

Probador Certificado de ISTQB®
Estrategia de automatización de pruebas
Programa de Estudio

Versión ES V1.0

Traducción realizada por
Hispanic America Software Testing Board
con el apoyo de
Spanish Software Testing Qualifications Board

Traducción del Programa de Estudio
“ISTQB®
Certified Tester Test Automation Strategy (CT-TAS), Version V1.0”



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 del International Software Testing Qualifications Board.

Copyright © 2024 los autores del programa de estudios Test Automation Strategy v1.0: Andrew Pollner (Presidente), Péter Földházi, Patrick Quilter, Gergely Ágnesz, László Szikszai

Todos los derechos reservados. Por la presente, los autores ceden 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 de 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
Programa de estudios versión 1.0	03/05/2024	CT-TAS v1.0 Liberado AG
Historial de Revisiones – Traducción en idioma español		
Versión	Fecha	Observaciones
Programa de estudios versión 1.0	03/05/2024	CT-TAS v1.0

Tabla de Contenidos

Nota sobre Derechos de Propiedad Intelectual	2
Historial de Revisiones.....	3
Tabla de Contenidos	4
Agradecimientos	7
Notas de la Versión en Idioma Español.....	8
0 Introducción	9
0.1 Objetivo de este Programa de Estudio	9
0.2 Estrategia de automatización de pruebas en pruebas de software	9
0.3 Carrera Profesional para Probadores.....	9
0.4 Resultados de Negocio.....	10
0.5 Objetivos de Aprendizaje y Niveles de Conocimiento Evaluables	11
0.6 El Examen de Certificación.....	11
0.7 Acreditación	12
0.8 Tratamiento de Estándares	12
0.9 Actualización.....	12
0.10 Nivel de Detalle.....	12
0.11 Organización de este Programa de Estudio	13
1 Introducción y objetivos de la estrategia de automatización de pruebas– (K2).....	15
1.1 Factores de éxito de un proyecto de automatización de pruebas	16
1.1.1 Definir metas y objetivos para la estrategia de automatización de pruebas	16
1.1.2 Identificar los factores técnicos de éxito en un proyecto de automatización de pruebas	16
1.1.3 Resumir los criterios de inversión adecuados para seleccionar proyectos candidatos a la automatización de pruebas	17
2 Recursos de automatización de pruebas: 60 minutos (K2)	18
2.1 Recursos de automatización de pruebas: 60 minutos (K2).....	19
2.1.1 Comparar soluciones técnicas alternativas en relación con el costo de propiedad	19
2.1.2 Explicar las consideraciones del modelo de licencia para herramientas de automatización de pruebas.....	19
2.1.3 Proporcionar ejemplos de factores a tener en cuenta al definir una estrategia de automatización de pruebas	20
2.2 Funciones y responsabilidades en la automatización de pruebas	21
2.2.1 Resumir los roles y las habilidades necesarias para una solución de automatización de pruebas exitosa	21
3 Preparación para la automatización de pruebas: 225 minutos (K3).....	22
3.1.1 Diferenciar las distribuciones de automatización de pruebas	23
3.1.2 Seleccionar un enfoque de automatización de pruebas basado en la arquitectura del sistema bajo prueba.....	24
3.1.3 Demostrar formas de optimizar la distribución de automatización de pruebas para lograr enfoques de desplazamiento a la izquierda y desplazamiento a la derecha	24
3.2 Consideraciones estratégicas en los distintos modelos de ciclo de vida del desarrollo de software	25
3.2.1 Explicación de la conformidad de los proyectos de automatización de pruebas con los modelos de ciclo de vida de desarrollo de software heredados	25
3.2.2 Explicar cómo los proyectos de automatización de pruebas se ajustan a las mejores prácticas de desarrollo de software ágil que admiten la automatización de pruebas	26
3.2.3 Prepárese para que los proyectos de automatización de pruebas se ajusten a las mejores prácticas de DevOps para lograr pruebas continuas	26
3.3 Aplicabilidad y viabilidad de la automatización de pruebas	26

3.3.1	Explicación de los criterios para determinar la conveniencia de las pruebas para la automatización de pruebas	26
3.3.2	Identifique los retos que sólo la automatización de pruebas puede abordar	27
3.3.3	Identificar condiciones de prueba difíciles de automatizar	27
4	Estrategias de implementación y lanzamiento organizacional para la automatización de pruebas (K2)	29
4.1	Planificación de la solución de automatización de pruebas	30
4.1.1	Identificar formas en que la automatización de pruebas permite acortar los tiempos de comercialización	30
4.1.2	Identificar formas en las que la automatización de pruebas ayuda a verificar los defectos informados	30
4.1.3	Definir enfoques que permitan el desarrollo de escenarios operativamente relevantes para la automatización de pruebas	31
4.2	Estrategias de implementación de automatización de pruebas	32
4.2.1	Definir la estrategia de implementación de automatización de pruebas	32
4.2.2	Identificar los riesgos de la automatización de pruebas durante la implementación	33
4.2.3	Definir enfoques para mitigar los riesgos de implementación	34
4.3	Dependencias dentro del entorno de pruebas	35
4.3.1	Definir enfoques para mitigar los riesgos de implementación	35
4.3.2	Identificar los componentes de infraestructura y las dependencias de la automatización de pruebas	35
4.3.3	Definir los requisitos de interfaz y datos de automatización de pruebas	36
5	Análisis del impacto de la automatización de pruebas (K3)	37
5.1	Inversión en la creación y el mantenimiento de la automatización de pruebas	38
5.1.1	Mostrar el retorno de la inversión de crear una solución de automatización de pruebas	38
5.2	Métricas de automatización de pruebas	39
5.2.1	Clasificación de métricas para la automatización de pruebas	39
5.3	El valor de la automatización de pruebas a nivel de proyecto y de organización	41
5.3.1	Identificar las consideraciones organizativas para el uso de la automatización de pruebas	41
5.3.2	Analizar las características del proyecto que ayudan a determinar la implementación óptima de los objetivos de prueba de la automatización de pruebas	42
5.4	Decisiones tomadas a partir de los informes de automatización de pruebas	43
5.4.1	Analizar los datos de los informes de pruebas para fundamentar la toma de decisiones	43
6	Estrategias de implementación y mejora para la automatización de pruebas (K3)	45
6.1	Transición de actividades de pruebas manuales a pruebas continuas	46
6.1.1	Describir los factores y las actividades de planificación en la transición de las pruebas manuales a la automatización de pruebas	46
6.1.2	Describir los factores y las actividades de planificación en la transición de la automatización de pruebas a las pruebas continuas	47
6.2	Estrategia de automatización de pruebas en toda la organización	48
6.2.1	Realizar una evaluación de los activos y prácticas de automatización de pruebas para identificar áreas de mejora	48
7	Referencias	50
7.1	Estándares	50
7.2	Documentos ISTQB®	51
7.3	Libros	51
7.4	Artículos y Páginas Web	52
8	Apéndice A - Objetivos de Aprendizaje/Nivel Cognitivo de Conocimiento	53
9	Apéndice B - Matriz de Trazabilidad de los Resultados de Negocio con respecto a los Objetivos de Aprendizaje	55
10	Apéndice C - Notas de la versión	59

11 Apéndice D - Términos específicos de cada ámbito 60

12 Índice 61

Agradecimientos

Este documento ha sido hecho público formalmente por la Asamblea General del ISTQB® el 3 de mayo de 2024.

Fue elaborado por un equipo del “Test Automation Task Force del Specialist Working Group” de la International Software Testing Qualifications Board: Graham Bath (Presidente del Specialist Working Group), Andrew Pollner (Vicepresidente del Specialist Working Group y Presidente del Test Automation Task Force), Péter Földházi, Patrick Quilter, Gergely Ágnes, László Szikszai. Los revisores del Test Automation Task Force fueron: Armin Beer, Armin Born, Geza Bujdosó, Renzo Cerquozzi, Jan Giesen, Arnika Hryszko, Kari Kakkonen, Gary Mogyorodi, Chris van Bael, Carsten Weise, Marc-Florian Wendland.

Revisor técnico: Gary Mogyorodi

Las siguientes personas participaron en la revisión, comentarios y votación de este programa de estudios: Horváth Ágota, Laura Albert, Prasunkumar Banerjee, Jürgen Beniermann, Armin Born, Piet de Roo, Nicola De Rosa, Dingguofu Ding Guofu, Elizabeta Fournieret, Jan Giesen, Erik Haartmans, Matthias Hamburg, Tobias Horn, Mattijs Kemmink, Ilia Kulakov, Ashish Kumar, Vincenzo Marrazzo, Marton Matyas, Patricia McQuaid, Smitha Mohandas, Ingvar Nordström, Sreeja Padmakumari, Nishan Portoyan, Meile Posthuma, Swapnil Shah, Péter Sótér, Szilard Szell, Richard Taylor, Giancarlo Tomasig, Chris Van Bael, Daniel Van der Zwan, Carsten Weise, Marc-Florian Wendland, Claude Zhang.

Notas de la Versión en Idioma Español

Este “Programa de Estudio de Probador Certificado-Estrategia de automatización de pruebas del ISTQB®, Versión 1.0” ha sido traducido por el Hispanic America Software Testing Board (HASTQB) con el apoyo de Spanish Software Testing Qualifications Board (SSTQB).

El equipo de traducción y revisión para este programa de estudio es el siguiente:

Responsables de la traducción: Irma Infante, (HASTQB, Perú) y Rodrigo Flores (México)

Revisor: Alfonsina Morgavi (HASTQB, Argentina), Fiorella Hernández Ochoa (Brightest) y Emilie Potin-Suau (Brightest)

El Comité Ejecutivo del HASTQB agradece especialmente sus aportaciones.

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

Medellín, Colombia, 20 de mayo de 2025

0 Introducción

0.1 Objetivo de este Programa de Estudio

Este programa de estudios constituye la base para la calificación internacional de pruebas de software para la certificación de Especialista en Estrategia de Automatización de Pruebas (CT-TAS). Proporciona este programa de estudios de la siguiente manera:

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.

El ISTQB puede permitir que otras entidades utilicen este programa de estudio para otros fines, siempre y cuando soliciten y obtengan permiso previo y por escrito por parte del ISTQB.

0.2 Estrategia de automatización de pruebas en pruebas de software

La certificación de Especialista en Estrategia de Automatización de Pruebas está dirigida a cualquier persona involucrada en pruebas de software y automatización de pruebas. Esto incluye personas en roles tales como probadores, analistas de pruebas, ingenieros de automatización de pruebas, consultores de pruebas, arquitectos de pruebas, gerentes de pruebas y desarrolladores de software. Esta certificación de Especialista también es apropiada para cualquier persona que desee tener conocimientos básicos de automatización de pruebas, como gerentes de proyectos, gerentes de calidad, gerentes de desarrollo de software, analistas comerciales, directores de TI y consultores de gestión.

El programa de estudios de Especialista en Estrategia de Automatización de Pruebas (CT-TAS) presenta múltiples factores que entran en juego al planificar la automatización de pruebas dentro de una organización. Los aspectos de implementación de ingeniería técnica de los métodos de automatización de pruebas y las mejores prácticas no están incluidos en este programa de estudios y se cubren por separado en el programa CTAL-TAE de ISTQB.

La estrategia de automatización de pruebas aborda las necesidades de automatización de pruebas más allá de los desafíos de integración y de implementación de herramientas técnicas. Una visión estratégica de la automatización de pruebas proporciona una visión de la implementación en todos los proyectos dentro de una organización de manera sistemática y consistente que, en última instancia, demuestra valor para la organización.

0.3 Carrera Profesional para Probadores

El esquema ISTQB® brinda apoyo a los profesionales de las pruebas en todas las etapas de sus carreras, ofreciendo tanto amplitud como profundidad de conocimientos. Las personas que logran la certificación

Especialista en Estrategia de Automatización de Pruebas ISTQB®, también puede interesarles la certificación de Especialista en Ingeniería de Automatización de Pruebas (CTAL-TAE).

Personas que alcanzan la certificación ISTQB® Certified Tester Test Automation Strategy Specialist también pueden estar interesados en los niveles avanzados core (analista de pruebas, analista técnico de pruebas y jefe de pruebas) y, posteriormente, el nivel experto (gestión de pruebas o mejora del proceso de pruebas). Cualquiera que busque desarrollar habilidades en prácticas de pruebas en un área de entorno ágil podría considerar las certificaciones Agile Technical Tester o Agile Test Leadership at Scale. La rama de Especialista ofrece una inmersión profunda en áreas que tienen enfoques y actividades de prueba específicos, por ejemplo, en ingeniería de automatización de pruebas, pruebas de rendimiento, pruebas de seguridad, pruebas de IA (Inteligencia Artificial) y pruebas de aplicaciones móviles, o donde se requieren conocimientos específicos del dominio (por ejemplo, pruebas de software automatiz o pruebas de juegos). Por favor visite www.istqb.org para obtener la información más reciente sobre el esquema [de certificaciones del](#) ISTQB.

0.4 Resultados de Negocio

En esta sección se enumeran los resultados de negocio que se esperan de una persona que haya obtenido la certificación de Especialista en estrategia de automatización de pruebas.

Una persona Especialista en Estrategia de Automatización de Pruebas certificado puede:

TAS - BO - 01	Comprender los factores del software y los sistemas que influyen en el éxito de la automatización de pruebas.
TAS - BO - 02	Identificar costos y riesgos de implementar una solución de automatización de pruebas
TAS - BO - 03	Comprender los roles y las responsabilidades de las personas que contribuyen a la automatización de pruebas.
TAS - BO - 04	Plan para la integración de la automatización de pruebas en todos los niveles de prueba
TAS - BO - 05	Identificar consideraciones estratégicas para la implementación de la automatización de pruebas en diferentes modelos de ciclo de vida de desarrollo de software.
TAS - BO - 06	Comprender la aplicabilidad y viabilidad de la automatización de pruebas.
TAS - BO - 07	Planificar soluciones de automatización de pruebas que satisfagan las necesidades de la organización.
TAS - BO - 08	Comprender las estrategias de implementación de la automatización de pruebas
TAS - BO - 09	Comprender las dependencias de la automatización de pruebas dentro del entorno de prueba
TAS - BO - 10	Comprender los costos de configuración y mantenimiento de la automatización de pruebas
TAS - BO - 11	Descubrir qué métricas de automatización de pruebas ayudan a impulsar la toma de decisiones
TAS - BO - 12	Identificar formas en las que la automatización de pruebas aporta valor al proyecto y a la organización.

TAS - BO - 13	Identificar los requisitos de informes de automatización de pruebas para abordar las necesidades de las partes interesadas
TAS - BO - 14	Definir actividades de transición de pruebas manuales a automatización de pruebas.
TAS - BO - 15	Definir una estrategia de automatización de pruebas que garantice que los proyectos compartan activos y métodos para garantizar una implementación consistente en toda la organización.

0.5 Objetivos de Aprendizaje y Niveles de Conocimiento Evaluables

Los objetivos de aprendizaje respaldan los resultados comerciales y se utilizan para crear los exámenes de Especialista en Estrategia de Automatización de Pruebas.

En general, todos los contenidos de este programa de estudio pueden ser evaluados en los niveles K2 y K3, a excepción de la Introducción y los Apéndices. Es decir, se le puede pedir al/a la candidata/a que reconozca, recuerde o recupere una palabra clave o concepto mencionado en cualquiera de los seis capítulos. Los niveles de objetivos de aprendizaje específicos se muestran al inicio de cada capítulo, y se clasifican de la siguiente manera:

- K1: recordar.
- K2: entender.
- K3: aplicar.

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

Se deben recordar todos los términos que aparecen como palabras clave justo debajo de los títulos de los capítulos, incluso si no se mencionan explícitamente en los objetivos de aprendizaje.

0.6 El Examen de Certificación

El examen de certificación Especialista en Estrategia de Automatización de Pruebas 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 objeto de examen, excepto la Introducción y los Apéndices. Los estándares y libros se incluyen como referencias, pero su contenido no es evaluable, más allá de lo que se resume en el propio programa de estudio a partir de dichos estándares y libros.

