los pasos necesarios para la evaluación e introducción de una herramienta de prueba. Un jefe de prueba puede participar en la introducción de una herramienta o puede fomentar o facilitar el proceso de introducción. Los jefes de prueba suelen ser responsables de una herramienta de prueba dedicada o de cualquier herramienta relacionada con la prueba, como una herramienta de gestión de requisitos, de gestión de defectos o de monitorización.

Existen buenas prácticas y consideraciones genéricas a la hora de evaluar y seleccionar una herramienta de prueba. Estas prácticas y consideraciones incluyen lo siguiente:

- Identificar las oportunidades de mejora del proceso, con el apoyo de las herramientas adecuadas.
- Comprender las tecnologías utilizadas en una organización y seleccionar una herramienta que sea compatible con dichas tecnologías.
- Comprender cómo se integra técnica y organizativamente una herramienta en el ciclo de vida de desarrollo de software (CVDS).
- Evalúe la herramienta en función de requisitos claros y criterios objetivos.
- Evalúe al proveedor si está teniendo en cuenta el uso de una herramienta comercial.
- Evalúe la compatibilidad con herramientas no comerciales (por ejemplo, de código abierto).
- Identifique los requisitos internos para entrenar, orientar o formar en el uso de la herramienta.
- Tener en cuenta los pros y los contras de los distintos modelos de licencia
- Como paso final, realizar una evaluación de prueba de concepto.

Entre las buenas prácticas genéricas en la adopción y puesta en marcha de una herramienta se incluyen:

- Ejecutar un proyecto piloto para validar los criterios de selección y los requisitos y para evaluar cómo encaja la herramienta con los procesos y prácticas existentes.
- Adaptar y mejorar los procesos para que encajen con el uso de la herramienta, también adaptar la herramienta a los procesos existentes, si es necesario.
- Definir directrices para el uso de la herramienta.
- Proporcione formación, entrenamiento y orientación a los usuarios de la herramienta.
- Desplegar la herramienta en la organización en incrementos.
- Implantar una forma de recopilar información del uso real de la herramienta para futuras mejoras.
- Definir la propiedad de la herramienta.

1.6.2 Aspectos Técnicos y de Negocio para las Decisiones sobre la Herramienta

Existen múltiples factores que influyen en la decisión sobre la implementación y el uso de una herramienta. Para un jefe de prueba es importante conocerlos y abordarlos.

- **Normativa y seguridad:** Las organizaciones que desarrollan software de seguridad crítica o de misión crítica, o que están sujetas al cumplimiento de normativas, pueden preferir las herramientas comerciales, ya que con más frecuencia cumplen los requisitos exigidos y suelen poseer la certificación adecuada.
- **Aspectos financieros:** Las herramientas de código abierto suelen tener un coste inicial más bajo gracias al apoyo y desarrollo de la comunidad. Las herramientas comerciales pueden tener un precio de compra que se paga una sola vez, así como costes de licencia recurrentes. El coste inicial de una herramienta a medida es difícil de determinar porque depende de los requisitos y de la fase de desarrollo de la herramienta. Además de los costes iniciales, hay que calcular y tener en cuenta el coste de formación y mantenimiento a lo largo de la vida útil de una herramienta. Todas las herramientas pueden tener unos costes de mantenimiento elevados.
- **Requisitos de implicados:** Es importante reunir los requisitos de todos los implicados para evaluar e identificar la herramienta más adecuada. Las herramientas comerciales y las de código abierto no necesariamente cumplen todos los requisitos en detalle. Las herramientas a medida pueden ser la mejor opción para satisfacer todos los requisitos individuales y en los casos en los que ninguna otra herramienta proporcione la funcionalidad requerida.
- **Panorama de software existente y estrategia de herramientas:** Debe evaluarse la composición existente de herramientas (panorama del software) y la estrategia de herramientas asociada, ya que puede haber proveedores preferidos o bloqueados, sistemas integrados que tengan dependencias con otros productos o un modelo especial de soporte de servicio completo para todo el panorama del software con normativas específicas.

1.6.3 Consideraciones sobre el Proceso de Selección y Evaluación del Retorno de la Inversión

Las herramientas de prueba pueden ser una inversión a largo plazo, que quizás se extienda a lo largo de muchas iteraciones de un único proyecto, y/o aplicables a muchos proyectos. Por lo tanto, una posible herramienta debe tenerse en cuenta desde distintos puntos de vista.

- En lo que respecta a la alta dirección, se requiere un ROI positivo.
- En lo que respecta al equipo de soporte y operaciones, se prefiere un número limitado, pero necesario, de herramientas utilizadas en toda la organización. Mantener un número mayor de herramientas, hacer un seguimiento de sus licencias y gestionar la cartera de herramientas no debería suponer ni un gasto ni un consumo de tiempo.
- Para los responsables de proyecto, la herramienta debe añadir un valor medible al proyecto o a la organización.
- Para las personas que utilizan la herramienta, la usabilidad es muy importante. La usabilidad incluye, por ejemplo, el apoyo a determinadas tareas, la capacidad de ser aprendido y la operabilidad.
- Para los miembros del personal operativo, la mantenibilidad es importante.

Las prestaciones deben analizarse para cada tipo de herramienta de negocio y técnica. En ese análisis influyen diferentes perspectivas e intereses: la gestión de la prueba, el análisis (técnico) de la prueba, la automatización de la prueba o el desarrollo. La persona de la organización responsable de la herramienta (el propietario de la herramienta) debe asegurarse de que se realiza el análisis y se tienen en cuenta los puntos mencionados anteriormente.

Todas las herramientas introducidas en el proceso de prueba deben asegurar también un retorno de la inversión positivo para la organización. Es responsabilidad del jefe de prueba ocuparse del cálculo y posterior evaluación del ROI. En el Desarrollo Ágil de Software puede ser responsabilidad de todo el equipo de desarrollo.

Antes de adquirir o construir una herramienta debería realizarse un análisis de coste-beneficio para asegurar que resulta en un beneficio para la organización. Este análisis debería tener en cuenta tanto los costes recurrentes como los no recurrentes.

Entre las actividades y los costes no recurrentes se encuentran los siguientes:

- Definir y determinar los requisitos de la herramienta para cumplir los objetivos.
- Evaluar y seleccionar la herramienta y el proveedor adecuados, prueba de concepto.
- Adquirir, adaptar o desarrollar la herramienta para su uso inicial.
- Definir directrices para el uso de la herramienta.
- Formación inicial para la herramienta.
- Integración de la herramienta en el panorama de herramientas existente.
- Adquirir el hardware/software necesario para dar soporte a la herramienta.

Entre las actividades y costes recurrentes se encuentran los siguientes:

- Cuotas recurrentes de licencia y soporte.
- Costes de mantenimiento.
- Costes de formación continua.
- Portar la herramienta a diferentes entornos.

También hay que tener en cuenta los costes de oportunidad. Esto significa que el tiempo dedicado a evaluar, administrar, formar y utilizar la herramienta podría haberse empleado en tareas de prueba reales. Por lo tanto, es posible que se necesiten más recursos de prueba antes de poder utilizar la herramienta para las actividades previstas.

En el proceso de selección de herramientas se deberían tener en cuenta los siguientes riesgos en relación con el retorno de la inversión:

- La inmadurez de la organización puede conducir a un uso ineficaz de la herramienta.
- Los cambios en la política de mantenimiento del proveedor pueden aumentar la carga de trabajo.
- Costes más elevados de lo esperado.
- Menor beneficio del esperado.

Las siguientes ventajas pueden aplicarse a las herramientas de prueba:

- Reducción del trabajo manual repetitivo (por ejemplo, prueba de regresión).
- Aceleración del tiempo del ciclo de prueba mediante la automatización.
- Ahorro de costes en la ejecución de pruebas gracias a la disminución de las actividades manuales.
- Aumento de la cobertura de determinados tipos de prueba admitidos por la herramienta.
- Reducción de los errores humanos por la disminución de las actividades manuales.
- Acceso más rápido a la información sobre las pruebas.

En la versión 4 del programa de estudio “ISTQB® Certified Tester, Foundation Level” y en el programa de estudio “ISTQB® Certified Tester, Test Automation Engineer” se pueden encontrar beneficios y riesgos adicionales, especialmente para las herramientas de automatización de la prueba.

En general, una organización de prueba rara vez utiliza una única herramienta. El ROI total de una organización suele ser una combinación del ROI de todas las herramientas que se utilizan para las pruebas. Las herramientas necesitan compartir información y trabajar de forma cooperativa. Es aconsejable una estrategia integral a largo plazo para las herramientas de prueba que incluya riesgos, costes y beneficios.

1.6.4 Ciclo de Vida de las Herramientas

Existen cuatro etapas diferentes en el ciclo de vida de una herramienta. Debe designarse un administrador de la herramienta para asegurar que las actividades de estas etapas se definen, se llevan a cabo y se gestionan.

- **Adquisición:** En primer lugar, se ha tomado la decisión de seleccionar una herramienta. En el segundo paso, es necesario asignar un propietario de la herramienta. Esta persona toma las decisiones sobre el uso de la herramienta (por ejemplo, las convenciones de nomenclatura de los productos de trabajo y dónde se almacenarán estos productos de trabajo). Tomar estas decisiones desde el principio puede suponer una diferencia significativa en la rentabilidad final de la herramienta.
- **Soporte y mantenimiento:** El propietario de la herramienta es responsable de su mantenibilidad. La responsabilidad de las actividades de mantenimiento debe recaer en el administrador de la herramienta o en un grupo dedicado a ella. En el caso de la interoperabilidad, deben tenerse en cuenta el intercambio de datos y los procesos de cooperación y comunicación. Asimismo, se requieren decisiones sobre la copia de seguridad y la recuperación de los artefactos relacionados con la herramienta.
- **Evolución:** A medida que pasa el tiempo, el entorno, las necesidades de negocio o las decisiones de los proveedores pueden requerir cambios en la herramienta. Cuanto más complejo sea el entorno operativo de una herramienta, más fácil será que un cambio altere su uso.
- **Retiro:** Al final de su vida útil, la herramienta debe retirarse. En la mayoría de los casos, la funcionalidad suministrada por la herramienta será reemplazada y los datos deberán conservarse y/o archivers. Esto puede deberse a una decisión del proveedor o a que se ha llegado a un punto en el que los beneficios y las oportunidades de pasar a una nueva herramienta superan sus costes y riesgos.

1.6.5 Métrica de Herramientas

Las métricas objetivas extraídas de las herramientas se diseñan y recopilan en función de las necesidades del equipo de prueba y de otros implicados. Las herramientas de prueba capturan sobre todo datos que resultan de valor en tiempo real y reducen el esfuerzo requerido para recopilar datos. Estos datos se utilizan para gestionar el esfuerzo global de las pruebas e identificar áreas de optimización.

Las distintas herramientas se concentran en recopilar diferentes tipos de datos. Algunos ejemplos son:

- Las herramientas de gestión de la prueba pueden suministrar una variedad de métricas diferentes relacionadas con los elementos de prueba disponibles, las pruebas, las pruebas planificadas, así como el estado de ejecución de la prueba actual y pasada (por ejemplo, aprobada, fallida, omitida, bloqueada o planificada).
- Las herramientas de gestión de requisitos proporcionan trazabilidad en relación con la cobertura de los requisitos mediante casos de prueba pasados y fallidos.
- Las herramientas de gestión de defectos pueden proporcionar información sobre defectos, como el estado, la severidad, la prioridad y la densidad de defectos de los elementos de prueba. Otros datos valiosos, como el porcentaje de detección de defectos, los niveles de prueba en los que se introducen los defectos y el plazo de ejecución de los defectos detectados ayudan a impulsar la mejora del proceso, pero puede que no todos sean proporcionados únicamente por la herramienta de gestión de defectos.
- Las herramientas de análisis estático, entre otras, suministran métricas relacionadas con la complejidad del código.
- Las herramientas para prueba de rendimiento pueden proporcionar información valiosa, como los tiempos de respuesta y las tasas de fallos bajo cargas máximas.
- Las herramientas de cobertura de código ayudan a comprender qué partes del objeto de prueba han sido practicadas por la prueba.
- Aunque las herramientas de prueba pueden utilizarse para recopilar métricas, también deben monitorizarse a sí mismas. En este contexto, puede medirse la calidad del proceso de prueba (por ejemplo, el número de defectos encontrados con y sin herramientas y la cobertura de requisitos).
- La eficiencia de la prueba (por ejemplo, la duración de la ejecución de la prueba y el número de pruebas ejecutadas).

Para más detalles sobre la colección y el uso de métricas, consultar el apartado 2.1 de este programa de estudio, Métricas de Prueba.

