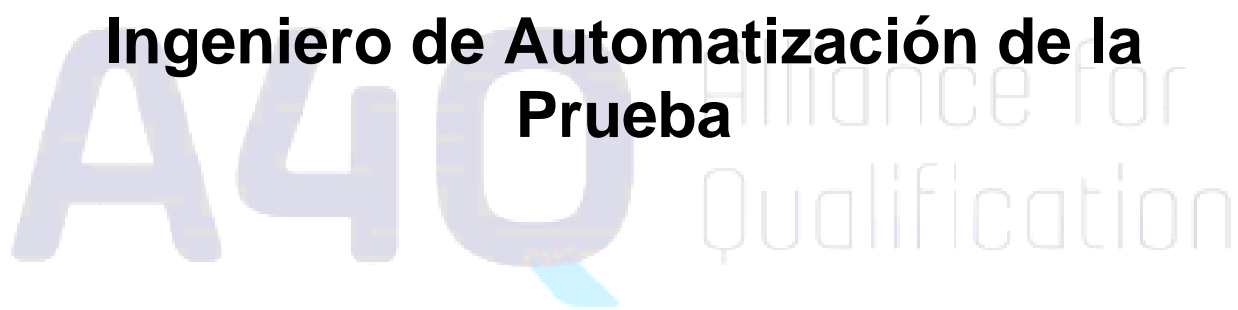


Probador Certificado Programa de Estudio de Nivel Avanzado

Ingeniero de Automatización de la Prueba



Versión 2016

Junta Internacional de Calificaciones de Pruebas
de Software



Aviso de Derecho de Autor

Este documento se puede copiar en su totalidad o se pueden realizar extractos, si la fuente es reconocida.



Probador Certificado

Programa de Estudio de Nivel Avanzado – Ingeniero de
Automatización de la Prueba



Junta Internacional de
Calificaciones de
Pruebas de Software

Copyright © 2018 Junta Internacional de Calificaciones de Pruebas de Software (en adelante, ISTQB®).

Grupo de Trabajo de Automatización de la Prueba de Nivel Avanzado: Bryan Bakker, Graham Bath, Armin Born, Mark Fewster, Jani Haukinen, Judy McKay, Andrew Pollner, Raluca Popescu, Ina Schieferdecker; 2016.



Historial de la Revisión

Versión	Fecha	Observaciones
Borrador Inicial	13AGO2015	Borrador Inicial
Segundo Borrador	05NOV2015	Clasificación y reposicionamiento de los OA (objetivos de aprendizaje)
Tercer Borrador	17DIC2015	OA refinados
Borrador Beta	11ENE2016	Borrador editado
Beta	18MAR2016	Entrega Beta
Programa de Estudio 2016	21OCT2016	Entrega GA