Consulte el documento Estructuras y reglas de examen v1.1 Compatible con el programa de estudios de los niveles básico y avanzado y módulos especialistas para obtener más detalles sobre el examen de certificación de programa de estudios de estrategia de automatización de pruebas.

Es requerido para realizar el examen de certificación de Estrategia de Automatización de Pruebas, es que los candidatos tengan interés en las pruebas de software y la automatización de pruebas. Sin embargo, se recomienda enfáticamente que los candidatos también:

- Tener al menos un conocimiento mínimo en desarrollo de software y sistemas y liderazgo para la implementación de tecnología en la empresa y experiencia como ingeniero de pruebas senior, líder de pruebas o como desarrollador de software.
- Tome un curso que haya sido acreditado según los estándares ISTQB (por uno de los comités miembros reconocidos por ISTQB).

Nota: Se debe obtener el certificado ISTQB® de Nivel Básico antes de tomar el examen de certificación de Especialista Estrategia de Automatización de Pruebas.

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 siguen las "Accreditation Guidelines" generales publicadas por el "Processes Management and Compliance Working Group".

0.8 Tratamiento de Estándares

Existen estándares a los que se hace referencia en el programa de estudio básico (por ejemplo, estándares IEEE o ISO). Estas referencias proporcionan un marco (como en las referencias a la norma ISO 25010 relativa a las características de calidad) o para proporcionar una fuente de información adicional si así lo desea el lector. Los documentos estándar no están pensados para ser objeto de examen. Consulte el capítulo 7 para obtener más información sobre los estándares.

0.9 Actualización

La industria del software cambia rápidamente. Para hacer frente a estos cambios y proporcionar a las partes interesadas acceso a información relevante y actualizada, los grupos de trabajo de ISTQB han creado enlaces en el sitio web www.istqb.org, que remiten a documentos de apoyo y cambios en las normas. Esta información no se puede evaluar en el marco del programa de estudios de Especialista en Estrategia de Automatización de Pruebas.

0.10 Nivel de Detalle

El nivel de detalle de este programa de estudio permite la realización de cursos y exámenes consistentes a nivel internacional. Para lograr este objetivo, el programa de estudio se compone de:

- Objetivos didácticos generales que describen el propósito de Especialista en Estrategia de Automatización de Pruebas.
- Una lista de términos (palabras clave) que los alumnos deben ser capaces de recordar.
- Objetivos de aprendizaje para cada área de conocimiento, que describen los resultados cognitivos del aprendizaje que se deben alcanzar.
- Una descripción de los conceptos clave, incluyendo referencias a fuentes reconocidas.

El contenido del programa de estudios no es una descripción de toda el área de conocimiento de las pruebas de software; refleja el nivel de detalle que se cubrirá en los cursos de capacitación de Especialista en Estrategia de Automatización de Pruebas. Se centra en los conceptos y técnicas de prueba que se pueden aplicar a todos los proyectos de software, incluidos los que siguen métodos ágiles. Este programa de estudios no contiene ningún objetivo de aprendizaje específico relacionado con las pruebas ágiles, pero sí analiza cómo se aplican estos conceptos en proyectos ágiles y otros tipos de proyectos.

0.11 Organización de este Programa de Estudio

Hay seis capítulos con contenido que se puede evaluar en el examen oficial. El encabezado de nivel superior de cada capítulo especifica el tiempo dedicado al capítulo; no se proporciona información sobre el tiempo en el nivel inferior del capítulo. Para los cursos de formación acreditados, el programa de estudios requiere un mínimo de 12,75 horas de instrucción, distribuidas en los seis capítulos de la siguiente manera:

- Capítulo 1: 45 minutos – Introducción y objetivos de la estrategia de automatización de pruebas
 - Se comprende los conceptos de automatización de pruebas y aprende los criterios de selección para proyectos candidatos.
 - Se comprende los factores que definen una implementación exitosa de la automatización de pruebas.
- Capítulo 2: 60 minutos – Recursos de automatización de pruebas
 - Se aprende las diferentes soluciones que están disponibles para la automatización de pruebas y la inversión relativa para cada una.
 - Se cubre el licenciamiento de software para herramientas de automatización de pruebas
 - Se comprenden las habilidades necesarias para la automatización de pruebas.
- Capítulo 3: 225 minutos – Preparación para la automatización de pruebas
 - Se aprende cómo se utiliza la automatización de pruebas en todos los niveles de prueba y dentro de ellos.
 - Se cubren estrategias de automatización de pruebas para distribuir adecuadamente las pruebas y lograr cambios a la izquierda y a la derecha.
 - Se aprende cómo la automatización de pruebas respalda los proyectos tradicionales y ágiles
 - Se cubre la automatización de pruebas dentro de DevOps y las prácticas de pruebas continuas.
 - Se comprende cómo definir criterios para el uso de la automatización de pruebas, incluidas las pruebas más adecuadas para la automatización de pruebas.
- Capítulo 4: 135 minutos – Estrategias de implementación y lanzamiento organizacional para la automatización de pruebas
 - Se aprende cómo la automatización de pruebas puede mejorar el tiempo de comercialización
 - Se comprende cómo desarrollar pruebas automatizadas operativamente relevantes y generar informes de defectos.
 - Se cubre la estrategia de implementación de automatización de pruebas y la mitigación de riesgos.
 - Se aprende sobre el entorno de automatización de pruebas y sus dependencias.
 - Se cubre la integración de la automatización de pruebas y los datos de prueba a un sistema bajo prueba.
- Capítulo 5: 150 minutos – Análisis del impacto de la automatización de pruebas
 - Se aprende cómo lograr un retorno de la inversión para la automatización de pruebas
 - Se cubren los objetivos para que una organización y un proyecto utilicen la automatización de pruebas.

- Se aprende a analizar informes de pruebas e informar a los tomadores de decisiones de una manera clara y comprensible.
- Capítulo 6: 150 minutos – Estrategias de implementación y mejora para la automatización de pruebas
 - Se aprende cómo pasar de la prueba manual a la automatización de pruebas y a la prueba continua.
 - Se cubre la evaluación de la automatización de pruebas para la mejora continua.

1 Introducción y objetivos de la estrategia de automatización de pruebas– (K2)

Palabras clave:

Arquitectura de automatización de pruebas, marco de automatización de pruebas, estrategia de automatización de pruebas

Objetivos de Aprendizaje para el Capítulo 1

1.1 Factores de éxito en un proyecto de automatización de pruebas

CT-TAS -1.1.1	(K2)	Explicar los objetivos y la relevancia de la automatización de pruebas.
CT-TAS -1.1.2	(K2)	Identificar los factores técnicos de éxito técnico en un proyecto de automatización de pruebas.
CT-TAS -1.1.3	(K2)	Resumir los criterios de inversión adecuados para seleccionar proyectos candidatos para la automatización de pruebas.

1.1 Factores de éxito de un proyecto de automatización de pruebas

1.1.1 Definir metas y objetivos para la estrategia de automatización de pruebas

Al definir la estrategia de un proyecto de automatización de pruebas, se deben abordar los siguientes aspectos:

- Definición del propósito previsto
- Identificación de riesgos
- Definición del alcance
- Identificación de los interesados involucrados
- Selección de herramientas para la automatización
- Diseño de la arquitectura de automatización de pruebas (AAP)
- Identificación de entornos

Los objetivos de la automatización de pruebas pueden incluir:

- Mejorar la eficiencia de las pruebas.
- Proporcionar una cobertura más amplia y profunda.
- Mejora de la calidad general del SSP
- Reducción del coste total y del tiempo de comercialización
- Realizar pruebas que los probadores manuales no pueden realizar
- Reducción del tiempo de ejecución de pruebas
- Aumento de la frecuencia de las pruebas

Dado que el desarrollo de software ágil ofrece ciclos de implementación más rápidos y las aplicaciones en la nube están más generalizadas, el desarrollo requiere una retroalimentación temprana y rápida sobre la calidad de los SSP. Como resultado de este cambio, la automatización de pruebas tiene más enfoque y relevancia en los proyectos modernos, dada su naturaleza y los objetivos de las pruebas.

1.1.2 Identificar los factores técnicos de éxito en un proyecto de automatización de pruebas

Los siguientes factores de éxito se aplican a proyectos de automatización de pruebas que se enfocan en los factores que afectan el éxito a largo plazo de un proyecto. El uso de un proyecto piloto ayuda a determinar las herramientas y la viabilidad de las tecnologías seleccionadas.

Los factores de éxito para la automatización de pruebas incluyen los siguientes:

- Capacidad de ser probado del SSP
 - Permite que la automatización de pruebas acceda a las interfaces SSP
- Estrategia de automatización de pruebas definida
 - La estrategia debe ser aplicable y personalizable; sus objetivos deben ser alcanzables dentro de las limitaciones de tiempo y costo, y debe mantenerse actualizada.
- AAP
 - Claridad de qué y cómo implementarla.
- Para obtener información adicional, consulte el programa de estudio CTAL-TAE, sección 3.1.1.
- Marco de automatización de pruebas (MTAP)

- Un MTAP fácil de usar, bien documentado y mantenible, que admita un enfoque coherente para automatizar las pruebas. Para obtener información adicional, consulte el programa de estudio CTAL-TAE, sección 3.1.3.
- Informes de pruebas definidos e implementados
- Fácil resolución de problemas
- Entorno de prueba apropiado
- Casos de prueba automatizados documentados
- Trazabilidad de los casos de prueba automatizadas con la definición de casos de prueba y los requerimientos.
- Fácil mantenimiento
- Pruebas automatizadas actualizadas
- Un plan de despliegue claro.
- Un plan de ejecución de pruebas claro
- Pruebas retiradas según sea necesario
- Manejo efectivo de excepciones

Antes de iniciar un proyecto de automatización de pruebas, es importante analizar las posibilidades de éxito del proyecto teniendo en cuenta los factores existentes y los factores faltantes, teniendo en cuenta los riesgos del enfoque de prueba elegido y el contexto del proyecto. No todos los factores son necesarios y, en la práctica, rara vez se cumplen todos.

1.1.3 Resumir los criterios de inversión adecuados para seleccionar proyectos candidatos a la automatización de pruebas

La implementación de la automatización de pruebas en un proyecto implica inversión y, en la mayoría de los casos, un coste significativo. Esto debe tenerse en cuenta, junto con la naturaleza del proyecto y si se debe utilizar la automatización de pruebas en el proyecto.

Considere los siguientes criterios de inversión antes de introducir la automatización de pruebas:

- Costo de introducción: además del trabajo necesario para configurar la automatización de pruebas, se requerirán costos adicionales para el proyecto, ya que puede ser necesario contratar nuevos ingenieros de automatización de pruebas (IAP), comprar nuevo hardware o iniciar capacitaciones.
- Fase actual en el ciclo de vida de desarrollo de software (CVDS) del proyecto: es mejor comenzar lo antes posible para que la automatización de pruebas pueda aportar más valor pronto.
- Duración esperada/planificada del proyecto de desarrollo de software: para un proyecto más corto, es posible que no haya suficientes recursos para comenzar o tiempo restante para que la automatización de pruebas aporte valor.
- Costo de mantenimiento: En el caso de comenzar desde cero, configurar una solución de automatización de pruebas (SAP) llevará tiempo junto con la necesidad de mantenimiento.

Si la inversión que supone introducir la automatización de pruebas en un proyecto es aceptable, se puede empezar con la preparación para la automatización de pruebas. Esto incluye la selección del enfoque de pruebas y las herramientas de automatización de pruebas adecuadas. La responsabilidad principal de este proceso recae en un arquitecto o jefe de pruebas, que tiene los conocimientos necesarios sobre automatización de pruebas para tomar decisiones relevantes.

2 Recursos de automatización de pruebas: 60 minutos (K2)

Palabras clave:

Ingeniero de automatización de pruebas, Solución de automatización de pruebas

Objetivos de Aprendizaje para el Capítulo 2

2.1 Costos y riesgos de implementar una solución de automatización de pruebas

CT-TAS -2.1.1	(K2)	Comparar soluciones técnicas alternativas con respecto al costo de propiedad.
CT-TAS -2.1.2	(K2)	Explicar las consideraciones del modelo de licencia para herramientas de automatización de pruebas.
CT-TAS -2.1.3	(K2)	Proporcionar ejemplos de factores a tener en cuenta al definir una estrategia de automatización de pruebas.

2.2 Funciones y responsabilidades dentro de la automatización de pruebas

CT-TAS -2.2.1	(K2)	Resumir los roles y habilidades necesarios para una solución exitosa de automatización de pruebas.
---------------	------	--

2.1 Recursos de automatización de pruebas: 60 minutos (K2)

2.1.1 Comparar soluciones técnicas alternativas en relación con el costo de propiedad

En términos de propiedad, un enfoque común es una solución interna personalizada basada en herramientas comerciales o de código abierto. Otros enfoques alternativos incluyen soluciones comerciales no personalizables o la firma de un contrato con una empresa subcontratada para que realice el trabajo del IAP.

La creación de toda la SAP internamente garantiza que todos los costos, recursos, riesgos y gobernanza estén dentro de la organización (por ejemplo, miembros del equipo/desarrolladores y los recursos de hardware y software que son necesarios para la solución actual). Un factor clave es que, al seguir este enfoque y asignar tiempo a esta actividad, la SAP se puede desarrollar sin necesidad de un contrato con un proveedor o una empresa de subcontratación, ya que todos los IAP necesarios y sus conocimientos están disponibles dentro de la organización. Sin embargo, para llevarlo a cabo con éxito, la organización necesita tener los IAP adecuados, contratados o capacitados, que puedan impulsar el desarrollo de la SAP.

Al trabajar con una solución basada en un proveedor, la propiedad será compartida entre el cliente y el proveedor, dependiendo de los detalles de su contrato. Si ya existe una herramienta de prueba en la organización que ya se ha probado mediante un proyecto piloto y cumple con todos los requisitos, y no hay necesidad de características adicionales en la herramienta de prueba, este enfoque será más fácil de adoptar. Los probadores dentro de la organización podrán trabajar de manera productiva una vez que reciban la capacitación necesaria del proveedor, y generalmente habrá Expertos en la Materia (SME por sus siglas en inglés o EM por sus siglas en español) asignados como propietarios de productos dentro de la organización. El riesgo con este enfoque es que si se necesita realizar algún trabajo adicional (por ejemplo, corregir defectos de la herramienta y solicitar características adicionales) en la herramienta de prueba, llevará tiempo y requerirá negociaciones con el proveedor para que se haga a tiempo y de la manera correcta, lo que puede afectar el trabajo de los probadores que están trabajando con esta herramienta de prueba.

Si la organización no quiere tener la responsabilidad de crear un equipo, la subcontratación es una solución recomendada. En el caso de trabajar con empresas subcontratadas, el cliente no tiene que contratar ni comprar ningún hardware o software, ni contratar empleados con las habilidades requeridas. Todo el trabajo y los costos adicionales correrán a cargo de la empresa subcontratada, junto con la propiedad de las herramientas. Las expectativas y métricas mensurables deben definirse en el contrato para que el progreso sea transparente y visible. Este enfoque se sugiere en caso de que la organización no planee invertir esfuerzos en la contratación de IAP debido a plazos de proyecto cortos, costos u otras consideraciones.

2.1.2 Explicar las consideraciones del modelo de licencia para herramientas de automatización de pruebas

La concesión de licencias es un aspecto importante a tener en cuenta cuando se establece la automatización de pruebas. Cada modelo de concesión de licencias mencionado a continuación tiene un impacto diferente en la usabilidad y los factores de coste, por ejemplo, los costes de ejecución de pruebas y la dificultad para configurar el entorno de desarrollo.

Los modelos de licencia incluyen:

- **Código abierto:** En muchos casos, las organizaciones utilizan herramientas de código abierto para lograr sus objetivos de automatización de pruebas. La razón principal es que no hay costos

de licencia ni tarifas de mantenimiento por el uso de las herramientas de prueba. Muchos usuarios y organizaciones contribuyen al desarrollo de herramientas de código abierto, lo que facilita la recopilación de información y la recepción de soporte. La mayoría de las licencias de código abierto también permiten a las personas modificar las herramientas si es necesario o incluso volver a publicarlas sin restricciones significativas.