2 Gestión de Producto

Duración: 390 minutos

Palabras Clave³

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	anomalía	anomaly
ISTQB	defecto	defect
ISTQB	informe de defecto	defect report
ISTQB	flujo de trabajo de defectos	defect workflow
ISTQB	fallo	failure
ISTQB	métrica	metric
ESPDOM	póker de planificación	planning poker
ISTQB	estimación de la prueba	test estimation
ISTQB	objetivo de prueba	test objective
ISTQB	avance de la prueba	test progress
ESPDOM	estimación de tres puntos	three-point estimation
ESPDOM	Delphi de banda ancha	Wideband Delphi

³ Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de Aprendizaje para “Capítulo 2”

2.1 Métrica de Prueba

- TM-2.1.1 (K2)** Dar ejemplos de métricas para alcanzar los objetivos de prueba.
- TM-2.1.2 (K2)** Explicar cómo controlar el avance de la prueba utilizando métricas de prueba.
- TM-2.1.3 (K4)** Analizar los resultados de prueba para crear informes de prueba que empoderen a los implicados en la toma de decisiones.

2.2 Estimación de la Prueba

- TM-2.2.1 (K2)** Explicar los factores que se necesitan tener en cuenta en la estimación de la prueba.
- TM-2.2.2 (K2)** Dar ejemplos de factores que pueden influir en la estimación de la prueba.
- TM-2.2.3 (K4)** Seleccionar una técnica o enfoque de estimación de la prueba apropiado para un contexto determinado.

2.3 Gestión de Defectos

- TM-2.3.1 (K3)** Implementar un proceso de gestión de defectos, incluyendo el flujo de trabajo de defectos, que se pueda utilizar para monitorizar y controlar los defectos.
- TM-2.3.2 (K2)** Explicar el proceso y los participantes necesarios para una gestión de defectos eficaz.
- TM-2.3.3 (K2)** Explicar los detalles de la gestión de defectos en el desarrollo ágil de software.
- TM-2.3.4 (K2)** Explicar los retos de la gestión de defectos en el desarrollo de software híbrido.
- TM-2.3.5 (K3)** Utilizar los datos y la información de clasificación que se deben recopilar durante la gestión de defectos.
- TM-2.3.6 (K2)** Explicar cómo se pueden utilizar las estadísticas del informe de defecto para idear la mejora del proceso.

2.1 Métricas de Prueba

Introducción - ¿Por qué Disponer de Métricas de Prueba?

Hay un dicho en gestión que dice: “Lo que se puede medir, se hace”. Del mismo modo, lo que no se mide no es probable que se haga porque es fácil de ignorar. Por lo tanto, es importante establecer un conjunto adecuado de métricas para cualquier empresa, incluidas la prueba.

Los objetivos de prueba son la respuesta a por qué probamos (véase la sección 1.4, La estrategia de prueba del proyecto). Para determinar si se han cumplido los objetivos de prueba, hay que definir una forma de medirlos. Las métricas de prueba son los indicadores que nos ayudan a responder a esta pregunta.

Las métricas de prueba pueden clasificarse de la siguiente manera:

- Las **métricas de proyecto** miden el avance respecto a los criterios de salida del proyecto en curso, como el porcentaje de pruebas ejecutadas, pasadas y falladas.
- Las **métricas de producto** miden atributos del producto como el grado en que el producto cumple las expectativas de calidad de los usuarios previstos.
- Las **métricas de proceso** miden la capacidad del proceso de prueba y la efectividad de la prueba. Las métricas del proceso se utilizan, por tanto, para informar sobre la efectividad y la eficiencia del proceso.

Para más información sobre la gestión de métricas de productos y procesos, consultar el programa de estudio ISTQB® Expert Test Management.

Para más información sobre el uso de métricas de proceso, consulte el programa de estudio ISTQB® Expert Improving the Test Process.

En las siguientes secciones se tratarán las métricas para la planificación de la prueba, la monitorización de la prueba, el control de la prueba y la compleción de la prueba. Estas son las cuatro principales actividades de gestión relacionadas con las métricas.

2.1.1 Métricas para las Actividades de Gestión de la Prueba

El programa de estudio de nivel avanzado de gestión de la prueba se concentra en las siguientes actividades genéricas de gestión de la prueba:

- Planificación de la prueba
- Monitorización de la prueba
- Compleción de la prueba (véase el apartado 1.1, El proceso de prueba).

La gestión de la prueba debe ser capaz de definir un conjunto de métricas para la monitorización de la prueba, el control de la prueba y la compleción de la prueba como parte de las actividades de planificación de la prueba. Cada métrica debe definirse, medirse, monitorizarse e informar.

Durante la planificación de la prueba, se definen las métricas de prueba apropiadas que coinciden con los objetivos de prueba de la estrategia de prueba del proyecto.

Las métricas utilizadas durante la monitorización de la prueba y el control de la prueba pueden ser diferentes de las utilizadas en la compleción de la prueba. Durante la monitorización de la prueba y el control de la prueba, las métricas se refieren al avance de las actividades de prueba. En la compleción de la prueba, deben alcanzarse los objetivos de la prueba. Podrían combinarse una o varias métricas para medir los criterios de salida de la prueba asignados a determinados objetivos de prueba.

La siguiente tabla ofrece ejemplos de métricas (hay muchas más) utilizadas en las actividades de gestión de la prueba:

Métrica (planificada/monitorizada para hitos definidos)	Monitorización de la prueba y control de prueba	Compleción de la prueba
Cobertura de requisitos	X	X
Cobertura del riesgo de producto	X	X
Cobertura de código	X	
Estimación real frente a la prevista (en horas) para las actividades de prueba	X	
Porcentaje de casos de prueba ejecutados por estado (por ejemplo, fallidos, bloqueados) frente a los casos de prueba planificados	X	X
Número acumulado de defectos resueltos frente al número acumulado de defectos	X	
Casos de prueba automatizados reales frente a casos de prueba automatizados planificados		X
Coste real frente al coste planificado de la prueba	X	

Tabla 2 - Ejemplos de métricas utilizadas en las actividades de gestión de la prueba

La métrica utilizada en una actividad de prueba específica se muestra en la tabla. Las métricas con una X en monitorización de la prueba y control de la prueba se utilizan principalmente para medir el avance y se informan en los informes de progreso de la prueba (CTFL). Las métricas con una X en completación de la prueba se utilizan principalmente para medir la consecución de los objetivos de la prueba y se informan en el informe de completación de la prueba (CTFL). Las métricas con una X en ambas columnas pueden utilizarse para cualquiera de las dos cosas.

También existen métricas para monitorizar la efectividad de la prueba (por ejemplo, el Porcentaje de Detección de Defectos (PDD)).

El DDP se trata en el programa de estudio ISTQB® Expert Level, módulo mejora del proceso de prueba, concretamente en la sección Implementing Test Process Improvement (Implementación de la mejora del proceso de prueba).

2.1.2 Monitorización, Control y Compleción

Las métricas de prueba son indicadores que muestran hasta qué punto ha avanzado la prueba y si se han alcanzado los criterios de salida o las tareas de prueba relacionadas.

La monitorización de la prueba es la actividad de recopilación de datos relativos a la prueba y a la evaluación asociada. Se utiliza para evaluar el avance de la prueba y comprobar el cumplimiento de los

criterios de salida o las actividades de prueba asociadas (véase el apartado 1.1.2, Monitorización y control de la prueba). Los criterios de salida se obtienen a partir de los objetivos de prueba.

El control de la prueba utiliza la información de la monitorización de la prueba para proporcionar directrices y acciones correctivas con el fin de lograr una prueba eficaz y eficiente. Entre los ejemplos de directivas de control de pruebas se incluyen el cambio de prioridad de las pruebas cuando un riesgo identificado se convierte en un problema, la reevaluación de si un elemento de prueba cumple los criterios de entrada o los criterios de salida debido a la repetición de pruebas, el ajuste del calendario de pruebas para tener en cuenta un retraso en la entrega del entorno de prueba y la adición de nuevos recursos cuando y donde sea necesario.

La compleción de la prueba recopila datos de las actividades de prueba completadas para consolidar las lecciones aprendidas, el producto de prueba y otra información relevante. La compleción de la prueba se produce en hitos del proyecto como la compleción de un nivel de prueba, la compleción de una iteración, la compleción (o cancelación) de un proyecto de prueba, la entrega de un producto o la compleción de una entrega de mantenimiento.

Entre las métricas de prueba habituales que se utilizan en las actividades de gestión de pruebas se incluyen las métricas de avance del proyecto y las métricas que muestran el avance con respecto al calendario y el presupuesto previstos, la calidad actual del elemento de prueba y la efectividad de la prueba con respecto a los objetivos de la prueba o los objetivos de la iteración.

2.1.3 Suministro de información de prueba

La gestión de la prueba debe comprender cómo interpretar y utilizar las métricas para comprender e informar del estado de la prueba. Para los niveles de prueba superiores, como la prueba de sistema, la prueba de integración de sistema, la prueba de aceptación y la prueba de seguridad, la base de prueba principal suelen ser los productos de trabajo, como las especificaciones de requisitos, los casos de uso, las historias de usuario y los riesgos de producto. Las métricas de cobertura estructural son más aplicables a los niveles inferiores de las pruebas, como la prueba de componentes y la prueba de integración de componentes. Aunque la gestión de la prueba puede utilizar métricas de cobertura de código para medir el grado en que sus pruebas ejercitan la estructura del sistema sometido a prueba, el suministro de información sobre los resultados de la prueba de nivel superior debe adaptarse al contexto y a las necesidades específicas del proyecto. Por ejemplo, en entornos que cambian con frecuencia, la métrica de cobertura de código puede ser útil para monitorizar el impacto de los cambios de código en el juego de pruebas e identificar posibles vacíos o riesgos. Además, la gestión de la prueba debe entender que, incluso si las pruebas de componentes y las pruebas de integración de componentes logran el 100% de su cobertura estructural, los defectos y los riesgos de calidad deben abordarse en niveles de prueba superiores.

El objetivo de informar métricas es proporcionar una comprensión inmediata de la información para fines de gestión. Las métricas pueden informarse como una instantánea de una métrica en un momento dado o como la evolución de una métrica a lo largo del tiempo para evaluar tendencias.

Los riesgos de producto, los defectos, el avance de la prueba, la cobertura y el coste y el esfuerzo de prueba relacionados se miden y se comunican de forma específica al final del proyecto.

A continuación, se presentan ejemplos de métricas que pueden utilizarse con distintos fines:

- **Métricas relacionadas con los riesgos de producto** (incluyen entre otras):
 - Porcentaje de riesgos de los que todas las pruebas han pasado.
 - Porcentaje de riesgos de los que fallaron algunas o todas las pruebas.
 - Porcentaje de riesgos que aún no se han probado en su totalidad.

Estas métricas pueden utilizarse para evaluar la calidad de la base de la prueba y la efectividad de los casos de prueba para cubrir los riesgos del producto.

- **Métricas relacionadas con los defectos** (incluyen entre otras):

- Número acumulado de defectos resueltos frente al número acumulado de defectos
- Desglose del número o porcentaje de defectos categorizados por:
 - Elementos de prueba o componentes.
 - Origen del defecto (por ejemplo, especificación, nueva prestación o regresión)
 - Entregas de prueba
 - Nivel de prueba o iteración introducida, detectada y eliminada
 - Prioridad/severidad
 - Causa raíz
 - Estado (por ejemplo, rechazado, duplicado, abierto, cerrado)

Estas métricas pueden utilizarse para monitorizar el proceso de detección y resolución de defectos, identificar las áreas de alta densidad de defectos o severidad de defectos, y evaluar la efectividad y eficacia de la prueba.

- **Métricas relacionadas con el avance de la prueba** (incluyen entre otras):
 - Estado de ejecución de pruebas: Número total de pruebas planificadas, implementadas, ejecutadas, pasadas, fallidas, bloqueadas y omitidas.
 - Esfuerzo de prueba: Número de horas de recursos reales frente a las planificadas dedicadas a probar.
- **Métricas relacionadas con la cobertura** (incluyen entre otras):
 - Cobertura de los requisitos: El porcentaje de requisitos cubiertos por los casos de prueba.
 - Cobertura del riesgo de producto: El porcentaje de riesgos de producto identificados que se ven mitigados por los casos de prueba.
 - Cobertura de código: El porcentaje de sentencias de código, ramas, rutas o condiciones que son ejecutadas por los casos de prueba.
- **Métricas relacionadas con los costes y el esfuerzo de prueba** (incluyen entre otras):
 - Riesgos residuales de los componentes no probados: El impacto potencial y la probabilidad de defectos en los componentes que no se han probado.
 - Coste de la prueba: El coste real frente al previsto de la prueba.

Adicionalmente, resulta útil combinar métricas de diferentes categorías (por ejemplo, una métrica que muestre la correlación entre las tendencias de los defectos abiertos frente a las tendencias de las pruebas ejecutadas, o una métrica que muestre la calidad de la base de prueba basada en el número de defectos encontrados en los requisitos). Cuando la ejecución de pruebas continúa y cada vez se identifican menos defectos, se puede tomar la decisión de finalizar las pruebas. Esta decisión debe basarse en el suministro de información de las métricas y en los criterios de salida acordados.