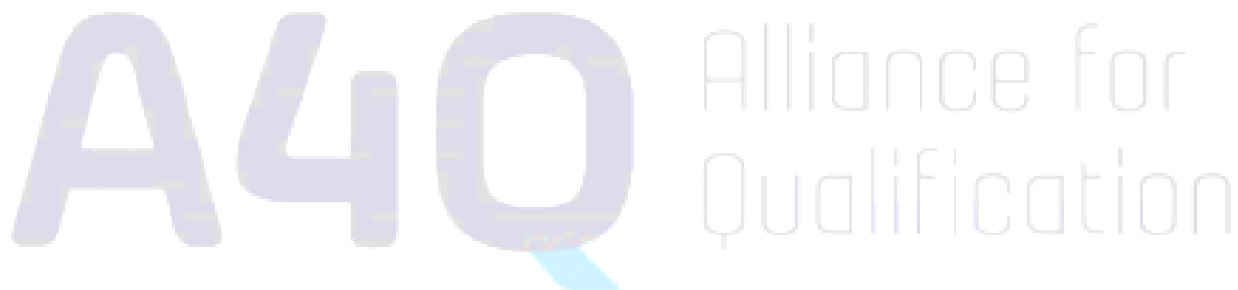


Tabla de Contenidos

Historial de la Revisión.....	3
Tabla de Contenidos	4
Agradecimientos	6
0. Introducción al Programa de Estudio	7
0.1 Objetivo de este Documento	7
0.2 Alcance de este Documento.....	7
0.2.1 Dentro del Alcance	7
0.2.2 Fuera del Alcance.....	7
0.3 El Ingeniero de Automatización de Pruebas Probador Certificado de Nivel Avanzado	8
0.3.1 Expectativas	8
0.3.2 Requisitos para el Ingreso y Renovación.....	8
0.3.3 Nivel de Conocimientos.....	8
0.3.4 Examen	8
0.3.5 Acreditación.....	8
0.4 Partes Normativas versus Informativas	9
0.5 Nivel de Detalles	9
0.6 Cómo está Organizado este Programa de Estudio	9
0.7 Términos, Definiciones y Acrónimos	9
1. Introducción y Objetivos para la Automatización de la Prueba - 30 minutos	11
1.1 Objetivo de la Automatización de la Prueba.....	12
1.2 Factores de Éxito en la Automatización de la Prueba.....	13
2. Preparación para la Automatización de la Prueba - 165 minutos	16
2.1 Factores del SSP que Influyen en la Automatización de la Prueba	17
2.2 Evaluación y Selección de Herramientas	18
2.3 Diseño para la Capacidad de ser Probado y la Automatización - 270 minutos	20
3. La Arquitectura de Automatización de la Prueba Genérica - 270 minutos	22
3.1 Introducción a la AAPg	23
3.1.1 Descripción General de la AAPg.....	24
3.1.2 Capa de Generación de la Prueba.....	26
3.1.3 Capa de Definición de la Prueba.....	26
3.1.4 Capa de Ejecución de la Prueba	26
3.1.5 Capa de Adaptación de la Prueba	27
3.1.6 Gestión de Configuración de una SAP.....	27
3.1.7 Gestión de Proyecto de una SAP	
3.1.8 Soporte de SAP para la Gestión de Prueba	27
3.2 Diseño de AAP.....	28
3.2.1 Introducción al Diseño de AAP.....	28
3.2.2 Enfoques para Automatizar los Casos de Prueba	31
3.2.3 Consideraciones Técnicas del SSP	36
3.2.4 Consideraciones para el Desarrollo/Procesos de Aseguramiento de la Calidad	37
3.3 Desarrollo de SAP	38
3.3.1 Introducción al Desarrollo de SAP	38
3.3.2 Compatibilidad entre la SAP y el SSP.....	39
3.3.3 Sincronización entre la SAP y el SSP	40
3.3.4 Reutilización de la Compilación de la SAP	42

Probador Certificado

Programa de Estudio de Nivel Avanzado – Ingeniero de
Automatización de la Prueba



Junta Internacional de
Calificaciones de
Pruebas de Software

3.3.5	Soporte para una Variedad de Sistemas Destino - 150 minutos	43
4	Riesgos de Implementación y Contingencias - 150 minutos	44
4.1	Selección del Enfoque de Automatización de la Prueba y Planificación del Despliegue/Introducción	45
4.1.1	Proyecto Piloto	45
4.1.2	Despliegue.....	46
4.1.3	Despliegue de la SAP dentro del Ciclo de Vida del Software	47
4.2	Evaluación de Riesgos y Estrategias de Mitigación	47
4.3	Automatización de Pruebas de Mantenimiento	49
4.3.1	Tipos de Mantenimiento	49
4.3.2	Alcance y Enfoque.....	49
5	Gestión de Información de Automatización de la Prueba y Métricas - 165 minutos	52
5.1	Selección de Métricas de SAP	53
5.2	Implementación de la Medición	56
5.3	Registro de la SAP y del SSP.....	57
5.4	Gestión de Información de Automatización de la Prueba	58
6	Transición de las Pruebas Manuales a un Entorno Automatizado - 120 minutos	60
6.1	Criterios para la Automatización.....	61
6.2	Identificar los Pasos Necesarios para Implementar la Automatización dentro de las Pruebas de Regresión.....	65
6.3	Factores a Tener en Cuenta al Implementar la Automatización dentro de las Pruebas de Nuevas Prestaciones	67
6.4	Factores a Tener en Cuenta al Implementar la Automatización dentro de las Pruebas de Confirmación	68
7	Verificación de la SAP - 120 minutos	69
7.1	Verificación de los Componentes del Entorno de Prueba Automatizado	70
7.2	Verificación del Juego de Pruebas Automatizadas	72
8	Mejora Continua - 150 minutos	74
8.1	Opciones para Mejorar la Automatización de Pruebas	75
8.2	Planificación de la Implementación de la Mejora de la Automatización de la Prueba	77
9	Referencias	79
9.1	Normas	79
9.2	Documentos de la ISTQB	80
9.3	Marcas Registradas	80
9.4	Libros	80
9.5	Referencias a la Web	81
10	Aviso a los Proveedores de Capacitación.....	82
10.1	Tiempos de Capacitación	82
10.2	Ejercicios Prácticos en el Lugar de Trabajo	82
10.3	Reglas para el Aprendizaje Virtual	82
11	Índice	83