- **Licencia por usuario/máquina:** Las herramientas comerciales suelen tener licencias por usuario o máquina. En términos de costos, las herramientas se pueden usar de manera eficiente cuando la organización conoce la cantidad de asistentes técnicos que trabajarán en un proyecto determinado a largo plazo. A menudo, cuantas más licencias solicita una organización, mejor es el precio que recibe.
- **Licencia flotante:** Se trata de una licencia que se puede compartir entre varias personas de la organización para utilizarla en diferentes momentos. Puede resultar muy útil cuando hay muchos IAP que trabajan en diferentes máquinas. La cantidad de licencias se calcula en función del uso simultáneo, no de la cantidad total de IAPs o de las máquinas específicas en las que ejecutan sus pruebas.
- **Licencia de tiempo de ejecución:** Muchas herramientas comerciales tienen licencias de tiempo de ejecución para ejecutar la automatización de pruebas. Este tipo de licencia se da con proveedores de servicios en la nube que alojan herramientas de automatización de pruebas, múltiples sistemas operativos (SO) y versiones de navegador como servicio. También hay proveedores de servicios en la nube que brindan acceso a granjas de dispositivos que tienen una variedad de dispositivos móviles, plataformas, versiones y redes celulares. A quienes utilizan estos servicios solo se les cobra por el tiempo que utilizan para ejecutar pruebas y, por lo tanto, se les cobra en función de la licencia de tiempo de ejecución.

2.1.3 Proporcionar ejemplos de factores a tener en cuenta al definir una estrategia de automatización de pruebas

Muchos factores pueden influir en las decisiones sobre la implementación de la automatización de pruebas y su estrategia.

Los factores más cruciales son:

- Limitaciones de tiempo
- Nivel de experiencia necesario y número de IAP's para desarrollar la SAP
- Prueba de hardware
- Licencias de herramientas de prueba
- Adaptabilidad
- Mantenimiento
- Soporte para diferentes plataformas (por ejemplo, Web, escritorio y/o dispositivos móviles)
- Soporte de integración continua/entrega continua (CI/CD, por sus siglas en inglés)
- Gestión de pruebas de Integración y generación de informes de pruebas

El primer y más importante factor de costo es el cronograma para el trabajo que debe realizarse y para la automatización de la prueba en sí.

Desde una perspectiva de tiempo, si el plazo es corto, un enfoque que se suele utilizar es contratar solo unos pocos IAP capacitados para crear la SAP. En términos de una hoja de ruta y un cronograma más largos, la organización puede decidir la cantidad de IAP's necesarios con más contexto. Cuando el SSP crece, mejora y se vuelve más complejo, se puede tomar la decisión de contratar más IAP's para implementar y mantener la SAP y las pruebas.

Otros factores de costo importantes son las herramientas y sus licencias. El presupuesto debe incluir las herramientas de prueba, el hardware y la capacitación adicional necesarios para los IAP's. Estos factores tienen una fuerte conexión con la integración con otras herramientas y sistemas para realizar funciones adicionales que no sean de prueba, como cargar los resultados de las pruebas al sistema de gestión de pruebas o activar una compilación en el sistema de gestión de configuración. Para estas funciones adicionales existen herramientas, ya sean gratuitas o de pago, y librerías para usar, pero estas deben considerarse durante la creación de la estrategia de automatización de pruebas y deben incluirse en el presupuesto.

Siempre depende del contexto cuántos entornos de prueba y agentes tiene un equipo de desarrollo. Cuanto mayor y más complejo sea el SSP (sistema bajo prueba) y el calendario de lanzamientos o pases a producción, más recursos requerirá la organización, lo que también aumentará el costo. Un enfoque reciente para limitar los costos de recursos utiliza proveedores de la nube y paga por los recursos de hardware de prueba y las licencias de ejecución bajo demanda, lo cual debe usarse con cuidado, ya que puede tener efectos contraproducentes. A veces, las máquinas pueden quedar en funcionamiento y volverse más caras que su propio hardware. Este enfoque puede reducir los altos costos de mantenimiento y operación para el equipo de IAP's.

2.2 Funciones y responsabilidades en la automatización de pruebas

2.2.1 Resumir los roles y las habilidades necesarias para una solución de automatización de pruebas exitosa

Para un SAP (Solución de Automatización de Pruebas) exitoso, las organizaciones requieren IAP's (Ingenieros de Automatización de Pruebas) calificados que deben tener un sólido conocimiento en programación y arquitectura técnica.

Dependiendo del tamaño y madurez del proyecto, también debería haber al menos un EM (experto en la materia) sólido (por ejemplo, un líder de pruebas, arquitecto o analista de negocios) que entienda el dominio empresarial real y los objetivos de la prueba. El papel de esta persona es ayudar a impulsar y crear el concepto, basado en los objetivos de la prueba, y un plan para la SAP real. Se necesitarán habilidades de gestión de equipos y habilidades blandas para capacitar, motivar y construir el equipo para el trabajo real. [CTEL-TM-MTT]

Un IAP debe tener sólidas habilidades técnicas y conocimiento sobre diferentes ciclos de vida del desarrollo de software (CVDS), así como sobre la arquitectura del SSP (sistema bajo prueba) y su entorno de desarrollo. Aparte de los aspectos técnicos, un IAP debe tener la capacidad de cooperar con analistas de pruebas y otros interesados sobre los objetivos de la automatización de pruebas. Dado que las pruebas exhaustivas no son alcanzables, lo mismo aplica para la automatización de pruebas: no es posible alcanzar una cobertura del 100% en la automatización de pruebas. Dado que el tiempo y el esfuerzo son limitados, los IAP's deben ser capaces de priorizar las condiciones de prueba más impactantes desde una perspectiva empresarial e inversora, contribuyendo a las evaluaciones de riesgos.

3 Preparación para la automatización de pruebas: 225 minutos (K3)

Palabras clave

Pruebas de API, Pruebas de componente, pruebas de contrato, desplazamiento a la izquierda (shift left), desplazamiento a la derecha (shift right), sistema bajo prueba, enfoque de automatización de pruebas, condición de prueba, doble prueba, nivel de prueba, pirámide de pruebas

Objetivos de Aprendizaje para el Capítulo 3

3.1 Integración en todos los niveles de pruebas

CT-TAS -3.1.1	(K2)	Diferenciar entre distribuciones de automatización de pruebas.
CT-TAS -3.1.2	(K2)	Seleccionar un enfoque de automatización de pruebas basado en la arquitectura del sistema bajo prueba.
CT-TAS -3.1.3	(K3)	Demostrar formas de optimizar la distribución de la automatización de pruebas para lograr enfoques de desplazamiento a la izquierda (shift left) y desplazamiento a la derecha (shift right).

3.2 Consideraciones estratégicas en diferentes modelos de ciclo de vida del desarrollo de software

CT-TAS -3.2.1	(K2)	Explicar cómo los proyectos de automatización de pruebas se ajustan a los modelos de ciclo de vida de desarrollo de software heredados
CT-TAS-3.2.2	(K2)	Explicar cómo los proyectos de automatización de pruebas se ajustan a las mejores prácticas de desarrollo de software ágil que soportan la automatización de pruebas.
CT-TAS-3.2.3	(K3)	Prepararse para que los proyectos de automatización de pruebas se ajusten a las mejores prácticas de DevOps que apoyan la automatización de pruebas en pruebas continuas.

3.3 Aplicación y viabilidad de la automatización de pruebas

CT-TAS-3.3.1	(K2)	Explicar los criterios para determinar la idoneidad de las pruebas para la automatización de pruebas
CT-TAS-3.3.2	(K2)	Identificar los retos que sólo la automatización de pruebas puede abordar
CT-TAS-3.3.3	(K2)	Identificar las condiciones de prueba difíciles de automatizar

3.1.1 Diferenciar las distribuciones de automatización de pruebas

Mike Cohn planteó el concepto original de una pirámide de automatización de pruebas que describe tres niveles: unidad (el ISTQB nombra a este nivel componente), servicio y UI. Desde entonces, han aparecido muchas variaciones, y todas comparten los mismos principios básicos de describir el papel de los niveles de prueba y la distribución de los casos de prueba a través de esos niveles (ver el ISTQB CTFL Syllabus, sección 5.1.6 Pirámide de Prueba).

La pirámide de automatización de pruebas original de Mike Cohn se centra en los tipos de pruebas que realiza el SAP. El nivel de servicio puede desglosarse en tres tipos de pruebas: pruebas de integración de componentes, pruebas de contratos y pruebas de API.

Las pruebas unitarias (el ISTQB las denomina pruebas de componentes) sirven para validar componentes individuales, centrándose en la calidad del código. Las pruebas de integración de componentes permiten validar la interfaz de usuario (IU) y la API mediante dobles de prueba, como mocks y stubs. Las pruebas de contratos permiten validar los contratos entre servicios. Las pruebas de API se centran en la validación funcional de determinados servicios con datos reales a través de conexiones de servicios reales. Las pruebas de interfaz de usuario son pruebas de extremo a extremo de un sistema, interactuando con su interfaz gráfica de usuario (IGU).

Resulta útil trazar el estado actual de las pruebas como referencia y el estado objetivo. De este modo, se entiende claramente lo que falta o lo que constituye un nivel aceptable de pruebas. Esto indicará la cantidad de pruebas realizadas en cada nivel de prueba y si se trata de pruebas manuales o automatizadas. El estado objetivo también depende del calendario y de lo que se pueda conseguir en ese plazo. Si es factible alcanzarlo, entonces la forma piramidal debería ser la forma objetivo.

Ejemplos de distribuciones de prueba:

- **Pirámide:** Se trata de una distribución equilibrada de las pruebas, en la que se realizan menos pruebas en los niveles superiores y más en los inferiores, con pruebas estables y más rápidas. Si los recursos disponibles y el marco temporal lo permiten, esta suele ser la pirámide del estado objetivo.
- **Cono de helado:** Esta es la versión invertida de la pirámide. Hay una cantidad equilibrada de pruebas de servicios, pero las pruebas dependen en gran medida de encontrar la mayoría de los defectos en el nivel de pruebas de interfaz de usuario, que suele ser más costoso de automatizar debido a su complejidad. Debido a la falta de pruebas de componentes, los defectos se encuentran más tarde en el CVDS.
- **Reloj de arena:** Las pruebas se concentran en los niveles más alto y más bajo, y la mayoría de las veces no se realizan pruebas a nivel de servicio, lo que provoca defectos de integración. Si la lógica empresarial la proporcionan las API (es decir, los servicios), muchas de las pruebas de interfaz de usuario pueden trasladarse fácilmente a niveles de prueba inferiores.
- **Paraguas:** Las pruebas dependen completamente de las costosas pruebas de interfaz de usuario. Esto se traduce en una lenta respuesta a los defectos, un costoso mantenimiento de los casos de prueba y del SAP. Si no es técnicamente posible implementar casos de prueba de nivel inferior, entonces alejarse de la forma paraguas podría no ser alcanzable, y el foco debería estar en

optimizar la suite de automatización de pruebas de interfaz de usuario, reducir el tiempo de ejecución de pruebas y mejorar la estabilidad.

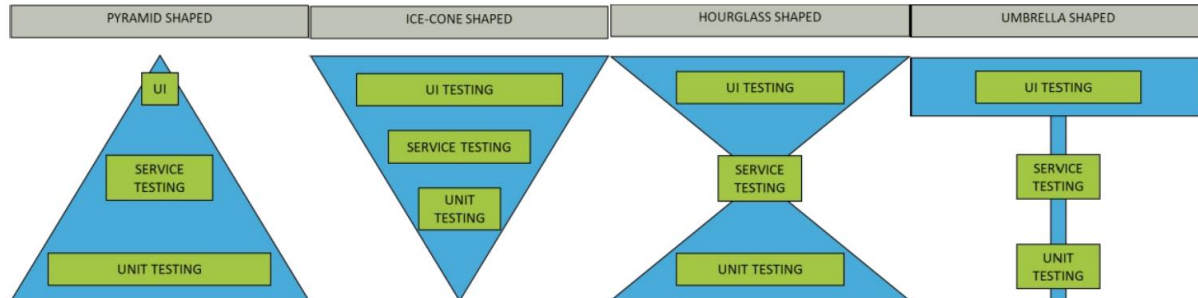


Figura 1: Ejemplos de distribuciones de prueba

3.1.2 Seleccionar un enfoque de automatización de pruebas basado en la arquitectura del sistema bajo prueba

Hay muchas maneras de definir los niveles de prueba en una estrategia de prueba. Cuál elegir depende en gran medida de varios factores, como la cultura de la organización, la madurez general de la ingeniería de software, el SDLC seguido y si el SUT tiene una arquitectura de mainframe o de microservicios. Aunque la forma piramidal se considera la distribución de prueba ideal, no siempre es alcanzable, o se tarda más tiempo en pasar a ese estado final. Es una buena práctica tener una forma realista como estado objetivo, como la transformación de la forma de paraguas a la de reloj de arena. Una vez conseguido, se puede determinar un nuevo estado objetivo. La elección de la distribución de automatización de pruebas adecuada para los sistemas mainframe difiere del desarrollo de software moderno y, de nuevo, depende de muchos factores, ya que algunos de los niveles podrían no ser posibles de automatizar en absoluto. El acceso a los mainframes se realiza tradicionalmente mediante emulación de terminal (es decir, pantallas verdes). La validación de los procesos por lotes es posible, pero limitada. Las pruebas de componentes son más difíciles de realizar. Dado que la presencia de microservicios es escasa o nula, no siempre es posible validar la comunicación de datos entre interfaces mediante pruebas de API. Son más frecuentes las pruebas mediante trabajos por lotes, bases de datos y una interfaz gráfica de usuario. Las organizaciones que están modernizando sus soluciones heredadas, cambiando lentamente hacia una arquitectura de microservicios, pueden introducir pruebas de API y pruebas de contratos en las partes modernizadas del sistema.

3.1.3 Demostrar formas de optimizar la distribución de automatización de pruebas para lograr enfoques de desplazamiento a la izquierda y desplazamiento a la derecha

Una vez identificada la distribución del estado actual y seleccionada la distribución del estado objetivo, puede planificarse una hoja de ruta de mejoras. Esto determina qué probar con la automatización (es decir, el alcance de la automatización de pruebas) y cómo probar (es decir, la estrategia de automatización de pruebas). Hay que establecer un orden de prioridad de los elementos que deben aplicarse y los criterios de entrada para la automatización de las pruebas. En caso de una distribución desequilibrada de la automatización de pruebas, se recomienda adoptar un enfoque ascendente. Si la cobertura del código es baja, es señal de que faltan casos de prueba de componentes, y hay que escribir casos de prueba de componentes adicionales. Como segundo paso, hay que revisar la calidad de las pruebas de componentes y, si es necesario, mejorarlas. Sugerir las mejores prácticas, seguir una técnica de desarrollo basada en pruebas y celebrar talleres sobre técnicas de prueba pueden mejorar la calidad de la automatización de las pruebas.

La introducción de pruebas de componentes para pruebas de interfaz de usuario y API, aprovechando los dobles de prueba (por ejemplo, mocks, stubs) ayuda a lograr un cambio a la izquierda de pruebas costosas, lentas y poco fiables. La eliminación de la dependencia de servicios y datos reales mejora la coherencia de la ejecución de las pruebas y proporciona información temprana que es fácil de integrar en las canalizaciones de CI/CD.

En los últimos años, las pruebas de contratos han sido cada vez más prominentes. Al validar el contrato entre un proveedor y un consumidor, es más fácil encontrar antes las causas raíz de los defectos. Los equipos no dependen de las pruebas de API, ni de las pruebas de interfaz de usuario para detectar defectos de los servicios subyacentes, y en su lugar pueden disminuir el número de casos de prueba de este tipo. Crear un gráfico de llamadas para las API es una forma inteligente de rastrear qué API están conectadas entre sí y cuáles son las consumidoras o productoras de una API seleccionada. Esto permite identificar mejor los posibles puntos débiles de un sistema.