2.2 Estimación de la Prueba

Introducción

Existen buenas prácticas de gestión de proyectos para la estimación en ingeniería de sistemas y software, en relación con todo tipo de recursos (por ejemplo, el coste, las personas o el tiempo). La estimación de la prueba es la aplicación de estas buenas prácticas a la prueba asociada a un proyecto u operación.

2.2.1 Estimación de las Actividades que Implicará la Prueba

La estimación de la prueba es una actividad de gestión de la prueba que estima cuánto tiempo, esfuerzo y coste requerirá completar una tarea. La estimación de la prueba es una de las tareas principales e importantes de la gestión de la prueba.

Las principales características de la estimación en la gestión de la prueba son:

- **Esfuerzo**
 - El esfuerzo suele calcularse en horas-persona o puntos-historia necesarios para terminar las tareas de prueba del proyecto. A menudo, el esfuerzo de la prueba y la duración de la prueba (tiempo transcurrido) pueden ser diferentes, por lo que la gestión de la prueba puede necesitar estimar la duración total de la actividad. ¿Cuántas horas-persona se necesitarán?
- **Tiempo**
 - Tiempo necesario para finalizar el proyecto. El tiempo es un recurso crítico en un proyecto. La planificación de la prueba necesita estimar el esfuerzo de la prueba en días naturales y en días laborables. Todo proyecto tiene hitos y un plazo de entrega. ¿Cuánto tiempo se tardará en terminar el proyecto de pruebas?
- **Coste**
 - El coste es el presupuesto del proyecto. Incluye los gastos de los recursos, las herramientas y la infraestructura de prueba. ¿Cuánto costará el proyecto de prueba?

La prueba suele ser un subproyecto dentro de un proyecto (extenso), a veces distribuido en varias ubicaciones de prueba (por ejemplo, centros de prueba). Para realizar la estimación de la prueba, el primer paso es identificar los niveles de prueba, las actividades de prueba y las tareas de prueba. A continuación, hay que dividir el proyecto de prueba en sus principales actividades de prueba (por ejemplo, la planificación de la prueba y la ejecución de la prueba) dentro del proceso de prueba (véase el programa de estudio de nivel básico de ISTQB®). En los proyectos ágiles, las actividades de prueba suelen estimarse dentro del trabajo de desarrollo, y no como valores separados. El siguiente paso consiste en estimar el esfuerzo de prueba necesario para finalizar las tareas o los productos de trabajo y cuáles son los costes esperados de ello.

Como la prueba es una subactividad de un proyecto, siempre hay algunas restricciones naturales del proyecto que influyen en ella y requieren un compromiso, y no se pueden manipular estos valores de forma arbitraria. Eso se puede encontrar en la gestión de la calidad como el triángulo tiempo-coste-calidad. En la gestión de proyectos, el triángulo tiempo-coste-calidad comprende tres valores que son interdependientes, lo que significa que están estrechamente relacionados y se influyen mutuamente. Esta relación se observa comúnmente en los escenarios de proyectos.

2.2.2 Factores que Pueden Influir en el Esfuerzo de Prueba

La estimación del esfuerzo de prueba consiste en predecir la cantidad de trabajo relacionado con las actividades de prueba que se necesitará para cumplir los objetivos de prueba de un determinado proyecto, entrega o iteración. Entre los factores que pueden influir en el esfuerzo de prueba se incluyen las características de:

- **Producto:**

- La calidad de la base de prueba.
- El tamaño del producto a probar (es decir, el objeto de prueba)
- La complejidad del dominio del producto (por ejemplo, el entorno, la infraestructura y la historia)
- Los requisitos para probar las características de calidad (por ejemplo, seguridad y fiabilidad).

Estos factores relacionados con el producto pueden influir en la estimación de la prueba porque crean un contexto específico para el sistema a prueba.

- **Proceso de Desarrollo:**

- La estabilidad y madurez de los procesos de desarrollo de la organización.
- El modelo de desarrollo (por ejemplo, desarrollo ágil de software/iterativo, o modelo híbrido de desarrollo de software) en uso
- Los factores materiales (por ejemplo, la disponibilidad de automatización de la prueba, herramientas y entornos de prueba).

Estos factores relacionados con el proceso de desarrollo pueden influir en la estimación de la prueba porque ésta está directamente relacionada con el desarrollo.

- **Personas:**

- La satisfacción de las personas (por ejemplo, la que proporcionan los días festivos, las vacaciones y otros beneficios previstos).
- Las competencias y la experiencia de las personas involucradas, especialmente en lo que respecta a proyectos y productos similares (por ejemplo, conocimiento del dominio).

Las personas son los recursos más necesarios, por lo que debe tenerse en cuenta cualquier inestabilidad. Por lo tanto, las personas son un factor importante en la estimación del esfuerzo de prueba. Véase también la sección 3.1 de este programa de estudio, El equipo de prueba.

- **Resultados de prueba:**

- El número y la severidad de los defectos encontrados durante la ejecución de prueba.
- La cantidad de retrabajo necesario

Las estadísticas históricas apoyan la estimación de la prueba. Por tanto, conocer estos factores apoyará una estimación con valores más precisos.

- **Contexto de la prueba:**

- La distribución de las pruebas en varias filiales, la composición y ubicación de los equipos, la complejidad del proyecto (por ejemplo, múltiples subsistemas).
- El tipo de trabajo (por ejemplo, virtual o in situ).

Los factores relacionados con el contexto son los que afectan a toda la estimación de la prueba. Véase también la sección 1.2. de este programa de estudio, El contexto de la prueba.

2.2.3 Selección de Técnicas de Estimación de Prueba

La estimación de la prueba debe abarcar todas las actividades que intervienen en el proceso de prueba. La estimación del coste, el esfuerzo y, sobre todo, la duración de la ejecución de la prueba suele ser lo más importante para la gestión de la prueba porque estos valores repercutirán en el proyecto. Sin embargo,

las estimaciones de la ejecución de la prueba tienden a ser difíciles de generar cuando la calidad del software en general es baja o desconocida. Además, es probable que la familiaridad y la experiencia con el producto afecten a la calidad de las estimaciones. Una práctica habitual es estimar el número de casos de prueba derivados de la base de prueba (por ejemplo, los requisitos o las historias de usuario). Las suposiciones realizadas durante la estimación de la prueba deben documentarse siempre como parte de la estimación.

Las técnicas o enfoques de estimación de la prueba pueden clasificarse como basadas en métricas o en expertos.

Para más detalles sobre las técnicas de estimación de la prueba, consulte el programa de estudio de nivel básico V.4 del ISTQB®.

En la mayoría de los casos, la estimación, una vez preparada, debe entregarse a la dirección del proyecto, junto con una justificación. Con frecuencia, algunos parámetros de entrada se modifican (por ejemplo, el alcance de la prueba), lo que suele dar lugar a un ajuste de la estimación. Lo ideal es que la estimación final de la prueba represente el mejor equilibrio posible entre los objetivos de la organización y los del proyecto en los ámbitos de la calidad, el calendario, el presupuesto y las prestaciones.

Es importante tener en cuenta que cualquier estimación se basa en la información disponible en el momento en que se elabora. Al principio de un proyecto, la información puede ser bastante limitada. Además, esta información puede cambiar con el tiempo. Para seguir siendo precisas, las estimaciones deben actualizarse para reflejar la información nueva y cambiante.

La selección de la técnica de estimación depende de varios factores, por ejemplo:

- **Error de estimación:** Algunas técnicas permiten calcular la desviación estándar, que es una medida de la incertidumbre o variabilidad de la estimación. Por ejemplo, la técnica de estimación de tres puntos utiliza las estimaciones optimista, pesimista y más probable para calcular el valor esperado y la desviación estándar de la estimación (véase el programa de estudio de nivel básico ISTQB® V.4, apartado 5.1.4 para más detalles).
- **Disponibilidad de datos:** Algunas técnicas requieren datos históricos de proyectos anteriores o similares que pueden no estar disponibles o no ser fiables. Por ejemplo, la estimación basada en proporciones y la extrapolación se basan en datos históricos para obtener las proporciones o las tendencias del proyecto actual.
- **Disponibilidad de expertos:** Algunas técnicas requieren la participación de expertos que tengan los conocimientos y la experiencia necesarios para proporcionar estimaciones precisas y realistas. Por ejemplo, el método Delphi y el póker de planificación se basan en las opiniones y juicios de expertos o miembros del equipo.
- **Conocimientos en modelado:** Algunas técnicas requieren el uso de modelos matemáticos o fórmulas para calcular las estimaciones, lo que puede exigir ciertas competencias y conocimientos en modelado. Por ejemplo, la extrapolación y la estimación de tres puntos utilizan fórmulas para obtener el valor esperado y la desviación estándar de la estimación.
- **Restricciones temporales:** Algunas técnicas requieren más tiempo y esfuerzo para su realización que otras, lo que puede afectar a la viabilidad y adecuación de la técnica. Por ejemplo, el póker de planificación es fácil de realizar, mientras que la extrapolación puede ser más difícil.

Esto demuestra que los criterios de selección de las técnicas adecuadas de estimación de la prueba dependen en gran medida del contexto de prueba (por ejemplo, el ciclo de vida de desarrollo de software (CVDS), los implicados, los niveles de prueba y los tipos de prueba utilizados en el proyecto) (véase la sección 1.2 de este programa de estudio, El contexto de la prueba). El jefe de prueba debe ser capaz de coordinar y aplicar las técnicas de estimación de la prueba, (por ejemplo, con diferentes modelos de ciclo de vida de desarrollo de software (CVDS) en un proyecto sobre diferentes entidades subsidiarias).

Por ejemplo, para seleccionar la técnica de estimación adecuada, determine primero la complejidad del tema. Si la complejidad es baja, podrían utilizarse técnicas basadas en métricas. Si la complejidad es alta, entonces podrían utilizarse técnicas basadas en expertos. Si se utiliza un modelo de desarrollo secuencial,

entonces podría utilizarse la técnica de estimación Delphi de banda ancha. Si se utiliza un modelo de desarrollo ágil de software, podría emplearse el póker de planificación.

BORRADOR

2.3 Gestión de Defectos

Introducción

El programa de estudio de nivel básico V.4 de ISTQB® describe las actividades que se inician tras observar resultados reales que difieren de los resultados esperados. El programa de estudio se refiere a estas actividades como gestión de defectos. Otros estándares utilizan el término “gestión de incidencias” (ISO/IEC/IEEE 29119-3 Standard o “gestión de anomalías” (TMAP) para enfatizar el hecho de que al principio del proceso puede que no sepamos si la discrepancia está causada por un defecto en un producto de trabajo, o se debe a algo más (por ejemplo, un fallo de automatización de la prueba o una mala comprensión de los requisitos por parte del probador). La gestión de defectos y la herramienta utilizada para gestionarlos son de vital importancia para los probadores y para los demás miembros del equipo que participan en el desarrollo del software. La información de un proceso eficaz de gestión de defectos permite al equipo de prueba y a otros implicados en el proyecto conocer el estado de un proyecto a lo largo de su ciclo de vida de desarrollo de software (CVDS). La gestión de defectos también es crucial para decidir qué defectos se corregirán. Esto asegura que el esfuerzo se invierte en trabajar con los defectos correctos. Recopilar y analizar los datos relacionados con los defectos a lo largo del tiempo puede ayudar a localizar áreas de mejora potencial tanto para las pruebas como para otros procesos dentro del ciclo de vida de desarrollo de software (SDLC) (por ejemplo, una mejor prevención de los defectos mediante la mejora de la arquitectura y el diseño técnico).

Además de comprender el ciclo de vida general de los defectos y cómo se utiliza para supervisar y controlar tanto el desarrollo de software como los procesos de prueba, el jefe de prueba y los probadores (o todo el equipo Ágil en el desarrollo ágil de software) también deben estar familiarizados con qué datos es fundamental recopilar. El jefe de prueba debe ser un defensor del uso adecuado tanto del proceso de gestión de defectos como de la herramienta de gestión de defectos seleccionada.

2.3.1 Ciclo de Vida de los Defectos

Cada fase del ciclo de vida de desarrollo de software (CVDS) debe incluir actividades para detectar y eliminar defectos potenciales. Por ejemplo, las técnicas de prueba estática (es decir, revisiones y análisis estático) pueden utilizarse en las especificaciones de diseño, las especificaciones de requisitos y el código antes de entregar esos productos de trabajo a las actividades posteriores. Cuanto antes se detecte y elimine cada defecto, menor será el coste global de la calidad del producto. El coste de la calidad se minimiza cuando cada defecto se elimina dentro de la misma fase en la que se introdujo (es decir, cuando el proceso del software logra una contención de fase perfecta).