Agradecimientos

Este documento fue producido por un equipo central del Grupo de Trabajo de Nivel Avanzado de la Junta Internacional de Calificaciones de Pruebas de Software.

El equipo central agradece al equipo de revisión y a todas las Juntas de Miembros por sus sugerencias y aportes.

En el momento en que se completó el Programa de Nivel Avanzado para este módulo, el Grupo de Trabajo de Nivel Avanzado - Automatización de Pruebas tenía la siguiente membresía: Bryan Bakker, Graham Bath (Presidente del Grupo de Trabajo de Nivel Avanzado), Armin Beer, Inga Birthe, Armin Born, Alessandro Collino, Massimo Di Carlo, Mark Fewster, Mieke Gevers, Jani Haukinen, Skule Johansen, Eli Margolin, Judy McKay (Vice-Presidente del Grupo de Trabajo de Nivel Avanzado), Kateryna Nesmyelova, Mahantesh (Monty) Pattan, Andrew Pollner (Presidente de la Automatización de Pruebas de Nivel Avanzado), Raluca Popescu, Ioana Prundaru, Riccardo Rosci, Ina Schieferdecker, Gil Shekel, Chris Van Bael.

Los autores del equipo central para este programa de estudio: Andrew Pollner (Presidente), Bryan Bakker, Armin Born, Mark Fewster, Jani Haukinen, Raluca Popescu, Ina Schieferdecker.

Las siguientes personas participaron en la revisión, comentarios y votación de este programa de estudio (por orden alfabético): Armin Beer, Tibor Csöndes, Massimo Di Carlo, Chen Geng, Cheryl George, Kari Kakkonen, Jen Leger, Singh Manku, Ana Paiva, Raluca Popescu, Meile Posthuma, Darshan Preet, Ioana Prundaru, Stephanie Ulrich, Erik van Veenendaal, Rahul Verma.

Este documento fue publicado formalmente por la Asamblea General de la ISTQB del 21 de octubre de 2016.

0. Introducción a este Programa de Estudio

0.1 Objetivo de este Documento

Este Programa de estudio constituye la base para la Calificación Internacional de Pruebas de Software en el Nivel Avanzado para Automatización de Pruebas - Ingeniería. La ISTQB proporciona este Programa de estudio de la manera siguiente:

- A las juntas de los miembros, para que lo traduzcan a su idioma local y acrediten a los proveedores de capacitación. Las Juntas Nacionales pueden adaptar el Programa de estudios a sus necesidades lingüísticas particulares y agregar referencias para adaptarse a sus publicaciones locales.
- A la Junta Examinadora, para derivar preguntas de examen en su idioma local adaptadas a los objetivos de aprendizaje para cada módulo.
- Para capacitar a los proveedores, para que produzcan cursos y determinen los métodos de enseñanza apropiados.
- Para los candidatos a la certificación, para que se preparen para el examen de certificación (ya sea como parte de un curso de capacitación o de forma independiente).
- A la comunidad internacional de software e ingeniería de sistemas, para que progrese la profesión de pruebas de software y de sistemas, y como base para libros y artículos.

La ISTQB puede permitir que otras entidades usen este Programa de estudio para otros propósitos, siempre que soliciten y obtengan un permiso previo por escrito de la ISTQB.