Mientras que el desplazamiento a la izquierda es un enfoque que implica mover las pruebas en una fase más temprana del SDLC, el desplazamiento a la derecha mueve las pruebas más tarde, cuando el SBP ha sido liberado, como medio para evaluar el rendimiento en un entorno de pruebas de preproducción o en producción. Al aplicar el desplazamiento a la derecha, los responsables de las pruebas pueden supervisar el rendimiento de la aplicación y la API mientras al mismo tiempo se obtienen comentarios de los usuarios reales. Aunque la automatización de las pruebas es más importante en un enfoque de desplazamiento a la izquierda, si se combina con la capacidad de observación, la automatización de las pruebas puede ayudar a tomar decisiones más rápidas sobre si se puede seguir adelante con la versión completa o si es necesario revertir la versión candidata.

Las pruebas de desplazamiento a la derecha tienen como objetivo:

- Comprender las preferencias de los usuarios
- Soporta lanzamientos canarios y lanzamientos oscuros para interrumpir mínimamente la funcionalidad del usuario.
- Detección precoz de defectos en la producción
- Ayudar a ampliar el alcance y el uso de la automatización de pruebas
- Aumentar la cobertura

3.2 Consideraciones estratégicas en los distintos modelos de ciclo de vida del desarrollo de software

3.2.1 Explicación de la conformidad de los proyectos de automatización de pruebas con los modelos de ciclo de vida de desarrollo de software heredados

En el modelo de cascada, la fase de pruebas viene después de las fases de análisis de requisitos, diseño del sistema e implantación. Debido a ese estricto orden, las actividades de automatización de pruebas empiezan más tarde. Ciclos más largos entre las pruebas significa menos oportunidades para aprovechar la automatización de pruebas, y la retroalimentación de la automatización de pruebas por lo general llega demasiado tarde, lo que resulta en un menor retorno de la inversión (ROI, por sus siglas en inglés).

En el modelo V, toda la planificación de las pruebas y la preparación de la documentación se realizan en las primeras fases. Por ejemplo, la fase de diseño de la arquitectura produce el diseño de la prueba de integración, que establece las pruebas antes que en la cascada. Desgraciadamente, la implementación real de un SAP sigue produciéndose más tarde en el CVDS y, aunque el ROI de la automatización de pruebas es mayor en comparación con el modelo en cascada, sigue estando por detrás de las prácticas modernas de desarrollo ágil de software.

3.2.2 Explicar cómo los proyectos de automatización de pruebas se ajustan a las mejores prácticas de desarrollo de software ágil que admiten la automatización de pruebas

Uno de los ideales de seguir un desarrollo de software ágil es lograr la automatización de las pruebas durante el proceso. Esto significa determinar todas las actividades de automatización de pruebas necesarias como parte de los criterios de salida para cada una de las historias de usuario. Incluye las definiciones de los casos de prueba, la implementación de los casos de prueba automatizados, las actualizaciones del FAP y, en algunos casos, la integración en un canal de CI/CD. Las organizaciones que no aplican correctamente las prácticas ágiles no incluyen una estimación del esfuerzo de las pruebas y las administran en un ticket aparte o no las supervisan en absoluto. Al lograr la automatización de pruebas en el sprint, los equipos ágiles se aseguran de estar listos para desplegar el alcance acordado dentro o al final de cada sprint. Si un equipo no está maduro desde el punto de vista ágil, el primer objetivo debe ser lograr las pruebas en el sprint, mientras que la automatización se retrasaría un sprint. A partir de ahí, un equipo puede trabajar para conseguir la automatización de las pruebas en el sprint.

3.2.3 Prepárese para que los proyectos de automatización de pruebas se ajusten a las mejores prácticas de DevOps para lograr pruebas continuas

El desarrollo ágil de software se centra en cómo se organiza el trabajo, mientras que DevOps es responsable de la entrega integral del software. Esto se consigue automatizando las actividades de creación, integración, prueba, despliegue y producción. Esto facilita las pruebas continuas a través de bucles de retroalimentación que garantizan la mejora continua.

Se hace más hincapié en la aplicación de casos de prueba automatizados de nivel inferior, incluidas las pruebas de componentes, integración de componentes y contratos. Al reducir la dependencia de datos y servicios reales, las pruebas se ejecutarán en menos tiempo. Si es factible, la automatización de las pruebas debe ejecutarse en la misma canalización en la que se construye el SUT. Se activan diferentes conjuntos de pruebas después de cada fase de desarrollo y construcción (por ejemplo, local, pull request, merge y deploy).

Si los encargados de las pruebas tienen capacidad para crear conjuntos de pruebas de interfaz de usuario y API más amplios, entonces se ejecutan por separado para aportar valor adicional. Se sugiere que las pruebas manuales se centren en pruebas exploratorias y comentarios de los usuarios finales, como actividad complementaria a la automatización de las pruebas

3.3 Aplicabilidad y viabilidad de la automatización de pruebas

3.3.1 Explicación de los criterios para determinar la conveniencia de las pruebas para la automatización de pruebas

La selección de casos de prueba para su automatización suele correr a cargo de un analista de pruebas que sabe qué casos de prueba pueden automatizarse, o bien de un TAE con los conocimientos necesarios para tomar esas decisiones.

Los siguientes aspectos se tienen en cuenta a la hora de seleccionar y priorizar los casos de prueba para la automatización de pruebas:

- ¿Es técnicamente posible aplicar los casos de prueba de forma automatizada?

- ¿Existe algún problema técnico que afecte a la entrega de casos de prueba automatizados?
- ¿Está el equipo preparado y bien formado para realizar el trabajo de implantación?
- ¿Proporciona el esfuerzo de codificación una rentabilidad adecuada? (véase el apartado 5.1.1)
- ¿Vale la pena ejecutar los casos de prueba con frecuencia?
- ¿Es una prueba funcional o no funcional? ¿Forma parte del conjunto de pruebas de humo, de regresión o de confirmación?
- ¿Se puede repetir el caso de prueba?
- ¿Es fácil mantener el caso de prueba cuando el SUT cambia debido a actualizaciones?
- ¿El caso de prueba cubre los flujos del negocio utilizados con frecuencia?
- ¿Existe un solapamiento funcional entre las pruebas que permita reutilizar los pasos y los datos de las pruebas?

3.3.2 Identifique los retos que sólo la automatización de pruebas puede abordar

Hay ciertas pruebas que sólo pueden llevarse a cabo con la automatización de pruebas. Estas incluyen categorías cuando:

1. El tiempo de ejecución manual de las pruebas es superior al adecuado
2. Es necesario sincronizar la ejecución de los casos de prueba
3. Los resultados de las pruebas deben estar disponibles en un canal de distribución
4. Es necesario analizar grandes archivos de registro para detectar defectos
5. Se requiere precisión en los tiempos de las pruebas
6. Se requieren permutaciones de pruebas en varios sistemas operativos, navegadores, dispositivos, ubicaciones o configuraciones.
7. Se requiere un gran volumen de ejecuciones de pruebas y/o entrada de datos para maximizar la cobertura.
8. Cualquier prueba no funcional que requiera supervisión y análisis automatizados, o la aportación de una gran cantidad de usuarios, como pruebas de estrés o de fiabilidad.

3.3.3 Identificar condiciones de prueba difíciles de automatizar

Hay ciertas condiciones de prueba que hacen que la automatización de casos de prueba sea un reto difícil o incluso una tarea imposible. Hay muchos ejemplos de la vida real. A continuación, se enumeran algunos de ellos:

- Validación de los requisitos de diseño, incluida la coherencia de la interfaz de usuario en distintas plataformas. El aspecto general del software es subjetivo y requiere una revisión humana.
- Cualquier caso de prueba que implique demasiada interacción humana. En el sector financiero, un buen ejemplo sería la solicitud de un préstamo. En este proceso puede haber casos de ciertas normas o condiciones (por ejemplo, ingresos insuficientes y datos personales incorrectos). En tales situaciones, los agentes tienen que volver a comprobar la solicitud de préstamo real y tomar decisiones manuales o ponerse en contacto con el cliente.

- Dificultades técnicas que bloquean las actividades de automatización de pruebas, como las restricciones a nivel de SO. Un ejemplo es el software nativo del SO que envía mensajes de texto a los usuarios. Las interacciones o validaciones de esos mensajes de texto no son posibles con las herramientas de automatización de pruebas de interfaz de usuario, ya que el sistema operativo no permite interacciones fuera del SUT de destino.
- Las condiciones de prueba con una dependencia temporal prolongada harían que la automatización de pruebas fuera ineficaz y, a menudo, difícil de configurar correctamente en comparación con la ejecución de pruebas manuales. Por ejemplo, el responsable de las pruebas tiene que iniciar sesión en el SUT, actualizar el contenido de la página de inicio y esperar dos horas a que el SUT cierre la sesión automáticamente debido a un tiempo de espera de la sesión. Si el tiempo transcurrido no puede modificarse de forma manual (p. ej., simulacros, respuestas del lado del servidor y cambios de tiempo a nivel del sistema operativo), no se recomienda implementar estos casos de prueba de forma automatizada.

4 Estrategias de implementación y lanzamiento organizacional para la automatización de pruebas (K2)

Duración: 135 minutos

Palabras Clave: componente, prueba de confirmación, Punto de Control de Calidad (quality gate)

Objetivos de Aprendizaje para el Capítulo 4

4.1 Planificación de soluciones de automatización de pruebas

CT-TAS-4.1.1	(K2)	Identificar formas en que la automatización de pruebas permite acortar el tiempo de comercialización.
CT-TAS-4.1.2	(K2)	Identificar formas en las que la automatización de pruebas ayuda a verificar los defectos informados de acuerdo con los requisitos.
CT-TAS-4.1.3	(K2)	Definir enfoques que permitan el desarrollo de escenarios operacionalmente relevantes para la automatización de pruebas.

4.2 Estrategias de implementación

CT-TAS-4.2.1	(K2)	Definir una estrategia de implementación de automatización de pruebas.
CT-TAS-4.2.2	(K2)	Identificar los riesgos de la automatización de pruebas en la implementación.
CT-TAS-4.2.3	(K2)	Definir enfoques para mitigar los riesgos de implementación de pruebas.

4.3 Dependencias dentro del entorno de pruebas

CT-TAS-4.3.1	(K2)	Definir los componentes de automatización de pruebas en el entorno de pruebas.
CT-TAS-4.3.2	(K2)	Identificar los componentes de infraestructura y las dependencias de la automatización de pruebas.
CT-TAS-4.3.3	(K2)	Definir los datos de automatización de pruebas y los requisitos de interfaz para la integración dentro del sistema bajo prueba.

4.1 Planificación de la solución de automatización de pruebas

4.1.1 Identificar formas en que la automatización de pruebas permite acortar los tiempos de comercialización

La automatización de pruebas ayuda a que el software llegue al mercado más rápido porque ayuda a reducir el tiempo del ciclo de prueba. Las pruebas previas al lanzamiento de un software requieren una gran cantidad de verificación y validación. Esto puede ser particularmente importante durante las pruebas de regresión, ya que este tipo de pruebas promueve la reutilización, reduce los costos y brinda coherencia entre las ejecuciones de pruebas.

La automatización de pruebas ayuda a reducir el esfuerzo de prueba manual y proporciona una retroalimentación rápida a los desarrolladores, mientras que se cubre el mismo alcance de prueba. También permite realizar pruebas en etapas más tempranas del ciclo de vida del desarrollo de software si se automatizan las pruebas de componentes y las pruebas de integración de componentes, que típicamente no se realizan de forma manual.

Un punto de control de calidad (quality gate) es una medida obligatoria incorporada en su proceso que el software debe cumplir antes de poder avanzar a la siguiente fase. La configuración de puntos de control de calidad basados en la automatización de pruebas permite un proceso de implementación acelerado en un entorno de preproducción o producción. Al aprovechar estos puntos de control de calidad, se pueden detectar defectos antes, lo que da como resultado un menor tiempo de comercialización. El tiempo de ejecución de las pruebas se puede reducir aprovechando la ejecución de pruebas en paralelo y siguiendo un enfoque de desplazamiento a la izquierda. Un enfoque de desplazamiento a la izquierda promueve una cultura de conciencia de calidad y fomenta la realización de pruebas en una etapa más temprana del ciclo de vida del desarrollo del software. Esto puede incluir múltiples casos de prueba independientes o pruebas entre navegadores (cross-browser testing).

Para obtener información adicional, consulte las secciones 3.1.3, 6.1.2 y 6.2.1.

4.1.2 Identificar formas en las que la automatización de pruebas ayuda a verificar los defectos informados

Las pruebas de confirmación realizadas después de la corrección de un defecto en el código pueden solucionar el defecto notificado. Por lo general, un probador sigue los pasos de prueba necesarios para replicar el defecto y verificar que ya no existe.

Los defectos tienen una forma de reintroducirse en versiones posteriores (por ejemplo, esto puede indicar un problema de gestión de configuración o de gestión del repositorio de código) y, por lo tanto, las pruebas de confirmación son candidatas adecuadas para la automatización de pruebas y se pueden agregar al conjunto de pruebas de regresión existente.

Una prueba de confirmación automatizada normalmente tiene un alcance de funcionalidad limitado. La implementación puede ocurrir en cualquier momento una vez que se informa un defecto y se comprenden los pasos de prueba necesarios para replicarlo.

El seguimiento de las pruebas de confirmación automatizadas permite informar el tiempo y la cantidad de ciclos de prueba empleados para resolver defectos.

Al verificar correcciones en múltiples Plataformas, dispositivos, navegadores y versiones de sistemas operativos con la automatización de pruebas, la cantidad de tiempo dedicado a las pruebas se reduce enormemente.

4.1.3 Definir enfoques que permitan el desarrollo de escenarios operativamente relevantes para la automatización de pruebas

Las pruebas de aceptación operativa garantizan la preparación del software para los sistemas de producción y, por lo general, se realizan justo antes de un lanzamiento. Este esfuerzo tiene como objetivo realizar una validación final de los sistemas, componentes y demás infraestructura del SSP, y probar si está listo para el pase a producción. Esto es necesario porque, a pesar de los mejores esfuerzos de un IAP, no hay garantía de que el SSP se comporte de la misma manera fuera del entorno de prueba y en producción.

Un buen plan de pruebas operativas incluirá pruebas de fiabilidad, tolerancia a fallas, integridad y mantenibilidad. A continuación, se presentan enfoques de automatización de pruebas que se pueden utilizar:

- **Análisis de código estático:** herramientas automatizadas que analizan el código en busca de seguridad y vulnerabilidades. Los resultados de las pruebas se almacenan para realizar análisis de tendencias a lo largo del tiempo.
- **Pruebas de extremo a extremo (End-to-End Testing):** Guiones automatizados que realizan escenarios completos de usuarios y evalúan todos los aspectos del entorno de pruebas para descubrir cualquier defecto.
- **Pruebas de conmutación por error (Failover Testing):** Guiones automatizados que prueban específicamente lo que sucede cuando el hardware de una aplicación se desconecta. Algunos ejemplos incluyen cómo se recupera la aplicación cuando los servidores físicos, servidores en la nube, redes, discos duros u otros componentes de hardware fallan. Los guiones de prueba están diseñados para medir el éxito o el fracaso de la capacidad del SSP para recuperarse automáticamente, lo cual es especialmente importante para las organizaciones que implementan ingeniería de caos.
- **Pruebas de copia de seguridad y restauración:** secuencias de comandos de prueba automatizadas que prueban el éxito de hacer una copia de seguridad de la versión actual y luego volver a un punto anterior (es decir, una versión anterior del software).
- **Pruebas de eficiencia de rendimiento (por ejemplo, pruebas de carga):** secuencias de comandos de prueba automatizadas que se pueden utilizar para simular que muchos usuarios utilizan el SSP a la vez. Los resultados de las pruebas se almacenan para realizar comparaciones y se muestran tendencias a lo largo del tiempo.
- **Revisión de la documentación operativa:** se pueden usar guiones de prueba automatizados para esta actividad para comparar versiones de la documentación del SSP y señalar a los equipos de desarrollo si es necesaria una actualización en función de las nuevas características que se agregan a una versión en particular.
- **Pruebas de seguridad:** se pueden crear guiones de prueba automatizados en combinación con herramientas de prueba de seguridad estándar de la industria para evaluar el SSP de manera estática y dinámica. Los resultados de las pruebas se almacenan a lo largo del tiempo para realizar comparaciones y análisis de tendencias.
- **Monitoreo basado en acuerdo de nivel de servicio de una organización:** los guiones de prueba automatizados utilizados durante el ciclo de vida del desarrollo de software se pueden reutilizar para monitorear las operaciones de producción y enviar alertas si una prueba automatizada falla. Esta es una medida proactiva para detectar interrupciones de producción antes de que los usuarios reales las experimenten.