Durante la prueba estática buscamos defectos. Durante las pruebas dinámicas, la presencia de un defecto se revela cuando provoca un fallo, lo que se traduce en una discrepancia entre los resultados reales y los resultados esperados de una prueba (es decir, una anomalía). En algunos casos, se produce un resultado de falso negativo cuando el probador no observa la anomalía. Cuando se observa una anomalía, debe realizarse una investigación más profunda. Esta investigación suele comenzar rellenando un informe de defecto de acuerdo con el proceso de prueba y gestión de defectos definido. Una prueba fallida no siempre da lugar a la creación de un informe de defecto. (Por ejemplo, en el desarrollo guiado por pruebas, donde las pruebas de componente, normalmente automatizadas, se utilizan como una forma de especificación de diseño ejecutable). Hasta que el desarrollo del componente esté completo, algunas o todas las pruebas deben fallar inicialmente. Hasta que el desarrollo del componente esté completo, algunas o todas las pruebas deben fallar inicialmente. Por lo tanto, el resultado de una prueba de este tipo no está necesariamente causado por un defecto y no suele ser objeto de seguimiento a través de un informe de defecto.

Un informe de defectos avanza a lo largo de un flujo de trabajo (por simplicidad y consistencia con la mayoría de las herramientas de gestión de defectos seguiremos utilizando el término “flujo de trabajo de defectos”) y se mueve a través de una secuencia de estados de defectos. En la mayoría de estos estados, una persona es la propietaria del informe de defecto y la responsable de llevar a cabo una tarea (por

ejemplo, análisis, eliminación de defectos o prueba de confirmación). El siguiente diagrama representa un flujo de trabajo de defectos sencillo:

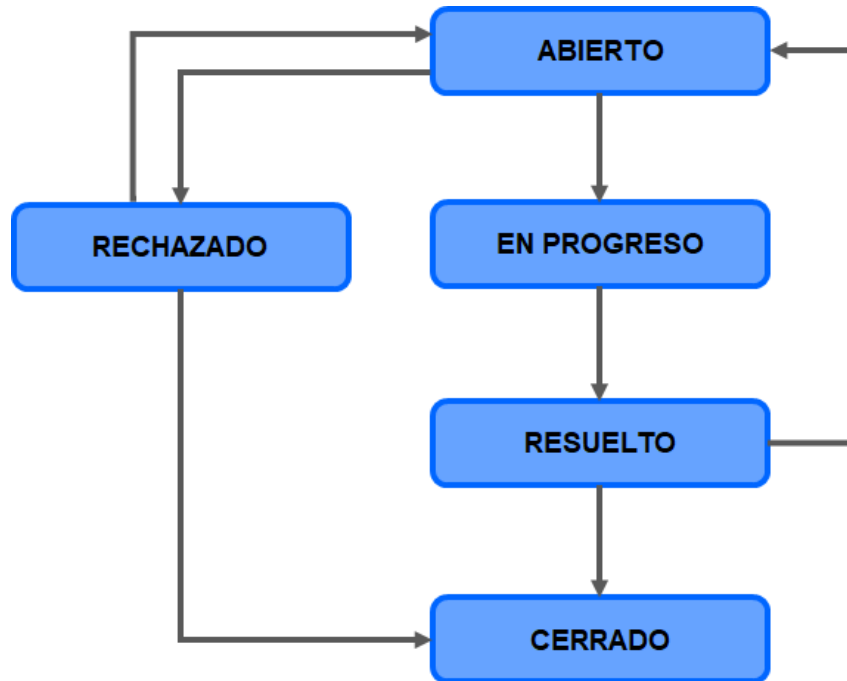


Imagen 2 - Flujo de trabajo simple de un defecto

Un flujo de trabajo simple de defectos puede abarcar los siguientes estados de defectos:

- **ABIERTO** (puede denominarse NUEVO): El estado inicial cuando se crea el informe de defecto.
- **EN PROGRESO**: El equipo está trabajando en el análisis del informe de defecto y/o en el arreglo correspondiente.
- **RECHAZADO**: Un informe de defecto es rechazado por la persona que lo ha procesado (normalmente un desarrollador o un analista). Puede haber muchas razones para el rechazo (por ejemplo, información no válida, prueba incorrecta, informe de defecto duplicado) y esta información se añade al informe de defecto.
- **RESUELTO** (puede denominarse ARREGLADO, LISTO PARA LA REPETICIÓN DE PRUEBA): Un probador ejecuta una prueba de confirmación siguiendo a menudo los pasos para reproducir el fallo a partir del propio informe de defectos para determinar si la corrección ha resuelto realmente el defecto.
- **CERRADO**: El informe de defecto ha alcanzado su estado terminal y no se pretende realizar más trabajo. El probador pasa el informe de defecto a este estado después de una prueba de confirmación satisfactoria o para reconocer el rechazo del informe de defecto.

En muchas organizaciones se utiliza un flujo de trabajo simple para los defectos, que se amplía con el uso de otros estados de defecto relevantes para un contexto determinado (por ejemplo, REABIERTO, ACEPTADO, CLARIFICACIÓN o APLAZADO).

El flujo de trabajo de defectos puede variar en las distintas organizaciones en cuanto a los distintos nombres de los estados de defectos, las reglas para las transiciones entre estados de defectos y los roles responsables de las tareas en determinados estados de defectos. A menudo, el flujo de trabajo de defectos es más sencillo en el desarrollo ágil de software que en los modelos de desarrollo secuencial. El flujo de

trabajo de defectos debe adaptarse a un contexto determinado. Al diseñar el flujo de trabajo de defectos, es aconsejable respetar una serie de buenas prácticas:

- Si es posible, el flujo de trabajo de defectos debe definirse a nivel de toda la organización para proporcionar una gestión de defectos unificada en todos los proyectos.
- Los defectos duplicados y los falsos positivos deben representarse mediante un estado separado o una combinación del estado RECHAZADO con la elección del motivo del rechazo. Pueden ser útiles en posteriores análisis de defectos con el objetivo de mejorar el proceso de prueba.
- Se recomienda utilizar un único estado terminal (por ejemplo, CERRADO). La transición a este estado suele requerir la elección de un motivo de cierre, útil para la evaluación de proceso y las actividades de mejora del proceso.
- Los nombres de los estados en el flujo de trabajo de defectos deben ser los mismos que los de los estados análogos utilizados para otras entidades (por ejemplo, historias de usuario y tareas de prueba) para simplificar el trabajo con ellos.
- Los estados de defecto consecutivos deben pertenecer a diferentes roles responsables. Si dos o más estados consecutivos pertenecen al mismo rol responsable, debe haber una buena razón (por ejemplo, para medir el tiempo pasado en un estado de defecto).
- Cada estado de defecto, excepto el estado terminal, debe tener más de una transición de salida para permitir al rol responsable una decisión respecto al siguiente paso. Las excepciones a esta regla deben estar justificadas (por ejemplo, para monitorizar el tiempo empleado en una actividad determinada).
- El conjunto de atributos que deben introducirse al realizar una transición de estado debe limitarse a aquellos que aporten un valor sustancial a la gestión de defectos.

2.3.2 Gestión de defectos transfuncional

Aunque la organización de prueba y el jefe de prueba suelen ser los propietarios del proceso general de gestión de defectos y de la herramienta de gestión de defectos, un equipo interfuncional suele ser el responsable de gestionar los defectos de un proyecto determinado. Este equipo, a veces denominado comité de gestión de defectos, puede incluir al jefe de prueba, representantes de desarrollo, proveedores, gestión de proyectos, gestión del producto o propietario del producto y otros implicados que tengan un interés en el software sujeto a prueba.

A medida que se descubren las anomalías y se introducen en la herramienta de gestión de defectos, el comité de gestión de defectos debe determinar si cada informe de defectos representa un defecto válido y si debe ser corregido (y por qué parte en caso de que varios equipos de desarrollo participen en la entrega), rechazado o aplazado. Esta decisión requiere que el comité de gestión de defectos tenga en cuenta los beneficios, riesgos y costes asociados a la reparación del defecto. Resulta beneficioso debatir esta consideración en una reunión (a menudo denominada reunión de clasificación). Si se va a arreglar el defecto, el equipo debe establecer la prioridad de arreglar el defecto en relación con otras tareas. El jefe de prueba y el equipo de prueba pueden ser consultados sobre la importancia relativa de un defecto y deben proporcionar la información objetiva disponible.

En proyectos muy grandes, el nombramiento de un jefe de defectos a tiempo completo puede estar justificado por el esfuerzo necesario para preparar y hacer un seguimiento de las decisiones tomadas en las reuniones del comité de gestión de defectos, al menos durante las fases del ciclo de vida de desarrollo de software (CVDS) en las que la prueba es más intensiva. En otras situaciones, varios proyectos grandes pueden compartir un jefe de defectos.

Una herramienta de gestión de defectos no debe utilizarse como sustituto de una buena comunicación, ni un comité de gestión de defectos como sustituto del uso eficaz de una buena herramienta de gestión de defectos. La comunicación, un soporte adecuado de la herramienta, un flujo de trabajo de defectos bien

definido (que incluya las propiedades del informe de defectos) y un equipo de gestión de defectos comprometido son necesarios para una gestión de defectos eficaz y eficiente.

2.3.3 Particularidades de la Gestión de Defectos en Equipos Ágiles

La gestión de defectos en las organizaciones que utilizan el Desarrollo Ágil de Software suele ser ligera y/o menos formal que en los modelos de desarrollo secuencial. Si los equipos ágiles están en ubicación común o disponen de medios de comunicación bien establecidos, la información sobre un defecto o fallo suele intercambiarse entre probadores, representantes del cliente y desarrolladores sin necesidad de un informe de defecto formal. Sin embargo, deberían crearse informes de defecto en los siguientes casos:

- Defectos que bloqueen otras actividades del sprint en curso (es decir, desarrollo, pruebas u otras) y que no puedan solucionarse inmediatamente dentro del equipo ágil.
- Defectos que no puedan resolverse dentro de la misma iteración. Algunos equipos ágiles tienen la norma de crear un informe de defecto si éste no puede resolverse durante el día en que se encuentra el fallo.
- Defectos que deben ser resueltos por o en cooperación con otros equipos en organizaciones multiequipo.
- Defectos que deben ser resueltos por un proveedor.
- Defectos para los que se solicita explícitamente un informe de defecto (por ejemplo, cuando un desarrollador no puede trabajar inmediatamente en su solución).

La práctica habitual consiste en añadir los defectos que no pueden resolverse dentro de la misma iteración a la lista de trabajo acumulado del producto, de modo que puedan priorizarse entre otros defectos e historias de usuario para una iteración posterior.

Aunque las bases de la gestión de defectos deben establecerse en la estrategia de prueba de una organización, muchos aspectos, como el nivel de formalidad, los desencadenantes de la creación de un informe de defectos y los atributos de los defectos que deben capturarse, pueden dejarse al acuerdo entre los miembros del equipo ágil. En general, el nivel de formalidad de la gestión de defectos y el enfoque para crear informes de defectos deben reflejar lo siguiente:

- Ubicación común de los miembros del equipo.
- La distribución de los miembros del equipo por zonas horarias.
- El número de equipos que cooperan en el desarrollo del producto.
- Madurez del equipo o equipos.
- Tamaño del equipo o equipos.
- Riesgos asociados al producto.
- Requisitos legales, contractuales o de otro tipo (si son aplicables).

La decisión final del equipo ágil sobre los detalles de la gestión de defectos debe documentarse siempre (por ejemplo, con directrices en una herramienta de gestión de conocimientos).

2.3.4 Retos de la Gestión de Defectos en el Desarrollo Híbrido de Software

En la práctica, es habitual que varios equipos colaboren en la entrega del sistema o sistema de sistemas. Los ejemplos incluyen el desarrollo ágil de software cuando un cliente utiliza el desarrollo ágil de software y uno de sus proveedores utiliza un modelo de desarrollo secuencial, o cuando una organización que utiliza un modelo de desarrollo secuencial requiere la entrega de un subsistema por parte de un equipo que utiliza el desarrollo ágil de software. Un entorno multiequipo de este tipo plantea varios retos:

- El alineamiento sobre los atributos de los defectos y las herramientas a utilizar para la gestión de defectos: En un escenario ideal, todos los equipos utilizan una misma herramienta de gestión de defectos. En la práctica, es habitual que cada equipo utilice una herramienta de gestión de defectos

diferente, especialmente cuando varios equipos de proveedores contribuyen a la entrega del proyecto. En estos casos es bueno establecer una sincronización entre las herramientas de gestión de defectos (preferiblemente de forma automática).