0.2 Alcance de este Documento

0.2.1 Dentro del Alcance

Este documento describe las tareas de un ingeniero de automatización de la prueba (IAP) en el diseño, desarrollo y mantenimiento de soluciones de automatización de la prueba. Se centra en los conceptos, métodos, herramientas y procesos para automatizar las pruebas funcionales dinámicas y la relación de esas pruebas con la gestión de prueba, la gestión de la configuración, la gestión de defectos, los procesos de desarrollo de software y el aseguramiento de la calidad.

Los métodos descritos son generalmente aplicables a una variedad de enfoques de ciclo de vida de software (p. ej., ágil, secuencial, incremental, iterativo), tipos de sistemas de software (p. ej., integrados, distribuidos, móviles) y tipos de prueba (pruebas funcionales y no funcionales).

0.2.2 Fuera del Alcance

Los siguientes aspectos están fuera del alcance de este programa de estudio de Automatización de la Prueba - Ingeniería:

- La gestión de prueba, creación automatizada de especificaciones de pruebas y generación automatizada de pruebas.
- Tareas del jefe de automatización de la prueba (JAP) en la planificación, supervisión y ajuste del desarrollo y evolución de las soluciones de automatización de la prueba.
- Detalles específicos de la automatización de las pruebas no funcionales (p. ej., rendimiento).
- Automatización del análisis estático (p. ej., análisis de vulnerabilidad) y herramientas de pruebas estáticas.
- Enseñanza de métodos y programación de ingeniería de software (p. ej., qué normas emplear y

Probador Certificado

Programa de Estudio de Nivel Avanzado – Ingeniero de
Automatización de la Prueba



Junta Internacional de
Calificaciones de
Pruebas de Software

qué habilidades tener para realizar una solución de automatización de la prueba).

- Enseñanza de tecnologías de software (p. ej., qué técnicas de secuencias de comandos utilizar para implementar una solución de automatización de la prueba).
- Selección de productos y servicios de prueba de software (p. ej., qué productos y servicios emplear para una solución de automatización de la prueba).



0.3 El Ingeniero de Automatización de Pruebas Probador Certificado de

0.3.1 *Expectativas*

La calificación de Nivel Avanzado está dirigida a las personas que desean desarrollar los conocimientos y habilidades adquiridos en el Nivel Básico y desarrollar su experiencia en una o más áreas específicas. Los módulos que se ofrecen en el programa de especialista de nivel avanzado cubren una amplia gama de temas de prueba.

Un Ingeniero de Automatización de la Prueba es aquel que tiene un amplio conocimiento de las pruebas en general, y un profundo conocimiento en el área especial de la automatización de la prueba. Una comprensión profunda se define como tener un conocimiento suficiente de la teoría y la práctica de la automatización de la prueba para poder influir en la dirección que toma una organización y/o proyecto al diseñar, desarrollar y mantener soluciones de automatización de prueba para pruebas funcionales.

El documento Descripción General de los Módulos de Nivel Avanzado [ISTQB-AL-Modules] describe los resultados comerciales de este módulo.

0.3.2 *Requisitos para el Ingreso y Renovación*

Los criterios generales de ingreso para el Nivel Avanzado se describen en el sitio Web de la ISTQB [ISTQB-Web], sección de Nivel Avanzado.

Además de estos criterios generales de ingreso, los candidatos deben poseer el certificado ISTQB Nivel Básico [ISTQB-CTFL] para poder hacer el examen de certificación de Ingeniero de Automatización de la Prueba de Nivel Avanzado.

0.3.3 *Nivel de Conocimientos*

Los objetivos de aprendizaje para este programa se incluyen al principio de cada capítulo para una identificación clara. Cada tema en el programa de estudio será examinado de acuerdo con el objetivo de aprendizaje asignado.

Los niveles cognitivos asignados a los objetivos de aprendizaje ("niveles K") se describen en el sitio Web de la ISTQB [ISTQB-Web].

0.3.4 *Examen*

El Examen de Certificación de Nivel Avanzado se basará en este programa de estudios además del Programa de Estudio de Nivel Básico [ISTQB-FL]. Las respuestas a las preguntas del examen pueden requerir el uso de material basado en más de una sección de este programa de estudio.

El formato del examen se describe en el sitio Web de la ISTQB [ISTQB-Web], sección de Nivel Avanzado. En el sitio Web de la ISTQB también se incluye información útil para quienes toman los exámenes.