La automatización de pruebas para la validación de software operativo puede brindar enormes beneficios, especialmente si las pruebas automatizadas se pueden ejecutar repetidamente y en

múltiples entornos. Se pueden crear conjuntos de pruebas automatizadas dedicados para que los resultados de las pruebas se puedan generar y comparar con ejecuciones de pruebas anteriores. Esto garantiza la coherencia cuando se lanzan nuevas versiones del SSP. Un conjunto de pruebas automatizadas confiable de condiciones de prueba específicas operativas puede reducir los costos a lo largo del tiempo.

4.2 Estrategias de implementación de automatización de pruebas

4.2.1 Definir la estrategia de implementación de automatización de pruebas

Una buena estrategia de implementación de automatización de pruebas tendrá en cuenta, entre otras cosas, el entorno de prueba general, las herramientas de prueba disponibles, el acceso al SSP, el almacenamiento de guiones de prueba y otras dependencias, y el aprovisionamiento de datos de prueba. Con estas consideraciones de alto nivel en mente, un IAP puede comenzar a pensar en una estrategia para desarrollar e implementar la SAP.

Entorno de prueba – Los IAP deben considerar cómo van a acceder al SSP en los diferentes entornos de prueba. Cuando comienzan a desarrollar su software de automatización de pruebas, ya sea que utilicen un enfoque basado en palabras clave u otra solución, deben considerar cómo se ejecutará esta solución en múltiples entornos de prueba. Por ejemplo, un guión de prueba debe desarrollarse de tal manera que pueda ejecutarse en un entorno de prueba y en un entorno de preproducción con cambios mínimos. Por lo general, se trata de cambiar una dirección web (URL) para una aplicación basada en la Web y luego el guión de prueba se ejecuta de la misma manera independientemente del entorno de prueba en el que se encuentre.

Herramientas – Los IAP también deberán considerar qué herramientas están utilizando para crear la SAP. Si se trata de una herramienta comercial, es probable que deban comprender los términos de licenciamiento. El IAP puede descubrir que su entorno de prueba necesita tener acceso al servidor de licencias de la herramienta. Esto debe tenerse en cuenta cuando se pretende utilizar el guión de prueba en varios entornos de prueba. El hecho de que el servidor de licencias esté disponible en el entorno de prueba no significa necesariamente que esté disponible en el entorno de preproducción.

Acceso al software – Es necesario comprender algunas consideraciones importantes sobre cómo acceder al SSP. Los guiones de prueba se pueden diseñar para aceptar parámetros de modo que los usuarios específicos o las credenciales con acceso especial se puedan actualizar rápidamente cuando cambien los end-points (por ejemplo, una URL) del SSP. Esto se vuelve muy importante cuando se pasa a diferentes entornos de prueba y es necesario cambiar la URL. Además, puede ser necesario utilizar diferentes credenciales (por ejemplo, a través de cuentas de usuario y de prueba, biometría y tarjetas inteligentes) según el entorno de preproducción. Un buen diseño de guión de prueba automatizado incluirá suficiente flexibilidad para que se puedan establecer parámetros simples y los guiones de prueba se ejecuten como se espera independientemente del entorno de prueba.

Almacenamiento de guiones de prueba – El IAP querrá decidir una ubicación central para almacenar y administrar los guiones de prueba automatizados. Una buena estrategia sería incorporar un repositorio de código fuente que utilice la gestión de configuración. De esta manera, los guiones de prueba pueden ser accesibles desde múltiples entornos de prueba siempre que tengan acceso al repositorio de código fuente y se puedan crear versiones para la versión específica del SSP. Todas las configuraciones y dependencias se pueden guardar en el mismo repositorio que los guiones de prueba y proporcionar una portabilidad óptima. En resumen, el SAP, el MTAP y todos los casos de prueba se pueden almacenar y administrar dentro de repositorios.

Aprovisionamiento de datos – Será importante entender si un guión de prueba automatizado depende de ciertos datos de prueba que ya existen en el entorno de prueba donde se está probando el SSP. Un buen diseño de guión de prueba evitará los datos estáticos (es decir, conjuntos de datos fijos) tanto como sea posible. Sin embargo, habrá situaciones en las que no sea factible. En estos casos, el IAP deberá determinar una solución para tener los datos de prueba precargados o utilizar la automatización de pruebas para crear guiones de prueba de configuración que generen los datos de prueba necesarios antes de que se ejecuten los guiones de prueba reales y prueben el SSP. Existen pros y contras para ambos enfoques. Por un lado, es conveniente que un administrador precargue los datos de prueba. Sin embargo, el IAP luego depende de otro recurso para tener la disponibilidad para respaldarlos. Ser autosuficiente con un guión de prueba de configuración elimina la necesidad de depender de otro recurso. Sin embargo, lleva tiempo armar estos guiones de prueba y se convierte en otra parte de la solución que necesita mantenimiento.

Al tener en cuenta todos los elementos mencionados anteriormente, la estrategia de automatización de pruebas podrá proporcionar una planificación y un control adecuados de las actividades de automatización de pruebas.

4.2.2 Identificar los riesgos de la automatización de pruebas durante la implementación

Los problemas técnicos pueden generar riesgos para el producto y el proyecto (para obtener más información, consulte el temario del CTFL, sección 5.2.2). Los problemas técnicos más comunes incluyen:

- Demasiadas abstracciones pueden generar dificultades para comprender lo que realmente hace el código de automatización de pruebas (por ejemplo, con palabras clave en el enfoque basado en palabras clave).
- Las tablas de datos de prueba pueden volverse demasiado grandes/complejas/engorrosas para migrarlas a otros entornos de prueba, lo que genera resultados de estado inconsistentes.
- Dependencia de la SAP para utilizar determinadas bibliotecas del SO u otros componentes que pueden no estar disponibles en todos los entornos de prueba del SSP.

Los riesgos típicos de un proyecto de implementación incluyen:

- Problemas de personal: conseguir a las personas adecuadas para mantener la automatización de las pruebas puede resultar difícil.
- Mantenimiento no planificado de la SAP debido a actualizaciones del SSP que provocan que el SAP funcione incorrectamente
- Retrasos en la introducción de la automatización de pruebas
- Retrasos en la actualización del SAP en función de los cambios realizados al SSP
- La SAP no puede capturar objetos de interfaz de usuario (UI) no estándar
- Permitir que los casos de prueba obsoletos permanezcan en los conjuntos de pruebas, desperdiciando tiempo de ejecución de pruebas.

Los posibles puntos de falla del proyecto SAP incluyen:

- Migración a un entorno de prueba diferente
- Implementación en un entorno de producción
- Olvidar que la automatización es un software que también debe probarse

Es importante tener en cuenta que diversos problemas técnicos, riesgos del proyecto y posibles puntos de falla pueden poner en peligro el éxito de los proyectos de automatización de pruebas. Los problemas técnicos más comunes incluyen complejidades que surgen de abstracciones excesivas,

desafíos con la gestión de datos de prueba y dependencias de componentes específicos. Los riesgos del proyecto también deben tener en cuenta las dificultades en la dotación de personal, los problemas de mantenimiento debido a las actualizaciones del sistema, los retrasos en la implementación y la presencia de casos de prueba obsoletos.

Además, los posibles puntos de falla, como la migración a diferentes entornos y el descuido de la necesidad de probar el software de automatización en sí, deben abordarse con una planificación adecuada. En general, la supervisión y las medidas proactivas garantizarán el éxito de los proyectos de automatización de pruebas

4.2.3 Definir enfoques para mitigar los riesgos de implementación

Existen muchas estrategias de mitigación de riesgos que pueden emplearse para abordar estas áreas de riesgo. Algunas se analizan a continuación.

La SAP tiene su propio ciclo de vida de desarrollo de software (SDLC), ya sea que se desarrolle internamente o sea una solución adquirida. Algo que hay que recordar es que la SAP, como cualquier otro software, necesita estar bajo gestión de configuración y sus características documentadas. De lo contrario, se vuelve extremadamente difícil implementar diferentes partes de esta y hacer que funcionen juntas o en ciertos entornos de prueba.

Además, debe existir un procedimiento de implementación documentado, claro y fácil de seguir. Este procedimiento depende de la versión, por lo que también debe incluirse en la gestión de configuración.

Existen dos casos distintos a la hora de implementar un SAP:

1. Implementación por primera vez
2. Implementación de mantenimiento: la SAP ya existe y es necesario implementar una actualización

Los riesgos relacionados con la implementación por primera vez incluyen:

- El tiempo total de ejecución de la prueba del conjunto de pruebas puede ser más largo que el tiempo de ejecución de la prueba planificado para el ciclo de pruebas. En este caso, es importante asegurarse de planificar tiempo suficiente para que el conjunto de pruebas se ejecute en su totalidad antes de que comience el siguiente ciclo de pruebas programado.
- Existen problemas de instalación y configuración con los entornos de prueba (por ejemplo, configuración y carga inicial de la base de datos, y el inicio y la detención de los servicios). La SAP necesita un dispositivo de prueba (es decir, un conjunto de datos predefinido) para crear las condiciones previas necesarias para que los casos de prueba automatizados se ejecuten dentro del entorno de prueba.

Para las implementaciones de mantenimiento, existen consideraciones adicionales. La SAP en sí debe evolucionar y las actualizaciones deben implementarse en producción. Antes de implementar una versión actualizada de la SAP en producción, es necesario probarla. Por lo tanto, es necesario verificar la nueva funcionalidad, verificar que se pueda ejecutar un conjunto de pruebas en la SAP actualizada, que se puedan enviar informes de pruebas y que no haya defectos de rendimiento u otros problemas de calidad. En algunos casos, es posible que sea necesario cambiar todo el conjunto de pruebas para que se ajuste a la última versión de la SAP.

4.3 Dependencias dentro del entorno de pruebas

4.3.1 Definir enfoques para mitigar los riesgos de implementación

Los componentes de automatización de pruebas suelen constar de herramientas, máquinas virtuales, guiones de pruebas de automatización, contenedores y configuraciones. El entorno de pruebas también incluye el SSP. Los componentes de automatización de pruebas del entorno de pruebas se pueden definir de la siguiente manera:

SSP – Esta es una parte obvia del entorno de prueba. El SSP se puede probar como un todo o dividido en subcomponentes (por ejemplo, API, interfaz basada en Web y base de datos).

Plataforma – La plataforma describe dónde se alojan los componentes de automatización de pruebas. Esto incluye la infraestructura en la nube, la red, las máquinas virtuales y los contenedores que se pueden utilizar para que la automatización de pruebas sea eficiente y portátil.

Casos de prueba y conjuntos de pruebas – Aquí se describen casos de prueba individuales que se componen de pasos de prueba que dirigen acciones tanto automáticas como manuales. Los conjuntos de pruebas describen una agrupación lógica de casos de prueba para que se puedan ejecutar juntos de manera eficiente.

Herramientas – Se utilizan distintas herramientas de automatización de pruebas por diversos motivos. Una colección de herramientas incluiría aquellas para la automatización de pruebas de interfaz de usuario, pruebas de end-points de API, generación de datos, monitoreo, gestión de requisitos, gestión de defectos, herramientas de registro e informes, y herramientas para generar tendencias basadas en métricas.

MTAP – Esto incluye todos los elementos que componen el diseño del MTAP. Los MTAP incluyen guiones de controladores, bibliotecas comunes, plantillas para casos de prueba automatizados, guiones de carga y aprovisionamiento de datos, documentación sobre cómo usar los componentes del MTAP y tutoriales que ayudan a los IAP a utilizar el MTAP. Para obtener información adicional, consulte el temario CTAL-TAE, sección 3.1.

El mantenimiento de los componentes de prueba es una consideración importante porque complicar demasiado el entorno de prueba puede dar como resultado horas de tiempo no deseado para corregir defectos en el MTAP en lugar de aprovechar la solución. Es importante encontrar la combinación adecuada de herramientas, configuración y portabilidad de la plataforma para que los componentes sean lo más reutilizables posible.

4.3.2 Identificar los componentes de infraestructura y las dependencias de la automatización de pruebas

Hay una serie de componentes y dependencias de infraestructura que se deben tener en cuenta al ensamblar la automatización de pruebas. En conjunto, cubren todos los requisitos previos necesarios para ejecutar una SAP. Los principales componentes y dependencias incluyen:

Servidores (Host Machine) – Pueden ser máquinas virtuales, servidores físicos, computadoras portátiles y dispositivos (por ejemplo, tabletas y dispositivos móviles). Tienen instalado el software de automatización de pruebas y es allí donde se crean y ejecutan los guiones de prueba.

Red – Esto es lo que le da al SAP acceso al SSP. También puede incluir la conexión en red de múltiples servidores para proporcionar la ejecución paralela de pruebas automatizadas. Por lo general,

se requiere que los servidores estén en la misma red y configuradas correctamente para comunicarse entre sí.

Plataforma – La automatización de pruebas, como cualquier otro software, puede ejecutarse en plataformas en la nube o estar diseñada para ejecutarse en contenedores. Siempre que la plataforma proporcione permisos y acceso al sistema operativo subyacente, se pueden instalar todas las herramientas y dependencias necesarias.

Dependencias de software – Para que las herramientas de automatización de pruebas funcionen correctamente, es necesario comprender todas las demás dependencias que tiene la automatización de pruebas. Por ejemplo, una herramienta de automatización de pruebas en particular puede requerir que primero se instale la última versión de un lenguaje de programación en el servidor. Los IAP deben tener en cuenta estas dependencias antes y después de seleccionar una herramienta.

SSP – Una vez que se han considerado y configurado correctamente los componentes, las dependencias y la infraestructura general, el último paso es garantizar el acceso al SSP. Además de la red, también es necesario considerar cómo interactuar con el SSP. Por ejemplo, en una aplicación basada en la Web, se debe instalar un navegador en un servidor. Se debe considerar el tipo y la versión del navegador.

4.3.3 Definir los requisitos de interfaz y datos de automatización de pruebas

Dos consideraciones que los IAP deben tener en cuenta antes de desarrollar guiones de automatización de pruebas son cómo planean interactuar con el SSP y cuáles son las dependencias de datos de los casos de prueba. A continuación, se ofrecen algunos ejemplos:

API – Si el SSP utiliza una API, esta se puede probar en el nivel de end point en lugar de requerir una interfaz de usuario para interactuar en el nivel de aplicación. Esto puede requerir una comprensión más profunda de los end points y las comunicaciones de datos que están ocultos por una interfaz de usuario. El IAP tendría que entender en qué orden llamar a las API para crear un proceso de negocios y cómo correlacionar los datos para mantener intacto el caso de prueba. Las API que están expuestas a Internet también se denominan API web o servicios web.

Interfaz de base de datos – Algunas herramientas de automatización de pruebas tienen la capacidad de interactuar directamente con la base de datos subyacente del SSP. Se pueden escribir guiones de prueba para verificar los datos dentro de las columnas y filas a fin de garantizar que los procedimientos almacenados y otras reglas de la base de datos estén configurados correctamente. Esto puede requerir que ya existan valores de datos específicos en la base de datos para realizar pruebas de fiabilidad.

Compatibilidad de interfaz – El uso de pruebas de contrato (contract testing) garantiza que dos sistemas separados (por ejemplo, dos microservicios) sean compatibles y puedan comunicarse entre sí. Las pruebas de contrato pueden ser impulsadas por el consumidor (Consumer-driven), impulsadas por el productor (Producer-driven) y bidireccionales. Se pueden encontrar detalles adicionales en el programa de estudios CTAL-TAE, sección 5.1.3.