- **Priorización de defectos:** El propietario o propietarios de producto deben participar en las reuniones de gestión de defectos y buscar activamente información sobre las consecuencias y los riesgos asociados a los defectos. Las reuniones de gestión de defectos deben celebrarse con más frecuencia en el desarrollo ágil de software que en los modelos de desarrollo secuencial para seguir el ritmo de entrega de incrementos de producto más rápido del equipo ágil. Sin embargo, estas reuniones pueden ser más breves con equipos ágiles. A veces es beneficioso que un grupo más pequeño de implicados de la gestión de defectos tenga la última palabra sobre la priorización de los defectos.
- **Alineamiento y transparencia del plan de prueba para el nuevo desarrollo y la corrección de defectos:** El trabajo de todos los equipos debe alinearse con el mismo plan de proyecto, independientemente de que utilicen modelos de desarrollo ágil de software o de desarrollo secuencial. Todos los entregables, incluidas las correcciones de defectos, deben alinearse con este plan de proyecto. Se puede lograr una mejor alineación mediante la participación activa de los miembros de todos los equipos en el proceso de planificación (por ejemplo, la participación de los equipos del modelo de desarrollo secuencial en las reuniones de desarrollo ágil de software en las que se discuten y priorizan los defectos). La transparencia de los planes de desarrollo puede mejorarse compartiéndolos entre los equipos (por ejemplo, a través de cuadros de mando o de la lista de tareas pendientes del producto).

2.3.5 Información del Informe de Defecto

La información de un informe de defecto debería ser suficiente para los siguientes fines:

- Gestión del informe de defecto a lo largo del ciclo de vida del defecto.
- Evaluación del estado general del proyecto, especialmente en términos de calidad del producto, y del avance de las pruebas.
- Evaluación del estado de un incremento de producto en términos de calidad del producto
- Evaluación de la capacidad del proceso.

La información necesaria para la gestión de defectos y el estado del proyecto puede variar en función del momento en que se detecte el defecto en el ciclo de vida de desarrollo de software (CVDS). Además, los informes de defecto relacionados con características de calidad no funcionales pueden necesitar más información (por ejemplo, las condiciones de carga para las dificultades de rendimiento). Sin embargo, la información básica recopilada debe ser consistente en todo el ciclo de vida de desarrollo de software (CVDS) e, idealmente, en todos los proyectos de una organización para permitir una comparación significativa de los datos de defectos en todo el proyecto y en todos los proyectos.

En un informe de defecto se pueden recopilar muchos elementos de información. El jefe de prueba debe decidir qué información es la adecuada para una gestión de defectos eficaz en el contexto de un proyecto determinado. Debido al hecho de que cada atributo adicional aumenta el tiempo empleado en el informe de defectos y puede aumentar la confusión de la persona que introduce el informe de defectos, es aconsejable recopilar únicamente los datos que se necesitan para la gestión de defectos en el contexto dado y/o que se utilizarán para la mejora del proceso.

Para gestionar el informe de defecto en la mayoría de los entornos, es obligatorio lo siguiente:

- Un título del defecto con un breve resumen de la anomalía.
- Una descripción detallada de la anomalía que incluya preferiblemente los pasos para reproducir el fallo.
- Severidad del impacto sobre el sistema sujeto a prueba y/o los implicados del producto,
- Prioridad para solucionar la anomalía.

La herramienta de gestión de defectos suele generar otros elementos de datos adicionales importantes:

- Identificador único para el informe de defecto
- Fecha/hora de creación del informe de defecto
- Nombre de la persona que descubrió y/o informó de la anomalía
- Fase del proyecto y del ciclo de vida de desarrollo de software (CVDS) en la que se descubrió la anomalía
- Estado actual del informe de defecto
- Propietario actual (es decir, la persona asignada actualmente para trabajar en el defecto)
- Histórico de cambios, como la secuencia de acciones, incluida la información de fecha/hora, llevadas a cabo por los miembros del equipo del proyecto para aislar, reparar y confirmar el defecto como solucionado
- Referencias (por ejemplo, al caso de prueba, a defectos conectados).

Dependiendo del contexto, también se puede recoger más información (por ejemplo, trazabilidad) en un informe de defecto (para más información, consulte la norma ISO/IEC/IEEE 29119-3). Los siguientes puntos agrupan la información en función de la finalidad prevista:

- **Ayudar a la resolución de defectos:** El subsistema o componente en el que se encuentra el defecto, el elemento de prueba específico y su número de versión en el que se observó la anomalía o el entorno de prueba en el que se observó el defecto.
- **Para evaluar el estado general del proyecto:** Información para monitorizar el avance, (por ejemplo, los riesgos, costes, oportunidades y beneficios asociados a la corrección o no del defecto, una descripción de cualquier solución provisional disponible o los requisitos afectados por los defectos).
- **Para evaluar el estado de un incremento de producto en términos de calidad del producto:** El tipo de defecto (normalmente correspondiente a una taxonomía de defectos), el producto de trabajo en el que se introdujo el defecto o la característica de calidad/subcaracterística afectada por el defecto.
- **Para evaluar la capacidad del proceso:** Información para monitorizar la efectividad y eficiencia de los procesos de desarrollo (por ejemplo, la fase del ciclo de vida de desarrollo de software (CVDS) de introducción, detección y eliminación del defecto o de la causa raíz del defecto).

2.3.6 Definición de las Acciones de Mejora del Proceso Utilizando la Información de los Informes de Defectos

Como se discute en la Sección 2.3.5 de este programa de estudio, Información del Informe de Defecto, los informes de defecto pueden ser útiles para monitorizar el estado del proyecto e informar sobre el mismo. Aunque las implicaciones de las métricas en el proceso de prueba se tratan principalmente en el programa de estudio de gestión de pruebas de expertos, en el nivel de gestión de pruebas avanzado, los jefes de prueba deben ser conscientes de lo que significan los informes de defecto para evaluar la capacidad de los procesos de desarrollo y prueba del software.

Además de la información sobre la monitorización del avance de la prueba que se menciona en este programa de estudio, en la sección 2.1.2, Monitorización, Control y Compleción, y en la sección 2.1.3, Informes de prueba, la información sobre defectos debería apoyar las iniciativas de mejora del proceso que se discuten durante las retrospectivas. Algunos ejemplos son:

- Utilizar la información sobre las fases de introducción, detección y eliminación de defectos para evaluar la contención de fase y/o realizar un análisis del coste de la calidad con el objetivo de sugerir formas de mejorar la efectividad de la detección de defectos en cada fase y minimizar el coste asociado a los defectos.
- Utilizar la información sobre la fase de introducción para analizar las fases en las que se introduce el mayor número de defectos, con el fin de habilitar mejoras específicas para la prevención de defectos.

- Utilizar la información sobre la causa raíz de los defectos para determinar las razones subyacentes de la introducción de defectos, a fin de habilitar mejoras del proceso que reduzcan el número total de defectos
- Utilizar la información sobre la ubicación de los defectos para realizar un análisis de agrupamiento de defectos, comprender mejor los riesgos técnicos (para las pruebas basadas en el riesgo) y habilitar la refactorización de los componentes problemáticos.
- Utilizar información sobre defectos reabiertos para evaluar la calidad de las implementaciones de depuración.
- Utilizar la información sobre defectos duplicados y rechazados para evaluar la calidad de la creación del informe de defecto
- Permitir mejoras del proceso que reduzcan el número total de defectos mediante la introducción de medidas proactivas para evitar errores por adelantado

El uso de métricas para evaluar la efectividad y la eficiencia del proceso de prueba se trata en el programa de estudio de gestión de la prueba.

En algunos casos, los equipos deciden no hacer un seguimiento de los defectos encontrados durante algunas o todas las fases del ciclo de vida de desarrollo de software (CVDS). Aunque esto se hace a menudo apelando a la eficiencia y en aras de reducir la sobrecarga del proceso, reduce enormemente la visibilidad de las capacidades del proceso de desarrollo y prueba del software. Esto hace que las mejoras sugeridas anteriormente sean difíciles de llevar a cabo debido a la falta de datos de apoyo fiables.

3 Gestión de Equipo

Duración: 225 minutos

Palabras Clave⁴

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	evaluación	appraisal
ISTQB	coste de la calidad	cost of quality
ISTQB	defecto	defect
ISTQB	prevención de defectos	defect prevention
ISTQB	fallo externo	external failure
ISTQB	fallo	failure
ISTQB	fallo interno	internal failure

⁴ Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

Objetivos de Aprendizaje para “Capítulo 3”

3.1 El Equipo de Prueba

- TM-3.1.1 (K2)** Dar ejemplos de las competencias típicas que necesitan los miembros del equipo de prueba dentro de cuatro áreas de competencia.
- TM-3.1.2 (K4)** Analizar el contexto de un proyecto determinado para determinar las competencias necesarias para los miembros del equipo de prueba.
- TM-3.1.3 (K2)** Explicar las técnicas típicas de evaluación de competencias para los miembros del equipo de prueba.
- TM-3.1.4 (K2)** Diferenciar entre los enfoques típicos para desarrollar las competencias de los miembros del equipo de pruebas.
- TM-3.1.5 (K2)** Explicar las competencias necesarias para gestionar un equipo de prueba.
- TM-3.1.6 (K2)** Dar ejemplos de factores de motivación e higiene para los miembros de un equipo de prueba.

3.2 Relación con los Implicados

- TM-3.2.1 (K2)** Dar ejemplos para cada una de las cuatro categorías determinantes del coste de la calidad.
- TM-3.2.2 (K3)** Aplicar un cálculo de coste-beneficio para estimar el valor añadido de la prueba para los implicados.

3.1 El Equipo de Prueba

Introducción

Cualquier equipo que realice tareas de prueba está formado por individuos con diferentes competencias. Mientras que en algunas organizaciones los equipos se autoorganizan, en otras los jefes de prueba se encargan de reclutar y de desarrollar estos equipos. La combinación adecuada de habilidades⁵ es un factor crítico para que todos los equipos completen con éxito las tareas de prueba.

Las habilidades que necesita un miembro de un equipo de prueba pueden cambiar con el tiempo. Es importante seleccionar a las personas adecuadas para el equipo de prueba y proporcionarles una formación adecuada y oportunidades de crecimiento. Además, las personas ajenas al equipo de prueba pueden aportar habilidades específicas adicionales.

Esta sección examina el proceso fundamental de análisis y desarrollo de las habilidades necesarias de los miembros del equipo de pruebas, así como las habilidades que se requieren para dirigir o entrenar a un equipo de pruebas. También incluye el conocimiento de los factores que motivan o desmotivan a los miembros del equipo de prueba y otros factores para asegurar el éxito del trabajo en equipo.

Cada individuo posee ya unas habilidades y puede desarrollarlas aún más a través de diversas vías como la experiencia laboral, la educación y la formación. El equipo de prueba ideal tiene todas las habilidades necesarias para las tareas de prueba dadas o sólo es responsable de las tareas de las que tiene las habilidades requeridas. Para tener éxito, un equipo de prueba necesita varias habilidades a distintos niveles. Dependiendo del contexto del proyecto, algunas habilidades serán más importantes o necesarias que otras. Puede tener sentido recurrir a expertos externos para tareas de prueba específicas que estén más allá de las capacidades del equipo de prueba.

3.1.1 Habilidades Habituales dentro de las Cuatro Áreas de Competencia

Las habilidades de una persona pueden clasificarse en cuatro áreas de competencia (Sonntag & Schmidt-Rathjens, 2005) (Erpenbeck & von Rosenstiel, 2017)⁶:

- **Competencia profesional:** Consiste en habilidades para realizar tareas especializadas. Por ejemplo, habilidades en técnicas de prueba, conocimientos tecnológicos y del negocio en el dominio de la aplicación, así como habilidades de gestión de proyectos.
- **Competencia metodológica:** Incluye las habilidades generales que una persona puede utilizar de forma independiente en un dominio y que permiten el desempeño independiente en tareas complejas o novedosas. Algunos ejemplos son las habilidades analíticas, conceptuales y de juicio.
- **Competencia social:** Incluye las habilidades relacionadas con la comunicación, la cooperación y la gestión de conflictos en contextos intra e interculturales. Permiten relacionarse con los demás para actuar adecuadamente en una situación determinada y alcanzar objetivos individuales y compartidos. Algunos ejemplos son las habilidades comunicativas, la capacidad de resolución de conflictos, la habilidad para trabajar en equipo, la adaptabilidad y asertividad.
- **Competencia personal:** Incluye la capacidad y la voluntad de desarrollarse uno mismo y de desarrollar el propio talento, la motivación y la voluntad de rendimiento, así como el desarrollo de actitudes específicas y de una personalidad individual. Algunos ejemplos son la autogestión, la responsabilidad personal, la capacidad de recibir críticas, la fiabilidad, la resiliencia, la capacidad de

⁵ El término «habilidad» se utiliza como término general para referirse a las habilidades en sí, a tener conocimientos sobre algo y a tener la capacidad de hacer algo.

⁶ Las cuatro áreas de competencia utilizadas aquí se basan en el modelo descrito en estas referencias, que es ampliamente utilizado. Existen otros modelos descritos en la bibliografía que agrupan las habilidades de forma diferente. Éstos no forman parte de este programa de estudio.

actuar con confianza, la disciplina, la apertura a los cambios, la voluntad de ayudar y de aprender y la capacidad de delegar.