0.3.5 *Acreditación*

Una Junta de Miembros de la ISTQB puede acreditar a proveedores de capacitación cuyo material del curso siga este programa de estudio.

El sitio Web de la ISTQB [ISTQB-Web], sección de Nivel Avanzado describe las reglas específicas que se aplican a los proveedores de capacitación para la acreditación de cursos.

0.4 Partes Normativas versus Informativas

Las partes normativas del programa de estudio son examinables. Estas son:

- Objetivos de aprendizaje
- Palabras clave

El resto del programa de estudio es informativo y explica en detalle los objetivos de aprendizaje.

0.5 Nivel de Detalles

El nivel de detalle en este programa de estudio permite una enseñanza y un examen coherente a nivel internacional. Para lograr este objetivo, el programa de estudio consiste en:

- Objetivos de aprendizaje para cada área de conocimiento, que se obtiene el resultado de aprendizaje cognitivo y el modo de pensar que se debe alcanzar (estos son normativos).
- Una lista de la información que se va a enseñar, que incluye una descripción de los conceptos clave a enseñar, fuentes como son literatura o normas aceptadas y referencias a fuentes adicionales si son necesarias (estas son informativas).

El contenido del programa de estudio no es una descripción de toda el área de conocimiento de la ingeniería de automatización de la prueba; refleja el nivel de detalle que se cubrirá un curso de capacitación de Nivel Avanzado.

0.6 Cómo está Organizado este Programa de Estudio

Hay ocho capítulos principales. El encabezado de nivel superior muestra el tiempo para el capítulo. Por ejemplo:

3. La Arquitectura de Automatización de la Prueba Genérica 270 minutos

Muestra que se prevé que el Capítulo 3 tenga un tiempo de 270 minutos para enseñar el material en el capítulo. Los objetivos específicos de aprendizaje se enumeran al comienzo de cada capítulo.

0.7 Términos, Definiciones y Acrónimos

Muchos términos utilizados en la literatura de software se usan indistintamente. Los términos que se usan en este Programa de Estudio de Nivel Avanzado están disponibles en el Glosario Estándar de Términos Utilizados en Pruebas Software, publicado por la ISTQB [ISTQB-Glossary].

Cada una de las palabras clave enumeradas al comienzo de cada capítulo en este Programa de Estudio de Nivel Avanzado se define en [ISTQB-Glosario].

Las siguientes siglas se utilizan en este documento:

CLI	Interfaz de Línea de Comandos
EPME	Esfuerzo de Prueba Manual Equivalente
AAPg	Arquitectura de Automatización de Prueba Genérica (para un diseño para las soluciones de automatización de la prueba)
GUI	Interfaz Gráfica de Usuario
SSP	Sistema Sujeto a Prueba, véase también objeto de prueba
AAP	Arquitectura de Automatización de Prueba (creación de una instancia de la AAPg para definir la arquitectura de una SAP)

IAP	Ingeniero de Automatización de la Prueba (la persona responsable del diseño de una AAP, incluida la implementación de la SAP resultante, su mantenimiento y evolución técnica)
TAF	Marco de Trabajo de Automatización de la Prueba (el entorno requerido para la automatización de pruebas, incluidos los archivos y los artefactos de prueba, como las bibliotecas de prueba)
JAP	Jefe de Automatización de la Prueba (la persona responsable de la planificación y supervisión del desarrollo y la evolución de una SAP)
SAP	Solución de Automatización de la Prueba (la realización/implementación de una AAP, incluidos los arneses y artefactos de prueba como las bibliotecas de prueba)
GUI	Interfaz Gráfica de Usuario

1. Introducción y Objetivos para la Automatización de la

Palabras clave

Pruebas de API, pruebas de CLI, pruebas de GUI, sistema sometido a prueba, arquitectura de automatización de la prueba, marco de automatización de la prueba, estrategia de automatización de la prueba, automatización de la prueba, guión de prueba, productos de prueba

Objetivos de Aprendizaje para la Introducción y Objetivos para la Automatización de la Prueba

1.1 Objetivo de la Automatización de la Prueba

ALTA-E-1.1.1 (K2) Explicar los objetivos, ventajas, desventajas, y limitaciones de la automatización de la prueba

1.2 Factores de Éxito en la Automatización de la Prueba

ALTA-E-1.2.1 (K2) Identificar los factores de éxito técnico de un proyecto de automatización de la prueba

1.1 Objetivo de la Automatización de la Prueba

En las pruebas de software, la automatización de la prueba es una o más de las siguientes tareas:

- Uso de herramientas de software especializado para controlar y configurar las condiciones previas de prueba
- Ejecución de pruebas
- Comparación de resultados reales con resultados previstos

Una buena práctica es el software utilizado para la prueba del sistema sometido a prueba (SSP) para minimizar la interferencia. Hay excepciones, por ejemplo, sistemas integrados donde el software de prueba debe implementarse en el SSP.