Además, los IAP deben considerar cuidadosamente cómo interactuarán con el SSP y comprender las dependencias de datos dentro de los casos de prueba. Ya sea que realicen pruebas a través de API, interfaces de bases de datos o que garanticen la compatibilidad de interfaces entre sistemas, prestar atención a estas consideraciones es esencial para una automatización de pruebas sólida y eficaz. Al abordar estos factores de manera reflexiva, los IAP pueden mejorar la confiabilidad y la eficiencia de sus esfuerzos de automatización, lo que en última instancia contribuye a la calidad y el éxito del SSP.

5 Análisis del impacto de la automatización de pruebas (K3)

Duración: 150 minutos

Palabras Clave: cobertura, informe de pruebas

Objetivos de Aprendizaje para el Capítulo 5

5.1 Inversión en la creación y el mantenimiento de la automatización de pruebas

CT-TAS-5.1.1	(K3)	Mostrar la rentabilidad de la inversión en la creación de una solución de automatización de pruebas.
--------------	------	--

5.2 Métricas de automatización de pruebas

CT-TAS-5.2.1	(K2)	Clasificar las métricas para la automatización de pruebas.
--------------	------	--

5.3 El valor de la automatización de pruebas a nivel de proyecto y de organización

CT-TAS-5.3.1	(K3)	Identificar las consideraciones organizativas para el uso de la automatización de pruebas
CT-TAS-5.3.2	(K3)	Analizar las características del proyecto que ayudan a determinar la implementación óptima de los objetivos de la prueba de automatización de pruebas.

5.4 Decisiones tomadas a partir de informes de automatización de pruebas

CT-TAS-5.4.1	(K2)	Analizar los datos de los informes de pruebas para fundamentar la toma de decisiones.
--------------	------	---

5.1 Inversión en la creación y el mantenimiento de la automatización de pruebas

5.1.1 Mostrar el retorno de la inversión de crear una solución de automatización de pruebas

Es importante calcular los costes de configuración y mantenimiento de la automatización de pruebas antes de proceder a la implementación de un proyecto. Más allá de los valores que la automatización puede aportar a un proyecto, comprender el cálculo del ROI del proyecto es beneficioso.

El cálculo del ROI puede proporcionar información significativa en cualquier momento del ciclo de vida de un proyecto al demostrar el rendimiento de una actividad por el esfuerzo invertido. El cálculo del ROI puede ajustarse aún más incluyendo tanto los costos de los probadores manuales como los de los TAE, simplemente multiplicando el tiempo invertido por sus respectivos costes laborales.

Para calcular el ROI, hay que determinar la inversión de la automatización de pruebas (es decir, tiempo y coste) y el ahorro que se consigue con ella:

$$\text{ROI} = \text{Ahorro} / \text{Inversión}$$

Es importante señalar que el ahorro y la inversión pueden calcularse considerando diferentes métricas y datos, en diferentes medidas y unidades. En el ámbito de este temario, se utiliza un modelo sencillo para demostrar el enfoque, con unidades de tiempo y no de costo. Si determinadas actividades sólo se miden en coste, esa cantidad puede convertirse en tiempo utilizando una tasa específica del proyecto.

Generalmente, el ahorro que se consigue con la automatización de pruebas se debe a que las mismas pruebas pueden ejecutarse en mucho menos tiempo que de forma manual. Esto también significa que pueden ejecutarse más a menudo. Por lo tanto, se puede aumentar el número de pruebas que se ejecutan.

Para calcular el ahorro, hay que tener en cuenta los siguientes parámetros:

- Tiempo de ejecución manual de un caso de prueba
- Tiempo de ejecución de un caso de prueba automatizado
- Número de casos de prueba
- Número de pruebas corridas

Para calcular la inversión, hay que tener en cuenta los siguientes parámetros:

- Tiempo de configuración para la automatización de pruebas
- Tiempo medio de desarrollo de scripts de pruebas automatizadas
- Número de scripts de prueba automatizados aplicados
- Tiempo medio de mantenimiento de un script de prueba automatizado
- Tiempo de ejecución de un script de prueba automatizado
- Porcentaje de scripts de pruebas automatizados fallidos
- Número de casos de prueba definidos
- Número de pruebas corridas

Adaptando el modelo descrito al desarrollo ágil de software, se pueden prever los sprints (iteraciones) de un proyecto como se ilustra en el gráfico siguiente. El gráfico permite determinar el sprint a partir del cual la automatización de pruebas ha rentabilizado su inversión.

Figura 2: Ejemplo de cálculo del ROI que muestra el punto de retorno de la inversión

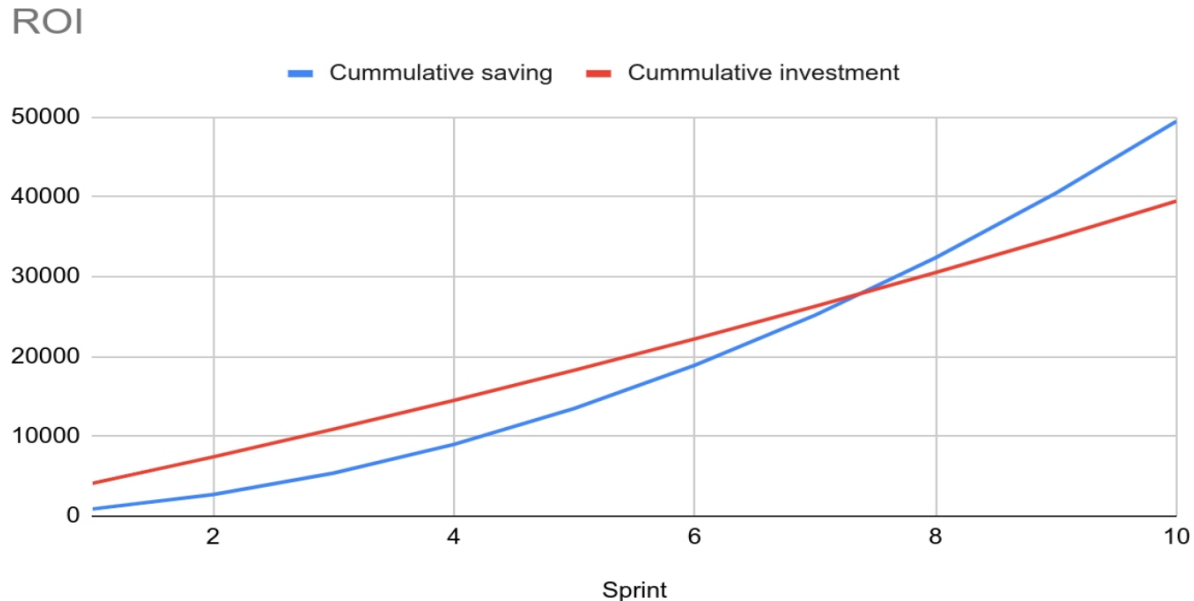


Figura 2: Ejemplo de cálculo del ROI que muestra el punto de retorno de la inversión

Tenga en cuenta ciertas conexiones entre las métricas medidas para calcular el ROI, entre las que se incluyen:

- Si la duración prevista del proyecto es inferior al punto de inflexión del ROI, no merece la pena introducir la automatización de las pruebas. En este caso, ejecutando las pruebas manualmente se ahorra tiempo y esfuerzo.
- Dado que la inversión depende en gran medida del tiempo de ejecución de las pruebas automatizadas, aplicar la pirámide de pruebas e implementar casos de prueba en el nivel de prueba adecuado puede reducir este tiempo de ejecución y mejorar el ROI.

5.2 Métricas de automatización de pruebas

5.2.1 Clasificación de métricas para la automatización de pruebas

El análisis de tendencias basado en datos objetivos ayuda a tomar decisiones. Mediante la recopilación de métricas de una SAP, los probadores pueden evaluar la SAP y tomar decisiones al respecto:

- Adecuación de la SAP al proyecto
- Adaptabilidad SAP para ampliar la funcionalidad a nuevas condiciones de pruebas
 - Un cambio en un flujo de usuario en el SUT
 - Un cambio en la forma de realizar las pruebas
- Mantenibilidad de la automatización de las pruebas debido a los defectos encontrados en la SAP

El coste de la medición debe ser lo más bajo posible, lo que a menudo puede lograrse automatizando la recolección y notificación de métricas. A continuación, se ofrecen algunos ejemplos.

Proporción de pruebas superadas/no superadas

Se trata de una métrica habitual que registra la relación entre las pruebas automatizadas que se han superado y las que no han logrado el resultado esperado. $\text{Proporción de pruebas superadas/no superadas} = \frac{\text{número de pruebas superadas}}{\text{número de pruebas no superadas}}$.

Relación entre fallos y defectos

Un problema común con las pruebas automatizadas es que muchas de ellas pueden fallar por la misma razón, es decir, por un único defecto en el SSP. Medir el número de pruebas automatizadas que fallan por un defecto determinado puede ayudar a indicar dónde puede estar el problema. Además, se puede recopilar el número de fallos por defecto y rastrear la razón del fallo: defecto en el subsistema, en todo el sistema, problema con los datos de prueba o con la infraestructura.

Tiempo de ejecución de la automatización de pruebas

Una de las métricas más fáciles de determinar es el tiempo que se tarda en ejecutar las pruebas automatizadas. Al principio de la SAP esto puede no ser importante, pero a medida que aumenta el número de casos de pruebas automatizadas, esta métrica puede llegar a ser bastante importante. Esta métrica incluye el tiempo de construcción de la SAP.

Número de casos de prueba automatizados

Esta métrica puede utilizarse para mostrar los avances del proyecto de automatización de pruebas. Pero hay que tener en cuenta que el número de casos de prueba automatizados no revela mucha información; por ejemplo, no indica el nivel de cobertura.

Cobertura funcional de la automatización de pruebas

Esta cobertura indica el porcentaje de requisitos funcionales cubiertos por casos de prueba automatizados.

Cobertura del código

La cobertura de código registra cuántas líneas de código son ejercitadas por pruebas de nivel inferior (es decir, de componentes). No existe un porcentaje absoluto que indique una cobertura adecuada, y a menudo es difícil alcanzar el 100% de cobertura del código en cualquier software que no sea el más sencillo. Sin embargo, en general se está de acuerdo en que una mayor cobertura es mejor, ya que aumenta la confianza en el SUT. A medida que se añaden pruebas automatizadas de bajo nivel, se espera que aumente la cobertura del código.

5.3 El valor de la automatización de pruebas a nivel de proyecto y de organización

5.3.1 Identificar las consideraciones organizativas para el uso de la automatización de pruebas

Antes de iniciar la automatización de pruebas en cualquier proyecto, es necesario identificar lo siguiente dentro de la organización:

Políticas y prácticas de desarrollo de software

Es bueno comprobar cómo trabajan los equipos de desarrollo y qué tipo de documentación existe. Cualquier documentación disponible puede ser beneficiosa para identificar cómo se puede conectar la automatización de pruebas en los procesos de los equipos de desarrollo. La documentación puede incluir la especificación técnica del SSP, el software y las herramientas de desarrollo utilizadas, o cualquier política disponible sobre desarrollo, como directrices de revisión de código, normas de codificación definidas y procesos de fusión.

Proyectos activos de automatización de pruebas y su estado

En el caso de un proyecto de desarrollo en curso, puede haber uno o varios proyectos de desarrollo de la SAP en curso por parte de distintos equipos. Una recomendación general para los responsables de la toma de decisiones es comprobar estos proyectos existentes y su estado, para ver si alguno de ellos se ajusta a los objetivos de prueba definidos de la nueva SAP. Analizando la solución, se puede determinar si conviene reutilizar alguno de los SAP existentes o crear uno nuevo basado en las necesidades actuales.

Los expertos de la organización en materia de automatización de pruebas

Se recomienda buscar expertos en la materia (EM o por sus siglas en inglés SMEs) que puedan ayudar en el despliegue de una nueva SAP en la organización. Los EM (EM o por sus siglas en inglés SMEs) pueden compartir historias y lecciones aprendidas sobre sus SAP y su implementación. A partir de ahí, se pueden identificar los distintos riesgos que hay que tener en cuenta o evitar para que la implantación de la SAP sea un éxito.

Disponibilidad de entornos de prueba

En el caso de grandes organizaciones, una recomendación general es recopilar información sobre los entornos de prueba existentes, su uso y disponibilidad. Esta información es crucial para evitar arruinar el trabajo de cualquier otro equipo al desplegar una nueva SAP y utilizar el sistema sin ninguna notificación o acuerdo. A menudo, las necesidades de un proyecto requerirán un nuevo entorno de pruebas, por lo que los existentes pueden utilizarse como referencia para crear uno nuevo y el trabajo puede realizarse más fácilmente si los equipos y personas responsables están identificados y disponibles.

Herramientas de prueba y licencias

Siempre merece la pena obtener información sobre las herramientas y licencias de que dispone actualmente la organización. Si se identifican éstas, los costes y los plazos, puede reducirse la planificación de una SAP. Y es posible que las herramientas para un nuevo proyecto ya estén disponibles. Por ejemplo, si las pruebas en la nube se realizan a través de un proveedor definido, y las licencias están disponibles para el nuevo proyecto, no tiene sentido utilizar un proveedor de nube diferente. Se recomienda utilizar las mismas herramientas y licencias para reducir los costes del proyecto.

5.3.2 Analizar las características del proyecto que ayudan a determinar la implementación óptima de los objetivos de prueba de la automatización de pruebas

Existen varias características principales de los proyectos que pueden definir una forma óptima de trabajar y ayudar a definir objetivos de automatización de pruebas satisfactorias.

Dominio

Es importante comprender que cada ámbito difiere en cuanto a reglamentos o normas. Por ejemplo, las normativas y los riesgos son diferentes en el ámbito del turismo que en el de las finanzas o la sanidad. Siempre es recomendable comprobar las diferentes normas y restricciones de dominio para asegurarse de que se cumplen los objetivos de automatización de pruebas previstos.

Plataformas

En cuanto a los objetivos de la automatización de pruebas, también se hace hincapié en evaluar qué plataformas cubre el proyecto y dónde sería beneficioso realizar la automatización de pruebas. La planificación de la automatización de pruebas en múltiples plataformas puede resultar más difícil, ya que puede ser necesario contar con varias soluciones para cubrir las distintas plataformas. En muchos casos, como para móviles y web, pueden utilizarse las mismas herramientas. Pero la planificación determinará la reutilización de la SAP.

Lenguaje de programación y pila tecnológica

Los detalles de ejecución de un proyecto, como su lenguaje de programación y su pila tecnológica, también determinan qué tipo de objetivos de pruebas de automatización de pruebas debe definir una organización. Se recomienda utilizar los mismos lenguajes de programación que utilizan los desarrolladores en un proyecto determinado. De este modo, la colaboración puede ser mucho más fácil, y el aprendizaje cruzado mediante la realización conjunta de revisiones de código mejorará aún más la calidad del SSP y los conocimientos de los IAP.

Madurez del proyecto

Analizando la madurez del proyecto, se puede obtener información para diseñar el objetivo ideal de automatización de pruebas. Al identificar los diferentes factores, como el número de casos de prueba, los conductos de CI/CD (por sus siglas en inglés) existentes, el número de probadores y IAP, se pueden crear diferentes objetivos de prueba y una hoja de ruta. Por ejemplo, si hay muchos casos de prueba manuales que tienen alta prioridad y su tiempo de ejecución manual es largo, es importante tratar de automatizarlos primero para ahorrar tiempo y esfuerzo. Aparte de los factores anteriores, el calendario del proyecto también es muy importante. Si el proyecto es corto o va a terminar en breve y no hay mucho tiempo para hacer las cosas, no merece la pena planificar una SAP grande y robusto. Sin embargo, si se trata de un proyecto totalmente nuevo, merece la pena diseñar una hoja de ruta incremental basada en el producto mínimo viable paso a paso.