Todas las áreas de competencia son importantes para el éxito de cualquier equipo de prueba. Como la competencia metodológica, social y personal no es específica de la prueba, el ISTQB® se concentra en el desarrollo de la competencia profesional. Esto incluye habilidades para gestionar tareas de prueba, analizar la base de prueba, diseñar pruebas, identificar y analizar riesgos y desarrollar, configurar y mantener datos de prueba, entornos de prueba y guiones de prueba.

3.1.2 Analizar las Habilidades Requeridas de los Miembros del Equipo de Prueba

La dotación de personal es una actividad dentro de la planificación de la prueba. Incluye la tarea de identificar los roles y habilidades del personal necesario para la implementación de prueba en la estrategia de prueba. Se requiere un análisis detallado del contexto para determinar las habilidades necesarias para un proyecto.

Competencia profesional y metodológica

En el caso de la prueba, la atención se centra en las habilidades de prueba necesarias para las tareas de prueba. A continuación, se presentan algunos ejemplos:

- La planificación de la prueba requiere conocimientos conceptuales para desarrollar una estrategia de prueba.
- La monitorización de la prueba y el control de la prueba requieren habilidades de gestión de proyectos para gestionar todas las tareas de prueba.
- El análisis de prueba requiere habilidades analíticas para analizar la base de prueba y los riesgos de producto.
- El diseño de pruebas requiere habilidades en técnicas de diseño de casos de prueba y conocimientos conceptuales para diseñar los entornos de prueba.
- La implementación de las pruebas requiere habilidades de juicio para la selección de pruebas y conocimientos técnicos para la programación de guiones de prueba y la configuración de entornos de prueba.
- La ejecución de pruebas requiere conocimientos técnicos para ejecutar pruebas automatizadas, realizar pruebas exploratorias y evaluar los resultados de las pruebas.
- La compleción de la prueba requiere la capacidad de comunicar los resultados del proyecto y la responsabilidad personal por las decisiones tomadas.

Los distintos tipos de pruebas y niveles de pruebas requieren habilidades diferentes (por ejemplo, conocimientos del negocio en el ámbito de la aplicación para evaluar la adecuación funcional de un sistema, o conocimientos técnicos para evaluar la mantenibilidad del código).

Además, el contexto del proyecto proporciona información valiosa sobre la competencia profesional necesaria:

- El dominio del sistema requiere conocimientos del negocio en el área de aplicación, como la tecnología de la información, la industria del automóvil o la industria del juego de apuestas.
- La arquitectura del software y del sistema y las tecnologías utilizadas en el proyecto requieren, por ejemplo, conocimientos técnicos en lenguajes de programación, tecnología de interfaz o vulnerabilidades de seguridad.
- El ciclo de vida de desarrollo de software (CVDS) requiere, por ejemplo, conocimientos sobre niveles de prueba, roles de prueba y técnicas de prueba específicas.

Competencia Social

En el contexto de la prueba, la competencia social permite a los miembros del equipo de prueba comportarse adecuadamente en las relaciones con otros miembros del equipo y alcanzar los objetivos de prueba. En particular, incluye habilidades de comunicación, cooperación y resolución de conflictos (por

ejemplo, tratar de forma constructiva las condiciones de prueba que no son óptimas o informar de los defectos a los desarrolladores).

El desarrollo de software y la prueba suelen correr a cargo de distintos miembros de grupos (diferentes) que coordinan sus tareas mediante la comunicación. Las habilidades comunicativas, la capacidad de trabajar en equipo y la habilidad para resolver conflictos son necesarias para el éxito del proyecto. Sin embargo, el nivel requerido de habilidades sociales puede variar en función del contexto del proyecto. Por ejemplo, el Desarrollo Ágil de Software puede requerir mayores exigencias de habilidades sociales que los modelos de desarrollo secuencial centrados en los documentos, así como las pruebas externas.

Competencia Personal

La efectividad y la eficiencia de los miembros del equipo de prueba también dependen de su capacidad y su voluntad de desarrollarse, de sus habilidades y de sus actitudes. Por ejemplo, trabajar en un equipo ágil autoorganizado puede requerir un mayor nivel de autogestión y disciplina por parte de todos los miembros del equipo, mientras que un jefe de prueba de un equipo de prueba jerárquico, por ejemplo, necesita ser capaz de delegar trabajo. A menudo se requiere un alto grado de fiabilidad y resistencia, sobre todo en proyectos en los que el tiempo es un factor crítico. Además, la disposición a ayudar, a aprender y la apertura al cambio son importantes en un proceso de cambio en todos los modelos de desarrollo de software (CVDS).

3.1.3 Evaluación de las Habilidades de los Miembros del Equipo de Prueba

En muchos casos, los equipos de prueba se forman con el personal existente. Para comprender las capacidades de los miembros del equipo y la necesidad de desarrollo personal, la gestión de la prueba necesita evaluar las habilidades del equipo de prueba existente y compararlas con las habilidades requeridas, que pueden documentarse en una matriz de habilidades.

Existen modelos para ayudar a los equipos y a sus miembros a trabajar de forma más eficaz (por ejemplo, Meredith Belbin's "Team Roles", DISG® or PCM®). Según Belbin (Belbin, 2010), los equipos funcionan con eficacia cuando están formados por diferentes tipos de personalidad y roles. Estos modelos ayudan a los equipos a identificar las habilidades que tienen y las que les pueden faltar.

La competencia profesional y metodológica de los miembros del equipo de prueba puede evaluarse mediante la demostración de tareas de prueba habituales:

- Esbozar una estrategia de prueba y discutir la retroalimentación con colegas.
- Revisar la base de prueba y comunicar los hallazgos, lo que también puede revelar habilidades de comunicación.
- Determinar técnicas de prueba para alcanzar objetivos de prueba específicos en el contexto de un proyecto determinado.
- Aplicar adecuadamente diversas técnicas de prueba.
- Redactar un informe de completación de la prueba que incluya una evaluación de los resultados de la prueba.

Además, las habilidades pueden evaluarse mediante credenciales externas, certificaciones, experiencia laboral y títulos.

Especialmente en el desarrollo ágil de software, los equipos identifican las habilidades que necesitan participando regularmente en retrospectivas y recibiendo retroalimentación. Los entrenadores u orientadores experimentados ayudan a desarrollar sus habilidades y a identificar y resolver los vacíos de conocimientos.

3.1.4 Desarrollo de Habilidades de los Miembros del Equipo de Prueba

Es posible que un equipo de prueba no disponga de todas las habilidades necesarias al inicio de un proyecto. Aunque puede que no se disponga de un conjunto perfecto de individuos, un equipo fuerte puede equilibrar los puntos fuertes y débiles de los individuos.

La gestión de la prueba o el equipo de prueba pueden identificar las necesidades de desarrollo comparando las habilidades requeridas con las disponibles en una matriz de habilidades. Sobre esta base, pueden determinar los enfoques para el desarrollo de la competencia:

- La formación y la capacitación transmiten conocimientos y prácticas predefinidos, normalmente en un aula (virtual), (por ejemplo, enviando a personas a un curso de formación, celebrando sesiones de formación en la empresa, desarrollando formación a medida o utilizando cursos de aprendizaje electrónico en directo).
- El autoestudio es una forma de aprender sobre un tema que implica estudiar solo, en lugar de en un aula (virtual), (por ejemplo, leyendo libros, viendo vídeos grabados o investigando en recursos de Internet).
- El aprendizaje entre iguales, en el que los compañeros comparten conocimientos, ideas y experiencias y aprenden unos de otros.
- Orientar o entrenar son enfoques en los que un miembro del equipo que es nuevo en un rol recibe orientación individual de un entrenador, o conocimientos, habilidades y/o experiencia de un mentor experimentado. La persona experimentada actúa como un recurso continuo para proporcionar asesoramiento y asistencia.
- También es muy conocida la formación en el puesto de trabajo, que es una mezcla de autoaprendizaje, aprendizaje entre iguales y orientación.

No todos los enfoques para el desarrollo de competencias son igual de efectivos y eficientes. El autoestudio y la formación, por ejemplo, son muy adecuados para desarrollar la competencia profesional y metodológica. Por ello, los conocimientos básicos en materia de prueba pueden desarrollarse participando en las sesiones de formación ISTQB® o mediante el autoaprendizaje basado en los programas de estudio ISTQB®. Sin embargo, para desarrollar la competencia social y personal, se recomienda utilizar enfoques como la formación y el entrenamiento, que suelen ser más prometedores que el autoestudio. El intercambio social, la retroalimentación y la reflexión se encuentran entre los factores clave del éxito para desarrollar la competencia social y personal.

3.1.5 Habilidades de Gestión Necesarias para Dirigir un Equipo de Prueba

Cualquiera que quiera dirigir con éxito un equipo de prueba debe tener habilidades de gestión. Éstas incluyen la competencia profesional y metodológica en actividades de gestión fundamentales, (por ejemplo, planificación, monitorización del avance, control y suministro de información). Se requieren conocimientos y habilidades específicas para la gestión de la prueba (por ejemplo, conocimiento de los diferentes enfoques de prueba, el desarrollo de estrategias de prueba o el uso de técnicas de prueba o del ciclo de vida de desarrollo de software (CVDS) aplicado).

Dirigir o entrenar a un equipo de prueba significa actuar adecuadamente en las relaciones con otros miembros del equipo de prueba y tener la capacidad y la voluntad de evolucionar en circunstancias cambiantes. Por este motivo, la competencia social y personal son factores de éxito esenciales para dirigir un equipo de prueba. Esto incluye la resiliencia, la capacidad de delegar y la capacidad de ser aceptado por el equipo de prueba. Además, incluye la capacidad de hacer valer los intereses de la prueba en el proyecto, de promover los beneficios de la prueba y de comunicarse y resolver conflictos con todos los implicados.

Para incorporar personas al equipo de prueba, se requiere habilidad en el análisis de las condiciones sociales, de equipo y de trabajo. Estas habilidades ayudan a asegurar que el equipo de prueba se ajuste a las condiciones de trabajo o, si es posible, que las condiciones de trabajo se adapten al equipo de prueba. Además, los equipos de prueba están sujetos a procesos de desarrollo dinámicos y, por lo tanto, requieren habilidades situacionales (por ejemplo, según las fases del modelo de Tuckman de desarrollo de grupos pequeños) (Tuckman, 1965) (Bonebright, 2010):

- La voluntad de ayudar a los miembros del equipo de prueba a integrarse en el equipo de prueba.
- La habilidad para resolver conflictos dentro del equipo de prueba (Storming)

- Disciplina y liderazgo orientado a los objetivos para asegurar los valores y normas acordados (Norming)
- La habilidad de delegar para dar al equipo de prueba responsabilidad personal (Performing)
- La habilidad para actuar con aprecio y confianza con los miembros del equipo de prueba que se marchan (Adjourning).

3.1.6 Factores que Motivan o Desmotivan a un Equipo de Prueba en Determinadas Situaciones

Unos miembros del equipo de prueba satisfechos y motivados aumentan la productividad y el rendimiento y, por lo tanto, tienen un impacto significativo en el éxito de los proyectos. Cuando esto se consigue, la formación cruzada tiene lugar de manera informal, los miembros del equipo de prueba pueden gestionar su propia carga de trabajo y la gestión de la prueba dispone de más tiempo para ocuparse de las dificultades externas. La teoría de la motivación y la higiene (Herzberg, et al., 1993) distingue entre motivadores y factores de higiene:

Los motivadores se perciben conscientemente y pueden conducir al crecimiento y la satisfacción. Pueden incluir:

- Reconocimiento y aprecio por el trabajo realizado (por ejemplo, incentivos y cualquier otro tipo de enfoque que los miembros del equipo de prueba encuentren gratificante)
- Mayor responsabilidad y autonomía (por ejemplo, para definir los procesos de prueba en un equipo de prueba)
- Tareas interesantes, significativas y desafiantes que los miembros del equipo de prueba perciban como alcanzables y, al mismo tiempo, por las que merezca la pena esforzarse (por ejemplo, la selección e introducción de una nueva herramienta para la automatización de la prueba).
- Avance y desarrollo profesional (por ejemplo, la evolución del probador experimentado a jefe de pruebas o propietario del proceso de prueba).

Los factores de higiene suelen darse por sentados. Cumplirlos no conduce automáticamente a una mayor satisfacción. Si faltan, pueden tener un efecto desmotivador en los miembros del equipo de pruebas:

- Remuneración adecuada (por ejemplo, salario de mercado, horas extraordinarias remuneradas, buenas prestaciones sociales).
- Política de personal y estilo de gestión gratificantes (por ejemplo, gestión ajustada, objetivos realistas, protección contra el acceso externo y la sobrecarga)
- Condiciones de trabajo agradables (por ejemplo, especificaciones inequívocas, objetos de prueba maduros, defectos corregidos adecuadamente, lugar de trabajo apropiado, un entorno de prueba estable)
- La protección como necesidad existencial (por ejemplo, lugar de trabajo seguro y cumplimiento de los acuerdos).
- Buenas relaciones interpersonales (por ejemplo, con los compañeros de trabajo, los supervisores)

En consecuencia, la gestión de la prueba debe eliminar continuamente los factores desmotivadores y, al mismo tiempo.