Se espera que la automatización de las pruebas ayude a ejecutar muchos casos de prueba de manera consistente y repetida en diferentes versiones del SSP y/o entornos. Pero la automatización de la prueba es más que un mecanismo para ejecutar un conjunto de pruebas sin interacción humana. Implica un proceso de diseño de productos de prueba, que incluye:

- Software
- Documentación
- Casos de prueba
- Entornos de prueba
- Datos de prueba

El software de prueba es necesario para las actividades de prueba que incluyen:

- Implementación de casos de prueba automatizados
- Monitorización y control de la ejecución de pruebas automatizadas
- Interpretar, gestionar información y registrar los resultados de las pruebas automatizadas.

La automatización de las pruebas tiene diferentes elementos para interactuar con un SSP:

- Pruebas a través de las interfaces públicas a clases, módulos o bibliotecas del SSP (pruebas de API)
- Pruebas a través de la interfaz de usuario del SSP (p. ej., pruebas de GUI o pruebas de CLI)
- Pruebas a través de un servicio o protocolo

Los objetivos de la automatización de la prueba incluyen:

- Mejora de la eficiencia de las pruebas
- Proporcionar una cobertura de funciones más amplia
- Reducir el costo total de la prueba
- Realizar pruebas que los probadores manuales no pueden realizar
- Acortar el periodo de ejecución de prueba
- Aumentar la frecuencia de prueba/reducir el tiempo requerido para los ciclos de prueba

Las ventajas de la automatización de la prueba incluyen:

- Se pueden ejecutar más pruebas por compilación
- La posibilidad de crear pruebas que no se puedan hacer manualmente (pruebas en tiempo real, remotas, paralelas)
- Las pruebas pueden ser más complejas
- Las pruebas se ejecutan más rápido
- Las pruebas están menos sujetas a errores del operador
- Uso más efectivo y eficiente de los recursos de prueba
- Retroalimentación más rápida con respecto a la calidad del software
- Fiabilidad mejorada del sistema (p. ej., repetibilidad, consistencia)
- Mejora de la consistencia de las pruebas

Las desventajas de la automatización de la prueba incluyen:

- Los costos adicionales están involucrados
- Inversión inicial para configurar la SAP
- Requiere tecnologías adicionales
- El equipo necesita tener habilidades de desarrollo y automatización
- Requisito de mantenimiento continuo de la SAP
- Puede distraerse de los objetivos de las pruebas, p. ej., centrarse en la automatización de los casos de prueba a expensas de la ejecución de pruebas
- Las pruebas se pueden volver más complejas
- Se pueden introducir errores adicionales por la automatización

Las limitaciones de la automatización de la prueba incluyen:

- No todas las pruebas manuales pueden ser automatizadas
- La automatización solo puede verificar los resultados interpretables por la máquina
- La automatización solo puede verificar los resultados reales que pueden ser verificados por un oráculo de prueba automatizado
- No es un reemplazo para las pruebas exploratorias

1.2 Factores de Éxito en la Automatización de la Prueba

Los siguientes factores de éxito se aplican a los proyectos de automatización de prueba que están en operación y, por lo tanto, el enfoque está en las influencias que impactan en el éxito a largo plazo del proyecto. Los factores que influyen en el éxito de los proyectos de automatización de pruebas en la etapa piloto no se tratan aquí.

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

Arquitectura de Automatización de la Prueba (AAP)

La arquitectura de automatización de la prueba (AAP) está muy alineada con la arquitectura de un producto de software. Debe quedar claro qué requisitos funcionales y no funcionales deben apoyar la arquitectura. Generalmente estos serán los requisitos más importantes.