Participación de las partes interesadas

Muchas veces, la automatización de pruebas no se aprovecha debido a que las partes interesadas clave no están del todo convencidos. Esto puede deberse al miedo a que la velocidad caiga por debajo de su objetivo, a plazos ajustados, al presupuesto o a otras preocupaciones. Tras analizar la madurez del proyecto, si hay margen para empezar a implantar una SAP, una parte interesada estratégica debe identificar los riesgos del producto, enumerar las ventajas de introducir la automatización de pruebas y elaborar un plan adecuado. Este plan debe destacar las metas que deben alcanzarse con y mediante la automatización de pruebas e indicar cómo debe mejorarse la capacidad de prueba del SUT para introducir con éxito una SAP que aborde los riesgos identificados. Por último, este plan debe presentarse a las partes clave interesadas.

Conocimiento del equipo y experiencia relevante

El factor más importante y relevante para el éxito de la automatización de pruebas son las habilidades y la experiencia de los probadores. Es beneficioso trabajar con los probadores y utilizar sus conocimientos para definir objetivos de prueba con los que ellos y los analistas de negocio se sientan cómodos. A menudo, los directores de pruebas o los líderes de pruebas crean una matriz de competencias y habilidades para identificar los conocimientos disponibles en los equipos y, al mismo tiempo, detectar las brechas. Estas brechas pueden identificarse con distintos objetivos, como la capacitación y la asignación de tareas de implementación.

Apoyo a la gestión de pruebas y presupuestación

Dependiendo del tamaño y la madurez del proyecto, hay que tener en cuenta el presupuesto disponible y el apoyo de la dirección de pruebas a la hora de definir los objetivos de las pruebas de automatización. Es importante definir objetivos de prueba que puedan cumplirse, lo que a su vez permitirá contar con el apoyo de la dirección de pruebas. Al hacer una propuesta de estrategia de automatización de pruebas a la dirección, la documentación debe ser concisa y definir claramente las carencias identificadas en la actualidad o los costes futuros de la automatización de pruebas si se aplica correctamente. Una lista de recomendaciones y sus ventajas, señalando el valor empresarial y la reducción de costes, animará a la dirección de pruebas a aprobar el desarrollo o la mejora de una SAP.

Características de calidad

La norma ISO/IEC 25010:2011 define las características de calidad que se enumeran en el ISTQB CTFL Syllabus, sección 2.2.2. Tipos de pruebas. Estas características de calidad pueden ser utilizadas para evaluar la SAP actual, o para determinar las métricas que serán recogidas por la SAP.

5.4 Decisiones tomadas a partir de los informes de automatización de pruebas

5.4.1 Analizar los datos de los informes de pruebas para fundamentar la toma de decisiones

El formato y el contenido de un informe de automatización de pruebas pueden variar en función de las partes interesadas que lo reciban. Puede crearse para las partes interesadas de gestión, operativas o técnicas. Encontrará más información en el programa de estudios de TAL-TAE, apartado 6.1.3.

Al recibir un informe de automatización de pruebas, puede incorporarse a un informe de pruebas más amplio, o puede consolidarse y, a continuación, escalarse dentro de la estructura organizativa.

Diferentes partes interesadas encuentran diferentes valores en un informe de automatización de pruebas de este tipo. Un enfoque estratégico consiste en identificar las métricas clave que son importantes para las partes interesadas implicadas, haciendo hincapié en estas métricas importantes.

Los datos recolectados con la automatización pueden ayudar:

- Identificar tendencias y realizar análisis de causas
- Desplazar los esfuerzos de automatización de pruebas al mantenimiento
- Desplazar los esfuerzos de automatización de las pruebas hacia la mejora y el desarrollo de un MTAP.
- Añadir capacidades a la SAP general

- Aumentar los enfoques de las pruebas por desplazamiento a la izquierda y a la derecha
- Ampliar la cobertura funcional de la automatización de pruebas en futuros sprints (iteraciones)
- Centrarse más en los grupos de defectos
- Asesorar a los desarrolladores sobre las áreas en las que se puede mejorar el código
- Asesoramiento sobre los procesos generales del SDLC (por sus siglas en inglés)
- Cambiar el contenido y el formato de los futuros informes de automatización de pruebas

A partir de la información descrita anteriormente, las SAP, en colaboración con otras partes interesadas, pueden identificar brechas y ciertos puntos de mejora en la cobertura de automatización de pruebas y los resultados de las pruebas existentes.

6 Estrategias de implementación y mejora para la automatización de pruebas (K3)

Duración: 150 minutos

Palabras clave: cobertura, precondition, conjunto de pruebas

Objetivos de Aprendizaje para el Capítulo 6

6.1 Transición de actividades de pruebas manuales a pruebas continuas

CT-TAS-6.1.1	(K2)	Describir los factores y las actividades de planificación en la transición de las pruebas manuales a las pruebas automatización.
CT-TAS-6.1.2	(K2)	Describir los factores y las actividades de planificación en la transición de la automatización de pruebas a pruebas continua.

6.2 Comprensión de los factores y actividades de planificación en la transición de la automatización de pruebas a Prueba continua

CT-TAS-6.2.1	(K3)	Realizar una evaluación de los activos y prácticas de automatización de pruebas para identificar áreas de mejora.
--------------	------	---

6.1 Transición de actividades de pruebas manuales a pruebas continuas

6.1.1 Describir los factores y las actividades de planificación en la transición de las pruebas manuales a la automatización de pruebas

La oportunidad más sencilla de pasar de las pruebas manuales a la automatización de las pruebas es centrarse en las pruebas de regresión. Un conjunto de pruebas de regresión crece a medida que las pruebas funcionales y no funcionales de hoy se convierten en las pruebas de regresión de mañana. Es solo cuestión de tiempo antes de que la cantidad de pruebas de regresión sea mayor que el tiempo y los recursos disponibles para un equipo de pruebas manuales tradicionales.

Costos de transición

Durante la transición de las pruebas manuales a las automatizadas, el proyecto debe esperar un incremento de costos, ya que tanto las pruebas manuales como las automatizadas se realizan simultáneamente. Una vez que se considera que las pruebas automatizadas reemplazan o sustituyen adecuadamente las actividades de ejecución de pruebas manuales, se puede dedicar más esfuerzo a las pruebas exploratorias y a la definición de casos de prueba adicionales para la automatización de pruebas, que tienen un costo diferente en comparación con las pruebas de regresión manuales.

Superposición funcional

La superposición funcional se produce cuando los desarrolladores de scripts de prueba incluyen exactamente los mismos pasos de automatización de pruebas en diferentes casos de prueba. Por ejemplo, la mayoría de los casos de prueba comenzarán con una secuencia de inicio de sesión como paso de prueba. Esto puede incluir ingresar un nombre de usuario, una contraseña y seleccionar un botón de inicio de sesión. Agregarlo en cada caso de prueba aumenta las actividades de mantenimiento. Si alguna vez se agrega un paso de prueba adicional al proceso de inicio de sesión, se deberá actualizar el mismo cambio en cada caso de prueba. Un mejor enfoque es hacer un componente de automatización de pruebas reutilizable a partir del proceso de inicio de sesión y hacer que todos los casos de prueba hagan referencia a esa funcionalidad. Consulte el programa de estudios CTAL-TAE, sección 3.1.5 para obtener detalles sobre el patrón del modelo de flujo.

Intercambio de datos

Las pruebas suelen compartir datos de prueba. Esto puede ocurrir cuando las pruebas utilizan el mismo registro de datos de prueba para ejecutar distintas funciones del SSP. Un ejemplo de esto podría ser el caso de prueba "A", que verifica el tiempo de vacaciones disponible de un empleado, mientras que el caso de prueba "B" verifica qué cursos tomó el empleado como parte de sus objetivos de desarrollo profesional.

Cada caso de prueba utiliza el mismo empleado, pero verifica diferentes parámetros. En un entorno de prueba manual, los datos de prueba del empleado normalmente se duplicarían muchas veces en cada caso de prueba manual que verificara los datos de prueba del empleado utilizando a este empleado. Sin embargo, en una prueba automatizada, los datos de prueba que se comparten deben, siempre que sea posible y factible, almacenarse y accederse a ellos desde una única fuente para evitar la duplicación o la introducción de errores.

Precondiciones para la ejecución de pruebas

Con demasiada frecuencia, los IAP comienzan inmediatamente a desarrollar scripts de prueba sin comprender primero las condiciones previas o precondiciones que se requieren para garantizar que un caso de prueba se pueda ejecutar de manera confiable. Esto puede volverse extremadamente complicado cuando se intenta ejecutar un caso de prueba sobre el mismo SSP en múltiples entornos de prueba. Algunos ejemplos de precondiciones incluyen nombres de usuario, roles de cuenta, valores de tablas de datos y otras entradas de datos específicas que hacen que el caso de prueba sea repetible. Una buena estrategia incluirá un análisis previo para comprender qué información debe existir en el SSP antes de automatizar un caso de prueba en particular. Además, la estrategia se puede mejorar automatizando la creación de precondiciones antes de que se ejecuten los casos de prueba reales, para ahorrar tiempo. Esto puede incluir la ejecución de scripts de prueba de precondiciones que completen el SSP desde la interfaz de usuario o procesos por lotes automatizados que carguen datos de prueba en una base de datos.

Cobertura funcional

Identifique las brechas funcionales en las pruebas que pueden ser candidatas para la automatización de pruebas, como se explica en el Capítulo 3. El 100% de los casos de prueba manuales que se automatizan no representa el 100% de todos los casos de prueba que se pueden automatizar con herramientas de prueba. A medida que más pruebas se automatizan, los probadores recuperan tiempo de ejecución de pruebas que pueden usar para identificar pruebas adicionales del SSP y aumentar la cobertura.

Pruebas ejecutables

Antes de convertir una prueba de regresión manual en una prueba de regresión automatizada, es importante verificar que la prueba de regresión manual funciona correctamente. Esto proporciona el punto de partida correcto para garantizar una conversión exitosa a una prueba de regresión automatizada. Si la prueba de regresión manual no se ejecuta correctamente, puede deberse a que se redactó de manera deficiente, utiliza datos de prueba no válidos, está desactualizada o no está sincronizada con el SSP actual, o debido a un defecto de SSP. Automatizarla antes de comprender o resolver la causa raíz de la falla creará una prueba de regresión automatizada que no funciona, lo que es un desperdicio y no es productivo. Es importante demostrar la funcionalidad equivalente que brindan las nuevas pruebas automatizadas para transmitir confianza en las pruebas automatizadas que reemplazarán a las antiguas pruebas manuales.

6.1.2 Describir los factores y las actividades de planificación en la transición de la automatización de pruebas a las pruebas continuas

Las pruebas continuas implican el uso de recursos de prueba con mucha más frecuencia que en los ciclos de vida de desarrollo de software tradicionales. Esto se logra comenzando a ejecutar constantemente conjuntos de pruebas inmediatamente después de que se realizan cambios en el código y se ponen a disposición en los entornos de prueba. Esto ayuda a proporcionar retroalimentación inmediata y reduce los costos de los defectos al detectarlos en una etapa más temprana del proceso. Esto se puede manejar con herramientas de desarrollo sofisticadas y la voluntad de iniciar las pruebas en una etapa más temprana del proceso de compilación.

La adaptación del SAP para pruebas continuas requiere lo siguiente:

- Los juegos de pruebas deben cambiar para ejecutarse en la SAP actualizada: realice los cambios necesarios en las suites de pruebas y pruébelas antes de implementarlas en la SAP.
- Los stubs, controladores e interfaces utilizados en las pruebas deben cambiar para adaptarse la SAP actualizada: realice los cambios necesarios en el arnés de prueba y pruébelo antes de implementarlo en la SAP.

- La infraestructura debe cambiar para adaptarse a la SAP actualizada: haga una evaluación de los componentes de la infraestructura que deben cambiarse.
- La SAP actualizada tiene defectos adicionales o defectos de rendimiento: realice un análisis de riesgos versus beneficios. Si los defectos descubiertos hacen que sea imposible actualizar la SAP, puede ser mejor no continuar con la actualización o esperar a una próxima versión de la SAP. Si los defectos son insignificantes en comparación con los beneficios de corregirlos, la SAP aún puede actualizarse.
- Asegúrese de crear notas de versión de los defectos conocidos para notificar a los IAP y otras partes interesadas e intente obtener una estimación de cuándo se solucionarán los defectos.

Todos los puntos mencionados en esta sección se vuelven particularmente importantes cuando se utiliza CI/CD. La creación de flujos (pipelines) y la automatización del proceso de compilación es una combinación natural con la SAP que se está desarrollando. Si la herramienta de orquestación de compilación se encuentra en el entorno de prueba correcto, el pipeline se puede ampliar para incluir pruebas automatizadas para verificar el SSP inmediatamente después de la implementación. Esto requiere que la herramienta de automatización de pruebas esté configurada correctamente y pueda acceder al SSP, y que pueda comunicarse con el repositorio de código y acceder a los guiones (scripts) de aprovisionamiento de datos.

6.2 Estrategia de automatización de pruebas en toda la organización

6.2.1 Realizar una evaluación de los activos y prácticas de automatización de pruebas para identificar áreas de mejora

Al igual que cualquier otra actividad de desarrollo, es importante tener una estrategia para pausar el desarrollo de la SAP y buscar oportunidades para refactorizar la solución. Las áreas que se deben monitorear son la implementación inicial, el mantenimiento y la capacidad de evaluar la solución desde un punto de vista de repetibilidad. Algunas categorías que conviene controlar son cuántas horas se dedican al desarrollo de la SAP, cuántas horas se dedican a reparar la SAP y cuánto tiempo ahorran los probadores en comparación con las pruebas manuales. Un indicador de que algo anda mal es si el tiempo de mantenimiento de la SAP es mayor que el tiempo de prueba manual.

Antes de comenzar con la primera implementación de una SAP, es importante asegurarse de que pueda ejecutarse en su propio entorno, que esté aislado de cambios aleatorios y que los casos de prueba se puedan actualizar y administrar. Tanto la SAP como su infraestructura deben recibir mantenimiento. En el caso de una implementación por primera vez, se necesitan los siguientes pasos básicos:

- Las herramientas de cobertura de código indican qué parte del código utiliza el conjunto de pruebas de componentes y dónde existen brechas que se pueden cubrir mediante pruebas de componentes adicionales.
- La cobertura funcional del SSP se puede determinar creando una matriz de trazabilidad de requisitos, que descubre qué funcionalidad aún no está cubierta por ningún caso de prueba.
- Definir un uso consistente de la infraestructura del SAP en todos los proyectos o en toda la organización.
- Desarrollar una estrategia de gestión de configuración consistente para los conjuntos de pruebas.

- Crear pautas comunes de desarrollo de la SAP aprovechando las mejores prácticas descritas en el Programa de estudios CTAL-TAE, sección 3.1.4.
- Implementar precondiciones. A menudo, no se puede ejecutar una prueba antes de establecer precondiciones. Estas precondiciones pueden incluir la selección de la base de datos correcta o los datos de prueba a partir de los cuales se realizará
- La prueba o la configuración de valores o parámetros iniciales. Muchos de estos pasos de inicialización que se requieren para establecer las condiciones previas de una prueba se pueden automatizar. Esto permite una solución más confiable y segura cuando estos pasos no se pueden omitir antes de ejecutar las pruebas. A medida que las pruebas de regresión se convierten en pruebas automatizadas, estas precondiciones deben ser parte del proceso de automatización de pruebas.