Para más información consultar (Belbin, 2010) (Marston, 1999) (Kahler, 2008).

3.2 Relación con los Implicados

Introducción

En la gestión de la prueba, es importante optimizar la prueba para ofrecer un valor de negocio satisfactorio. Una prueba excesiva puede dar lugar a retrasos y costes poco razonables que superen los beneficios, mientras que una prueba insuficiente puede conducir a la entrega de un producto de baja calidad a los usuarios. El enfoque óptimo se encuentra entre estos dos extremos. Es responsabilidad del jefe de la prueba ayudar a los implicados a comprender este equilibrio y el valor añadido de la prueba para conseguirlo, sin olvidar, por ejemplo, las restricciones de tiempo típicas de un proyecto.

3.2.1 Coste de la Calidad

Los beneficios de la prueba se ven contrarrestados por los costes de la calidad. Un medio de cuantificar el coste total de los esfuerzos y los defectos relacionados con la calidad se denomina coste de la calidad. El coste de la calidad consiste en clasificar los costes del proyecto y de la operación en cuatro categorías relacionadas con los costes de los defectos del producto:

- **Costes de prevención de defectos:** El coste de todas las actividades planificadas y proactivas para prevenir la mala calidad (por ejemplo, la cualificación de los desarrolladores para sus tareas, como la formación en la creación de código mantenible o seguro, la revisión de la base de prueba lo antes posible y la comunicación adecuada dentro del equipo).
- **Costes de evaluación:** El coste de todas las actividades encaminadas a la detección de defectos (por ejemplo, la realización de pruebas estáticas y pruebas dinámicas y la revisión de los productos de trabajo).
- **Costes de fallo interno:** El coste de todas las actividades reactivas (por ejemplo, arreglar los defectos encontrados durante las pruebas, proporcionar soluciones provisionales).
- **Costes de fallo externo:** El coste de todas las actividades reactivas y sin valor añadido (por ejemplo, pérdida de ingresos, activos, salud humana, vida humana o medio ambiente, costes legales relacionados con la corrección de defectos, pruebas, despliegue y asistencia debido a que se entrega al cliente un producto defectuoso (“post release”), corrección de defectos de campo (señalados por los clientes)).

Los costes totales de evaluación y los costes de fallo interno suelen ser significativamente inferiores a los costes de fallo externo. Por lo tanto, esto hace que la prueba sea extremadamente valiosa. Al determinar los costes en estas cuatro categorías, los jefes de prueba pueden crear un caso de negocio convincente para la prueba.

Hay más enfoques que se pueden tener en cuenta para definir el coste de la calidad. El programa de estudio de ISTQB® admite dos de ellos. Este programa de estudio se basa en el enfoque de Feigenbaum, y el programa de estudio de nivel básico V.4 presenta el enfoque de Boehm (véase el programa de estudio de nivel básico V.4, sección 1.3, Principios de la prueba). Estos dos enfoques se han seleccionado para alcanzar una comprensión más amplia del coste de la calidad. El enfoque de Feigenbaum (Feigenbaum, Nov/Dic 1956) considera la calidad como un proceso orientado al cliente y a toda la empresa, mientras que el enfoque de Boehm (Boehm, 1979) se concentra en la compensación entre el coste de la prevención y el coste del fallo en el desarrollo de software (Hadjicostas, 2004).

3.2.2 Relación Coste-Beneficio de la Prueba

Aunque la mayoría de las organizaciones consideran que la prueba es valiosa en algún sentido, pocos directivos, incluidos los jefes de prueba, pueden cuantificar, describir o articular ese valor. Además, muchos directores de prueba, jefes de prueba y probadores se concentran en los detalles operativos de las pruebas (es decir, en aspectos específicos de la tarea de prueba o del nivel de prueba), mientras que ignoran las

cuestiones tácticas y estratégicas más amplias (de más alto nivel) relacionadas con la prueba, que preocupan a otros implicados, especialmente a los directivos.

La prueba aporta beneficios a la organización, al proyecto y/o a la operación tanto de forma cuantitativa como cualitativa:

- **Beneficios Cualitativos**
 - Los **beneficios cualitativos** incluyen la mejora de la reputación respecto a la calidad, entregas más fáciles y predecibles, aumento de la confianza, protección frente a la responsabilidad legal y una reducción del riesgo de pérdida de misiones enteras o incluso de vidas.
- **Beneficios Cuantitativos**
 - Entre los **beneficios cuantitativos** se incluyen los defectos encontrados, prevenidos o corregidos antes de la entrega, los defectos conocidos antes de la entrega (es decir, no corregidos pero documentados, quizás con soluciones alternativas), los beneficios en términos de costes (Bohm 1981, Böhler 2008), la reducción del nivel de riesgo mediante la realización de pruebas y la entrega de información sobre el estado del proyecto, el proceso y el producto.

El coste de la calidad por defecto de prueba se calcula del siguiente modo:

Ahorro Medio por Defecto = Coste Medio de Fallos Externos - (Coste Medio de Evaluación + Coste Medio de Fallos Internos)

Coste total de la calidad = (Costes de prevención de defectos + (Costes de evaluación * Número de defectos encontrados antes de la entrega) + (Costes de fallo interno * Número de defectos encontrados antes de la entrega) + (Costes de fallo externo * Número de defectos encontrados después de la entrega))

Como ejemplo, suponga que ha calculado el siguiente coste de la calidad por defecto para un producto:

- Costes de prevención de defectos: \$180**
- Costes medios de evaluación por defecto: 500 \$.**
- Costes medios de fallo interno por defecto: 200 \$.**
- Costes medios de fallo externo por defecto: 4.000 \$.**

Los costes medios de prevención de defectos, los costes de evaluación y los costes de fallo interno se calculan utilizando el número de defectos encontrados antes de la entrega, mientras que los costes medios de fallo externo se calculan utilizando el número de defectos encontrados después de la entrega. Con estos valores, se puede calcular el ahorro medio por defecto de la siguiente manera:

Ahorro medio por defecto = 4.000 \$ - (500 + 200 \$) = 3.300 \$.

La curva de Boehm es una representación gráfica de los costes de corrección de defectos a lo largo del tiempo en el ciclo de vida de desarrollo de software (CVDS). De ella se deduce que la prueba debería realizarse antes en el ciclo de vida de desarrollo de software (CVDS) para reducir el coste de la corrección de defectos. La curva de Boehm muestra que los costes de fallo interno, o el coste de arreglar un defecto, aumenta cuanto más tarde se descubre un defecto en el ciclo de vida de desarrollo de software (CVDS). Utilizando esta información, el jefe de la prueba debe esforzarse por encontrar la relación óptima entre los costes de prevención de defectos frente a los costes internos y externos.

El esfuerzo de prueba debe basarse en el riesgo específico del proyecto y del producto y en el riesgo que la empresa está dispuesta a asumir. Demasiadas pruebas pueden causar costes más elevados que el beneficio de la reducción del nivel de riesgo. Si se prueba demasiado poco, los defectos omitidos pueden suponer un alto riesgo que genere costes más elevados que los que habrían supuesto las pruebas omitidas. La prueba basada en el riesgo (véase la sección 1.3 de este programa de estudio, Prueba Basada en el Riesgo) apoya la relación coste-beneficio de las pruebas invirtiendo niveles de esfuerzo de prueba proporcionales a los niveles de riesgo y priorizando las pruebas en función de sus niveles de riesgo.

Los jefes de prueba deben entender cuáles de estos beneficios y costes se aplican a su organización, proyecto y/o operación, y ser capaces de comunicar el valor añadido de las pruebas en términos de estos beneficios y del coste de la calidad por defecto.

BORRADOR

4 Referencias

4.1 Estándares

- IEC 61508 (2010) Functional safety of electrical/electronic/programmable electronic safety-related systems -Parts 1 to 7
- ISO/IEC/IEEE 29119-2 (2021) ISO/IEC/IEEE 29119-2 Software and systems Engineering-Software testing-Part 2 Test processes
- ISO/IEC/IEEE 29119-3 (2021) ISO/IEC/IEEE 29119-3 Software and systems Engineering-Software testing-Part 3 Test documentation

4.2 Documentos de ISTQB®

- ISTQB® Certified Tester Agile Test Leadership at Scale Syllabus v2.0 (2023)
- ISTQB® Certified Tester Foundation Level Syllabus V.4 (2023)
- ISTQB® Certified Tester Expert Level Test Management Syllabus v1.0 (2011)
- ISTQB® Certified Tester Expert Level Improving the Test Process Syllabus v1.0.1 (2011)

4.3 Libros

- Basili, V., Trendowicz, A., Kowalczyk, M., Heidrich, J., Seaman, C., Münch, J., & Rombach, D. (2014). *Aligning Organizations Through Measurement – The QM+ Strategies Approach*. Springer International.
- Bath, G., & van Veenendaal, E. (2014). *Improving the Test Process - chapter 6: Process for Improvement*. Rocky Nook.
- Belbin, R. M. (2010). *Management Teams: Why They Succeed or Fail*. London: Routledge.
- Black, R. (2009). *Managing the Testing Process*, 3rd Edition. John Wiley & Sons.
- Boehm, B. (1979). *Software Engineering Economics*. Prentice-Hall.
- Bonebright, D. A. (2010). 40 years of storming: a historical review of Tuckman's model of small group development (1 Ausg., Bd. 13). *Human Resource Development International*, 1, 2010, Vol. 13.
- Craig, R., & Jaskiel, S. P. (2002). *Systematic Software Testing*. Artech House.
- Derby, E., & Larsen, D. (2006). *Agile Retrospectives – Making Good Teams Great*. The Pragmatic Bookshelf.
- Erpenbeck, J., & von Rosenstiel, L. (2017). *Handbuch Kompetenzmessung*. Stuttgart: Schäffer-Poeschel.
- Fowler, M. (2010). Hybrid development processes. *IEEE Software*, 27(2), 57-63.
- Erpenbeck, J., & von Rosenstiel, L. (2017). *Handbuch Kompetenzmessung*. Stuttgart: Schäffer-Poeschel.
- Fowler, M. (2010). Hybrid development processes. *IEEE Software*, 27(2), 57-63.
- Herzberg, F., Mausner, B., & Bloch Snyderman, B. (1993). *Motivation to Work*. London: Routledge.

- Kahler, T. (2008). The Process Therapy Model: The Six Personality Types with Adaptations. Taibi Kahler Associates, Inc.ç
- Marston, W. M. (1999). Emotions Of Normal People. London: Routledge.
- Sonntag, K., & Schmidt-Rathjens, C. (2005). Anforderungsanalyse und Kompetenzmodelle. Wiesbaden: VS Verlag für Sozialwissenschaften.
- Tuckman, B. W. (1965). Developmental sequence in small groups (Bd. 63(6)). Psychological Bulletin.
- van Ewijk, A. (2013). TPI NEXT – Business Driven Test Process Improvement,. Sogeti Nederland B.V.
- van Solingen, R., & Berghout, E. (1999). The Goal Question Metric Method – A Practical Guide for Quality Improvement of Software Development. McGraw-Hill.
- van Veenendaal, E. (2012). The PRISMA Approach: Practical Risk-Based Testing. UTN Publishers.
- van Veenendaal, E. (2020). TMMi in the Agile world, version 1.4. TMMi Foundation.
- van Veenendaal, E., & Cannegieter, J. J. (2011). The Little TMMi – Objective-Driven Test Process Improvement. UTN Publishers.

4.4 Artículos

- Feigenbaum, Armand V. (Nov/Dec 1956) Harvard Business Review, Vol. 34 Issue 6, p93-101.
- Hadjicostas, Evsevios (2004) Total Quality Management and Cost of Quality, Springer
- https://link.springer.com/chapter/10.1007/978-3-662-09621-5_7

4.5 Páginas Web

- www.tmmi.org Test Maturity Model integration (TMMi®); last visit January 31st, 2024
- www.tmap.net Test Process Improvement (TPI); last visit January 31st, 2024
- www.wikipedia.org/wiki/PDCA Plan-Do-Check-Act; last visit January 31st, 2024

Las referencias anteriores apuntan a información disponible en Internet y en otros lugares. Aunque esas referencias fueron comprobadas en el momento de la publicación de este programa de estudio, el ISTQB® no se hace responsable de la disponibilidad de dichas referencias.

5 Anexo A - Objetivos de Aprendizaje/Nivel Cognitivo de Conocimiento

Los siguientes objetivos de aprendizaje se definen como aplicables a este programa de estudio. Cada tema del programa de estudio se examinará de acuerdo con el objetivo de aprendizaje correspondiente. Los objetivos de aprendizaje comienzan con un verbo de acción correspondiente a su nivel cognitivo de conocimiento, como se indica a continuación.