A menudo, la AAP está diseñada para el mantenimiento, el rendimiento y la capacidad de ser aprendido. (Consulte la ISO/IEC 25000: 2014 para obtener detalles de estas y otras características no funcionales). Es útil involucrar a los ingenieros de software que entienden la arquitectura del SSP.

Capacidad de Ser Probado del SSP

El SSP debe estar diseñado para que la capacidad de ser probado sea compatible con las pruebas automatizadas. En el caso de las pruebas de GUI, esto podría significar que el SSP debe desacoplar tanto como sea posible la interacción de la GUI y los datos de la apariencia de la interfaz gráfica. En el caso de las pruebas de API, esto podría significar que más clases, módulos o la interfaz de línea de comandos deben ser expuestos como públicos para que puedan ser probados.

Las partes comprobables del SSP deben enfocarse primero. Por lo general, un factor clave para el éxito de la automatización de las pruebas reside en la facilidad de implementación de guiones de prueba automatizados. Con este objetivo en mente, y también para proporcionar una prueba de concepto exitosa, el Ingeniero de Automatización de la Prueba (IAP) necesita identificar los módulos o componentes del SSP que se pueden probar fácilmente con la automatización y comenzar desde allí.

Estrategia de Automatización de la Prueba

Una estrategia práctica y consistente de automatización de pruebas que aborda la mantenibilidad y la consistencia del SSP.

Puede que no sea posible aplicar la estrategia de automatización de prueba de la misma manera a las partes antiguas y nuevas del SSP. Al crear la estrategia de automatización, tenga en cuenta los costos, beneficios y riesgos de aplicarla a diferentes partes del código.

Se debe tener en cuenta la posibilidad de probar tanto la interfaz de usuario como la API con casos de prueba automatizados para verificar la consistencia de los resultados.

Marco de Trabajo de Automatización de la Prueba (TAF)

Un marco de trabajo de automatización de la prueba (TAF, por sus siglas en inglés) que sea fácil de utilizar, está bien documentado y se puede mantener, admite un enfoque coherente para automatizar las pruebas.

Para establecer un TAF fácil de usar y mantener, se debe hacer lo siguiente:

- Implementar servicios de gestión de información: Los informes de prueba deben proporcionar información (paso/fallo/error/no ejecutado/abortado, estadístico, etc.) sobre la calidad del SSP. La gestión de información debe proporcionar la información a los probadores involucrados, jefes de prueba, desarrolladores, jefes de proyecto y otras partes interesadas para que tengan una visión general de la calidad.
- Habilitar la resolución fácil de problemas: Además de la ejecución y el registro de la prueba, el TAF debe proporcionar una manera fácil de solucionar problemas en las pruebas fallidas. La prueba puede fallar debido a
 - fallos encontrados en el SSP
 - fallos encontrados en la SAP
 - problema con las propias pruebas o con el entorno de prueba.
- Abordar el entorno de prueba adecuadamente: Las herramientas de prueba dependen de la consistencia en el entorno de prueba. Tener un entorno de prueba dedicado es necesario en las pruebas automatizadas. Si no hay control sobre el entorno de prueba y los datos de prueba, es posible que la configuración de las pruebas no cumpla con los requisitos para la ejecución de la prueba y es probable que produzca resultados de ejecución falsos.
- Documentar los casos de prueba automatizados: Los objetivos para la automatización de pruebas deben ser claros, p. ej., qué partes de la aplicación se deben probar, en qué grado y qué atributos se deben probar (funcionales y no funcionales). Esto se debe describir y documentar claramente.
- Rastrear la prueba automatizada: El TAF admitirá el rastreo para que el ingeniero de automatización de la prueba rastree los pasos individuales de los casos de prueba.
- Permitir un mantenimiento fácil: En el mejor de los casos, los casos de prueba automatizados deberían mantenerse fácilmente para que el mantenimiento no consuma una parte significativa del esfuerzo de automatización de la prueba. Además, el esfuerzo de mantenimiento debe ser proporcional a la escala de los cambios realizados en el SSP. Para ello, los casos deben