Cuando se producen actualizaciones incrementales de la SAP para nuevas funciones o fines de mantenimiento, se debe considerar lo siguiente:

- Evaluar actualizaciones de herramientas de prueba u otras herramientas de prueba más nuevas que puedan proporcionar capacidades extendidas para la SAP.
- Evaluar formas de optimizar aún más las características y el rendimiento de la SAP.
- Identificar oportunidades para descomponer y modularizar aún más los (scripts) de prueba para mejorar la reutilización.
- Garantizar el conocimiento y la conciencia de los componentes reutilizables y su utilización constante.
- Recopilar evidencia sobre áreas de mejora potenciales y brindar recomendaciones y sus beneficios.
- Evaluar y corregir áreas de superposición funcional. Al automatizar pruebas de regresión existentes, es una buena práctica identificar cualquier superposición funcional que exista entre los casos de prueba y, cuando sea posible, reutilizar los componentes de automatización de pruebas desarrollados previamente.
- Evaluar casos de prueba manuales adicionales para tener la oportunidad de automatizarlos y crear elementos pendientes para su implementación.
- Resaltar las oportunidades de mejora en el diseño de pruebas y la gestión de datos de pruebas.
- Asegúrese de que todos los conjuntos de pruebas existentes estén adaptados a la última versión de la SAP.
- Si la automatización de pruebas está provocando que el flujo de trabajo se ponga en cola, reduzca el alcance de las pruebas integradas a las más críticas, es decir, cree un conjunto de pruebas de simulación. El conjunto de pruebas de regresión más grande se puede activar por separado o ejecutar a pedido.

7 Referencias

7.1 Estándares

Los estándares para la automatización de pruebas incluyen, entre otras, las siguientes

El Automatic Test Markup Language (ATML) del IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) consta de:

- Norma IEEE 1671.1: Descripción de la prueba
- IEEE Std 1671.2: Descripción del instrumento
- IEEE Std 1671.3: Descripción de la UUT
- Norma IEEE 1671.4: Descripción de la configuración de prueba
- IEEE Std 1671.5: Descripción del adaptador de prueba
- IEEE Std 1671.6: Descripción de la estación de pruebas
- IEEE Std 1641: Definición de señales y pruebas
- Norma IEEE 1636.1: Resultados de las pruebas

ISO/IEC 30130:2016 (E) Ingeniería del software - Capacidades de las herramientas de prueba desoftware.

La Notación de Control de Pruebas y Ensayos (TTCN-3) del ETSI (Instituto Europeo de Normas de Telecomunicación) y la UIT (Unión Internacional de Telecomunicaciones), que consta de:

- ES 201 873-1: Lengua básica de TTCN-3
- ES 201 873-2: Formato de presentación tabular (TFT) TTCN-3
- ES 201 873-3: Formato de presentación gráfica (GFT) de TTCN-3
- ES 201 873-4: Semántica operativa de TTCN-3
- ES 201 873-5: Interfaz de tiempo de ejecución TTCN-3 (TRI)
- ES 201 873-6: Interfaz de control TTCN-3 (TCI)
- ES 201 873-7: Uso de ASN.1 con TTCN-3
- ES 201 873-8: Uso de IDL con TTCN-3
- ES 201 873-9: Uso de XML con TTCN-3
- ES 201 873-10: Documentación TTCN-3
- ES 202 781: Extensiones: Soporte de configuración y despliegue
- ES 202 782: Extensiones: Pruebas de rendimiento y tiempo real de TTCN-3
- ES 202 784: Extensiones: Parametrización avanzada
- ES 202 785: Extensiones: Tipos de comportamiento
- ES 202 786: Extensiones: Soporte de interfaces con señales continuas
- ES 202 789: Extensiones: TRI ampliado

El perfil de pruebas UML (UTP) de OMG (Object Management Group) especifica conceptos de especificación de pruebas para:

- Arquitectura de prueba
- Datos de la prueba
- Comportamiento de la prueba
- Registro de pruebas
- Gestión de pruebas

7.2 Documentos ISTQB®

Identificador	Referencia
ISTQB-AL-TM	ISTQB Certified Tester, Advanced Level Syllabus, Test Manager, Version 2.0, Octubre 2012, disponible en [ISTQB-Web].
ISTQB-EL-TM-MTT	ISTQB Certified Tester, Expert Level Test Management Managing the Test Team, Versión 2.0, Noviembre 2011, disponible en [ISTQB-Web].
ISTQB-FL	ISTQB Certified Tester, Foundation Level Syllabus, Version 4.0, Abril 2023, disponible en [ISTQB-Web].
ISTQB-PT	ISTQB Certified Tester, Performance Testing Syllabus, diciembre de 2018, disponible en [ISTQB-Web].
ISTQB-TAE	ISTQB Certified Tester, Test Automation Engineering Syllabus, febrero de 2024, disponible en [ISTQB-Web].
ISTQB-Glosario	Glosario de términos ISTQB, disponible en línea en [ISTQB-Web].

7.3 Libros

Paul Baker, Zhen Ru Dai, Jens Grabowski e Ina Schieferdecker, "Model-Driven Testing: Using the UML Testing Profile", Springer 2008 edition, ISBN-10: 3540725628, ISBN-13: 978-3540725626

Efriede Dustin, Thom Garrett, Bernie Gauf, "Implementing Automated Software Testing: how to save time and lower costs while raising quality", Addison-Wesley, 2009, ISBN 0-321-58051-6

Efriede Dustin, Jeff Rashka, John Paul, "Automated Software Testing: introduction, management, and performance", Addison-Wesley, 1999, ISBN-10: 0201432870, ISBN-13: 9780201432879

Mark Fewster, Dorothy Graham, "Experiencias de automatización de pruebas: Case Studies of Software Test Automation", Addison-Wesley, 2012.

Mark Fewster, Dorothy Graham, "Automatización de pruebas de software: Effective use of test execution tools", ACM Press Books, 1999, ISBN-10: 0201331403, ISBN-13: 9780201331400

Boby Jose, "Test Automation, a manager's guide", septiembre de 2021, ISBN: 9781780175478

James D. McCaffrey, ".NET Test Automation Recipes: A Problem-Solution Approach", APRESS, 2006 ISBN-13: 978-1-59059-663-3, ISBN-10: 1-59059-663-3

Daniel J. Mosley, Bruce A. Posey, "Just Enough Software Test Automation", Prentice Hall, 2002, ISBN- 10: 0130084689, ISBN-13: 9780130084682

Casey Rosenthal, "Ingeniería del caos: System Resiliency in Practice por Casey Rosenthal",
abril de 2020, ISBN-13: 1492043869

Colin Willcock, Thomas Deiß, Stephan Tobies y Stefan Keil, "An Introduction to TTCN-3"
Wiley, 2ª edición 2011, ISBN-10: 0470663065, ISBN-13: 978-0470663066

7.4 Artículos y Páginas Web

Robert V. Binder, Suzanne Miller, "Five Keys to Effective Agile Test Automation for Government Programs" 24 de agosto de 2017, Software Engineering Institute, Carnegie Mellon University,
https://resources.sei.cmu.edu/asset_files/Webinar/2017_018_101_503516.pdf

DoD CIO, Modern Software Practices "DevSecOps Fundamentals Guidebook: Activities & Tools",
Versión 2.2, Mayo 2023,
https://dodcio.defense.gov/Portals/0/Documents/Library/DevSecOpsActivitesToolsGuidebookTables.pdf?ver=_Sylg1WJB9K0Jxb2XTvzDQ%3d%3d

Naveen Jayachandran, "Understanding roi metrics for software test automation", 2005,
<https://digitalcommons.usf.edu/cgi/viewcontent.cgi?article=3937&context=etd>

Thomas Pestak, William Rowell, PhD, "Prácticas y dificultades de las pruebas de software automatizadas", septiembre de 2018,
https://www.afit.edu/stat/statcoe_files/Automated%20Software%20Testing%20Practices%20and%20Pitfalls%20Rev%201.pdf

Andrew Pollner, Jim Simpson, Jim Wisnowski, "Automated Software Testing Implementation Guide for Managers and Practitioners", octubre de 2018,
https://www.afit.edu/stat/statcoe_files/0214simp%20%20AST%20IG%20for%20Managers%20and%20Practitioners.pdf

8 Apéndice 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 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	Notas
Clasifique las herramientas de prueba en función de su finalidad y de las actividades de prueba a las que dan soporte.	
Compare los distintos niveles de prueba.	Puede utilizarse para buscar similitudes, diferencias o ambas.
Diferenciar las pruebas de la depuración.	Busca diferencias entre conceptos.
Distinguir entre riesgos del proyecto y del producto.	Permite clasificar dos (o más) conceptos por separado.
Explicar el impacto del contexto en el proceso de prueba.	
Dé ejemplos de por qué son necesarias las pruebas.	
Inferir la causa raíz de los defectos a partir de un perfil dado de fallos.	
Resuma las actividades del proceso de revisión del producto del trabajo.	

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:

Ejemplos	Notas
Aplice el análisis de valores límite para derivar casos de prueba a partir de requisitos dados.	Debe referirse a un procedimiento / técnica / proceso, etc.
Aplicar métodos de recopilación de métricas para apoyar los requisitos técnicos y de gestión.	
Preparar el entorno de automatización de pruebas.	
Utilice la trazabilidad para supervisar el progreso de las pruebas y comprobar su integridad y coherencia con los objetivos, la estrategia y el plan de pruebas.	Podría utilizarse en una LO que quiere que el candidato sea capaz de utilizar una técnica o procedimiento. Similar a "aplicar".

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

9 Apéndice 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.

[illegible]

Especialista en estrategia de automatización de pruebas de resultados empresariales			BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11	BO - 12	BO - 13	BO - 14	BO - 15
Capítulo 3	Preparación para la automatización de pruebas																
3.1	Integración en todos los niveles de pruebas																
3.1.1	Diferenciar las distribuciones de automatización de pruebas	2				X											
3.1.2	Seleccionar un enfoque de automatización de pruebas basado en la arquitectura del sistema bajo prueba	3				X											
3.1.3	Demostrar formas de optimizar la distribución de automatización de pruebas para lograr enfoques de desplazamiento a la izquierda y a la derecha	2				X											
3.2	Consideraciones estratégicas en los distintos modelos de ciclo de vida del desarrollo de software																
3.2.1	Explicar cómo los proyectos de automatización de pruebas se ajustan a los modelos de ciclo de vida de desarrollo de software heredados.	2					X										
3.2.2	Explique cómo se ajustan los proyectos de automatización de pruebas a Prácticas recomendadas de desarrollo ágil de software compatibles con la automatización de pruebas	2					X										
3.2.3	Prepárese para que los proyectos de automatización de pruebas se ajusten a las mejores prácticas de DevOps que respaldan la automatización de pruebas en las pruebas continuas	3					X										
3.3	Aplicabilidad y viabilidad de la automatización de pruebas																
3.3.1	Explicación de los criterios para determinar la idoneidad de las pruebas para la automatización de pruebas	2						X									
3.3.2	Identificar los retos que sólo la automatización de pruebas puede abordar	2						X									
3.3.3	Identificar condiciones de prueba difíciles de automatizar	2						X									

Especialista en estrategia de automatización de pruebas deresultados empresariales			BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11	BO - 12	BO - 13	BO - 14	BO - 15
Capítulo 4	Estrategias organizativas de despliegue y lanzamiento para la automatización de pruebas																
4.1	Planificación de soluciones de automatización de pruebas																
4.1.1	Identificar formas en las que la automatización de pruebas ayuda a reducir el tiempo de comercialización	2							X								
4.1.2	Identificar las formas en que la automatización de pruebas ayuda a verificar los defectos notificados de acuerdo con los requisitos	2							X								
4.1.3	Definir enfoques que permitan el desarrollo de escenarios relevantes desde el punto de vista operativo para la automatización de pruebas.	2							X								
4.2	Estrategias de implantación																
4.2.1	Definir una estrategia de despliegue de automatización de pruebas	2								X							
4.2.2	Identificar los riesgos de la automatización de pruebas en la implantación	2								X							
4.2.3	Definir enfoques para mitigar los riesgos de implantación	2								X							
4.3	Dependencias en el entorno de pruebas																
4.3.1	Definición de componentes de automatización de pruebas en el entorno de pruebas	2									X						
4.3.2	Identificar los componentes de infraestructura y las dependencias de la automatización de pruebas	2									X						
4.3.3	Definir los datos de automatización de pruebas y los requisitos de interfaz para la integración en el sistema sometido a prueba.	2									X						
Capítulo 5	Análisis del impacto de la automatización de pruebas																
5.1	Inversión en la creación y el mantenimiento de la automatización de pruebas ^a																

Especialista en estrategia de automatización de pruebas deresultados empresariales			BO - 01	BO - 02	BO - 03	BO - 04	BO - 05	BO - 06	BO - 07	BO - 08	BO - 09	BO - 10	BO - 11	BO - 12	BO - 13	BO - 14	BO - 15
5.1.1	Demuestre el retorno de la inversión al crear una solución de automatización de pruebas	3										X					
5.2	Métricas de automatización de pruebas																
5.2.1	Clasificación de métricas para la automatización de pruebas	2											X				
5.3	El valor de la automatización de pruebas a nivel de proyecto y de organización																
5.3.1	Identificar las consideraciones organizativas para el uso de la automatización de pruebas	3												X			
5.3.2	Analizar las características del proyecto que ayudan a determinar la implementación óptima de la automatización de pruebas Objetivos de la prueba	3												X			
5.4	Decisiones tomadas a partir de informes de automatización de pruebas																
5.4.1	Analizar los datos de los informes de pruebas para fundamentar la toma de decisiones	2													X		
Capítulo 6	Estrategias de aplicación y mejora de la automatización de pruebas																
6.1	Actividades de transición de las pruebas manuales a las pruebas continuas																
6.1.1	Describe los factores y las actividades de planificación en la transición de las pruebas manuales a la automatización de pruebas	2														X	
6.1.2	Describe los Factores y las Actividades de Planificación en la Transición de la Automatización de Pruebas a las Pruebas Continuas	2														X	
6.2	Estrategia de automatización de pruebas en toda la organización																
6.2.1	Realización de una evaluación de los activos y prácticas de automatización de pruebas para identificar áreas de mejora	3															X

10 Apéndice C - Notas de la versión

ISTQB® Test Automation Strategy Syllabus 2024 es un nuevo syllabus ISTQB que combina aspectos estratégicos de la versión anterior del syllabus ISTQB Test Automation Engineer de 2016 con actualizaciones adicionales y mejores prácticas actuales para implementar y medir el éxito de la automatización de pruebas. Por esta razón, no hay notas detalladas por capítulo y sección.

11 Apéndice D - Términos específicos de cada ámbito

Término Nombre	Definición
liberación canaria	Una estrategia de despliegue y pruebas destinada a reducir el riesgo y verificar el nuevo software mediante su lanzamiento a sólo unos pocos usuarios.
contenedor	Unidad de software que empaqueta el código y sus dependencias para que el software se ejecute de forma rápida y fiable en distintos entornos.
DevOps	Metodología que integra y automatiza el trabajo de desarrollo de software y operaciones informáticas para mejorar y acortar el ciclo de vida del desarrollo de software.
patrón del modelo de flujo	Una visión de alto nivel del ámbito de trabajo, sus componentes y las interconexiones entre ellos.

12 Índice

Todos los términos están definidos en el Glosario ISTQB® (<http://glossary.istqb.org/>).

cobertura, 16, 21, 24, 25, 27, 37, 40, 43, 44, 46, 48
componente, 22, 23, 24, 25, 26, 29, 30, 40,45,48
condición de prueba, 22
condición previa, 44, 46
conjunto de pruebas, 27, 30, 31, 34, 44, 45, 48
contenedor, 34, 35, 59
desplazamiento a la derecha, 22, 25, 43
desplazamiento a la izquierda, 22, 25, 30, 43
DevOps, 22, 26, 55, 59
enfoque de automatización de pruebas, 22
IAP, 15, 16, 17, 18, 19, 21, 26, 31, 32, 33,35, 36, 38, 43, 45, 48
informe de pruebas, 37, 43
liberación canaria, 25, 59
modelo de flujo patrón, 45, 59
MTAP, 15, 17, 26, 32, 35, 43
nivel de prueba, 22, 23, 39
patrón de diseño, 59
pirámide de pruebas, 22, 39
prueba doble, 22
pruebas API, 22, 23, 24

pruebas contractuales, 22, 23, 24, 25, 36

pruebas de componentes, 22, 23, 25

pruebas de confirmación, 29, 30

puerta de calidad, 29, 30

SAP, 15, 17, 18, 19, 20, 21, 22, 23, 25, 29, 32, 33, 34, 35, 37, 39, 40, 41, 42, 43, 44,45, 47, 48

sistema en prueba, 22, 29

solución de automatización de pruebas, 17, 18, 37