Nivel 1: Recordar (K1): el candidato recordará, reconocerá y rememorará un término o concepto.

Verbos de acción: identificar, recordar, rememorar, reconocer.

Ejemplos:

- "Identificar los objetivos característicos de la prueba".
- "Recordar los conceptos de la pirámide de prueba".
- "Reconocer cómo un probador añade valor a la planificación de la iteración y la entrega".

Nivel 2: Comprender (K2): el candidato puede seleccionar las razones o explicaciones de los enunciados relacionados con el tema y puede resumir, comparar, clasificar y dar ejemplos para el concepto de prueba.

Verbos de acción: clasificar, comparar, contrastar, diferenciar, distinguir, ejemplificar, explicar, dar ejemplos, interpretar, resumir.

Ejemplos:

- "Clasificar las diferentes opciones para redactar los criterios de aceptación".
- "Comparar los diferentes roles en la prueba" (buscar similitudes, diferencias o ambas).
- "Distinguir entre riesgos de proyecto y riesgos de producto" (permite diferenciar conceptos).
- "Dar ejemplos de la finalidad y el contenido de un plan de prueba".
- "Explicar el impacto del contexto en el proceso de prueba".
- "Resumir las actividades del proceso de revisión".

Nivel 3: Aplicar (K3): el candidato puede llevar a cabo un procedimiento cuando se enfrenta a una tarea conocida, o seleccionar el procedimiento correcto y aplicarlo a un contexto determinado.

Verbos de acción: aplicar, implementar, preparar, utilizar.

Ejemplos:

- "Aplicar la priorización de casos de prueba" (debe referirse a un procedimiento, técnica, proceso, algoritmo, etc.).
- "Preparar un informe de defectos".
- "Utilizar el análisis del valor frontera para obtener casos de prueba".

Referencias para los niveles cognitivos de los objetivos de aprendizaje

- Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching
- Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

6 Anexo B - Matriz de Trazabilidad de los Resultados de Negocio con respecto a los Objetivos de Aprendizaje

Esta sección enumera el número de objetivos de aprendizaje de nivel básico relacionados con los resultados de negocio y la trazabilidad entre los resultados de negocio de nivel básico y los objetivos de aprendizaje de nivel básico.

Resultados de Negocio: Nivel Básico		BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11
TM_01	Gestionar la prueba en varios proyectos de desarrollo de software mediante la aplicación de procesos de gestión de la prueba establecidos para el equipo de proyecto o la organización de pruebas.	12										
TM_02	Identificar a los implicados en las pruebas y los modelos de ciclo de vida de desarrollo del software que sean relevantes en un contexto determinado.		4									
TM_03	Organizar sesiones de identificación del riesgo y de evaluación del riesgo dentro de cualquier ciclo de vida de desarrollo de software y utilizar los resultados para guiar las pruebas con el fin de alcanzar los objetivos de prueba.			6								
TM_04	Definir una estrategia de prueba del proyecto consistente con la estrategia de prueba organizativa y el contexto del proyecto.				11							
TM_05	Monitorizar y controlar continuamente las pruebas para lograr los objetivos del proyecto.					4						
TM_06	Evaluar e informar del avance de las pruebas a los implicados en el proyecto.						3					
TM_07	Identificar las habilidades necesarias y desarrollarlas en su equipo.							6				
TM_08	Preparar y presentar un caso de negocio para las pruebas en distintos contextos que describa los costes y los beneficios esperados.								5			

Resultados de Negocio: Nivel Básico		BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11
TM_09	Dirigir las actividades de mejora del proceso de prueba en proyectos o flujos de productos de desarrollo de software y contribuir a las iniciativas organizativas de mejora del proceso de prueba.									5		
TM_10	Planificar las actividades de prueba, incluida la infraestructura de la prueba necesaria y estimar el esfuerzo requerido para la prueba.										9	
TM_11	Crear informes de defectos y un flujo de trabajo de defectos adecuado para un ciclo de vida de desarrollo del software.											6

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11
1	Gestión de las Actividades de Prueba												
1.1	El proceso de Prueba												
TM-1.1.1	Resumir la planificación de la prueba.	(K2)	X			X							
TM-1.1.2	Resumir la monitorización de la prueba.	(K2)	X				X						
TM-1.1.3	Resumir la compleción de la prueba.	(K2)	X					X					
1.2	El Contexto de la Prueba												
TM-1.2.1	Comparar por qué los distintos implicados están interesados en la prueba.	(K2)		X		X							
TM-1.2.2	Explicar por qué es importante el conocimiento de los implicados en la gestión de la prueba.	(K2)		X		X							
TM-1.2.3	Explicar la prueba en un modelo híbrido de desarrollo de software.	(K2)		X		X							
TM-1.2.4	Resumir las actividades de gestión de la prueba para varios ciclos de vida de desarrollo del software.	(K2)	X	X		X							
TM-1.2.5	Comparar las actividades de gestión de la prueba en varios niveles de prueba.	(K2)	X			X							

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11
TM-1.2.6	Comparar las actividades de gestión de la prueba para varios tipos de prueba.	(K2)	X			X							
TM-1.2.7	Analizar un proyecto dado y determinar las actividades de gestión de la prueba que enfatizan la planificación de la prueba, la monitorización de la prueba y el control de la prueba.	(K4)	X			X							
1.3	Prueba Basada en el Riesgo												
TM-1.3.1	Explicar las distintas medidas que se toman en la prueba basada en el riesgo para responder a riesgos.	(K2)			X								
TM-1.3.2	Aportar ejemplos de las distintas técnicas que puede utilizar un jefe de prueba para identificar riesgos relacionados con la calidad del producto.	(K2)			X								
TM-1.3.3	Resumir los factores que determinan los niveles de riesgo relacionados con la calidad del producto.	(K2)			X								
TM-1.3.4	Seleccionar las actividades de prueba adecuadas para mitigar los riesgos en función de su nivel de riesgo en un contexto determinado.	(K4)			X								
TM-1.3.5	Diferenciar entre ejemplos pesados y ligeros de técnicas de prueba basadas en el riesgo.	(K2)			X								
TM-1.3.6	Aportar ejemplos de métricas de éxito y dificultades asociadas a la prueba basada en el riesgo.	(K2)			X								
1.4	La Estrategia de Prueba del Proyecto												
TM-1.4.1	Explicar las opciones típicas de un enfoque de prueba.	(K2)				X							

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11
TM-1.4.2	Analizar una estrategia de prueba organizativa y el contexto del proyecto para seleccionar el enfoque de prueba adecuado.	(K4)				X							
TM-1.4.3	Utilizar la metodología de objetivos S.M.A.R.T. para definir objetivos de prueba y criterios de salida medibles.	(K3)				X							
1.5.	Mejora del Proceso de Prueba												
TM-1.5.1	Explicar cómo utilizar el modelo de IDEAL para la mejora del proceso de prueba en un proyecto determinado.	(K2)									X		
TM-1.5.2	Resumir el enfoque de mejora de proceso basado en modelos y comprender cómo aplicarlo en el contexto de un proyecto.	(K2)									X		
TM-1.5.3	Resumir el enfoque de mejora del proceso basado en el análisis y comprender cómo aplicarlo en el contexto de un proyecto.	(K2)									X		
TM-1.5.4	Aplicar una retrospectiva de proyecto o de iteración para evaluar los procesos de prueba y descubrir áreas de prueba que puedan ser objeto de mejora.	(K3)									X		
1.6	Herramientas de prueba												
TM-1.6.1	Resumir las buenas prácticas para la introducción de herramientas.	(K2)										X	
TM-1.6.2	Explicar el impacto de los diferentes aspectos técnicos y de negocio en la decisión sobre un tipo de herramienta.	(K2)										X	
TM-1.6.3	Analizar una situación dada para crear un plan para la selección de una herramienta que cubra riesgos, costes y beneficios.	(K4)										X	

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11
TM-1.6.4	Diferenciar entre las etapas del ciclo de vida de una herramienta.	(K2)										X	
TM-1.6.5	Aportar ejemplos de recopilación y evaluación de métricas mediante el uso de herramientas.	(K2)									X	X	
2	Gestión de Producto												
2.1	Métrica de Prueba												
TM-2.1.1	Dar ejemplos de métricas para alcanzar los objetivos de prueba.	(K2)					X						
TM-2.1.2	Explicar cómo controlar el avance de la prueba utilizando métricas de prueba.	(K2)					X						
TM-2.1.3	Analizar los resultados de prueba para crear informes de prueba que empoderen a los implicados en la toma de decisiones.	(K4)					X	X					
2.2	Estimación de la Prueba												
TM-2.2.1	Explicar los factores que se necesitan tener en cuenta en la estimación de la prueba.	(K2)	X							X		X	
TM-2.2.2	Dar ejemplos de factores que pueden influir en la estimación de la prueba.	(K2)	X							X		X	
TM-2.2.3	Seleccionar una técnica o enfoque de estimación de la prueba apropiado para un contexto determinado.	(K4)	X							X		X	

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11
2.3	Gestión de Defectos												
TM-2.3.1	Implementar un proceso de gestión de defectos, incluyendo el flujo de trabajo de defectos, que se pueda utilizar para monitorizar y controlar los defectos.	(K3)											X
TM-2.3.2	Explicar el proceso y los participantes necesarios para una gestión de defectos eficaz.	(K2)											X
TM-2.3.3	Explicar los detalles de la gestión de defectos en el desarrollo ágil de software.	(K2)	X										X
TM-2.3.4	Explicar los retos de la gestión de defectos en el desarrollo de software híbrido.	(K2)	X										X
TM-2.3.5	Utilizar los datos y la información de clasificación que se deben recopilar durante la gestión de defectos.	(K3)											X
TM-2.3.6	Explicar cómo se pueden utilizar las estadísticas del informe de defecto para idear la mejora del proceso.	(K2)										X	X
3	Gestión de Equipo												
3.1	El Equipo de Prueba												
TM-3.1.1	Dar ejemplos de las competencias típicas que necesitan los miembros del equipo de prueba dentro de cuatro áreas de competencia.	(K2)							X				
TM-3.1.2	Analizar el contexto de un proyecto determinado para determinar las competencias necesarias para los miembros del equipo de prueba.	(K4)							X				

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11
TM-3.1.3	Explicar las técnicas típicas de evaluación de competencias para los miembros del equipo de prueba.	(K2)							X				
TM-3.1.4	Diferenciar entre los enfoques típicos para desarrollar las competencias de los miembros del equipo de pruebas.	(K2)							X				
TM-3.1.5	Explicar las competencias necesarias para gestionar un equipo de prueba.	(K2)							X				
TM-3.1.6	Dar ejemplos de factores de motivación e higiene para los miembros de un equipo de prueba.	(K2)							X				
3.2	Relación con los Implicados												
TM-3.2.1	Dar ejemplos para cada una de las cuatro categorías determinantes del coste de la calidad.	(K2)								X			
TM-3.2.2	Aplicar un cálculo de coste-beneficio para estimar el valor añadido de la prueba para los implicados.	(K3)						X		X			

7 Anexo C - Traducciones Específicas, Definiciones, Acrónimos, Abreviaturas y Notas

En este capítulo se presentarán traducciones específicas, definiciones, acrónimos, abreviaturas y notas de términos y conceptos que no forman parte del Glosario de Términos del ISTQB y tampoco están incluidos en su traducción al idioma español o castellano. Se ha incorporado este apartado para facilitar la lectura y comprensión de esta “Programa de Estudio”.

Las “Traducciones Específicas, Definiciones, Acrónimos, Abreviaturas y Notas” de este apartado están identificados con el texto **Consultar **** en el pie de página. Sólo se identificará la primera ocurrencia del término.

Es conveniente observar que las traducciones de los distintos términos sean o no del glosario, se han realizado con el objetivo de evitar conflictos en sus traducciones y siguiendo las normas de traducción del SSTQB. Algunos conflictos surgen en el momento de traducir un nuevo término o un nuevo programa de estudio.

Traducciones específicas			
Traducción del término al español	Término en el idioma (Id) de la fuente	Idioma	Observaciones
déjà vu	déjà vu	FR	
planificar-hacer-comprobar-actuar	plan-do-check-act	EN	
habilidad	skill	EN	
competencia	competence	EN	

Tabla 3 - Traducciones Específicas

Definiciones, Acrónimos y Abreviaturas			
Tipo de Término	Idioma del Acrónimo	Término	Descripción

Tabla 4 – Definiciones, Acrónimos y Abreviaturas

10.1 Notas

Código de la nota	Nota
NT – CTAL – TM - 0001	<p>El término “déjà vu” se puede traducir como paramnesia. La paramnesia es una alteración de la memoria por la que el sujeto cree recordar situaciones que no han ocurrido o modifica algunas circunstancias de aquellas que se han producido.</p> <p>Una traducción literal y coloquial sería “ya visto/conocido”. En la traducción del programa de estudio se mantendrá el término “déjà vu” debido a</p>

Tabla 5 – Notas