- ser fácilmente analizables, modificables y ampliables. Además, la reutilización automática del software de prueba debe ser alta para minimizar la cantidad de elementos que requieren cambios.
- Mantener las pruebas automatizadas actualizadas: cuando los requisitos nuevos o modificados hacen que las pruebas o los juegos de pruebas fallen, no deshabilite las pruebas fallidas, corríjalas.
 - Plan de implementación: Asegúrese de que los guiones de prueba se puedan implementar, cambiar y volver a implementar fácilmente.
 - Retire las pruebas según sea necesario: Asegúrese de que los guiones de pruebas automatizadas puedan retirarse fácilmente si ya no son útiles o necesarios.
 - Supervisar y restaurar el SSP: En la práctica real, para ejecutar continuamente un caso de prueba o un conjunto de casos de prueba, el SSP debe monitorearse continuamente. Si el SSP encuentra un error fatal (como un bloqueo), el TAF debe tener la capacidad de recuperarse, omitir el caso actual y reanudar las pruebas con el siguiente caso.

El código de automatización de la prueba puede ser complejo de mantener. No es inusual tener tantos códigos para pruebas como códigos para el SSP. Es por esto que es de suma importancia que el código de prueba sea mantenible. Esto se debe a las diferentes herramientas de prueba que se utilizan, los diferentes tipos de verificación que se utilizan y los diferentes artefactos de productos de prueba que deben mantenerse (tales como los datos de entrada de prueba, oráculos de prueba, informes de prueba).

Teniendo en cuenta estas consideraciones de mantenimiento, además de los elementos importantes que deben hacerse, hay algunos que no deben hacerse, como son:

- No crear un código que sea sensible a la interfaz (es decir, se vería afectado por cambios en la interfaz gráfica o en partes no esenciales de la API).
- No crear una automatización de prueba que sea sensible a los cambios de datos o que tenga una alta dependencia de valores de datos particulares (p. ej., entrada de prueba que dependan de otras salidas de prueba).
- No crear un entorno de automatización que sea sensible al contexto (p. ej., fecha y hora del sistema operativo, parámetros de localización del sistema operativo o el contenido de otra aplicación). En este caso, es mejor usar stubs de prueba según sea necesario para poder controlar el entorno.

Cuanto más factores de éxito se cumplan, más probable será que el proyecto de automatización de la prueba tenga éxito. No todos los factores son necesarios y, en la práctica, rara vez se cumplen todos los factores. Antes de comenzar el proyecto de automatización de la prueba, es importante analizar la posibilidad de éxito del proyecto considerando los factores implementados y los factores que faltan teniendo en cuenta los riesgos del enfoque elegido, así como el contexto del proyecto. Una vez que la AAP está instalada, es importante investigar qué elementos faltan o aún necesitan trabajo.

2. Preparación para la Automatización de la Prueba - 165

Palabras clave

capacidad de ser probado, controlador, nivel of intrusion, stub, herramienta de ejecución de prueba, anzuelo de prueba, jefe de automatización de la prueba

Objetivos de Aprendizaje para la Preparación de la Automatización de la Prueba

2.1 Factores del SSP que Influyen en la Automatización de la Prueba

ALTA-E-2.1.1 (K4) Analizar un sistema sujeto a prueba para determinar la solución de automatización adecuada

2.2 Evaluación y Selección de Herramientas

ALTA-E-2.2.1 (K4) Analizar herramientas de automatización de la prueba para un proyecto dado e informar hallazgos técnicos y recomendaciones

2.3 Diseño para la Capacidad de ser Probado y la Automatización

ALTA-E-2.3.1 (K2) Entender los métodos de "diseño para la capacidad de ser probado" y "diseño para automatización de la prueba" aplicables al SSP



2.1 Factores del SSP que Influyen en la Automatización de la Prueba

Al evaluar el contexto del SSP y su entorno, se deben identificar los factores que influyen en la automatización de las pruebas para determinar una solución adecuada. Estos pueden incluir lo siguiente:

- Interfaces del SSP

Los casos de prueba automatizados invocan acciones sobre el SSP. Para ello, el SSP debe proporcionar interfaces a través de las cuales se puede controlar el SSP. Esto se puede hacer a través de controles de IU, pero también a través de interfaces de software de nivel inferior. Además, algunos casos de prueba pueden interconectarse en el nivel de comunicación (p. ej., mediante TCP/IP, USB o interfaces de mensajería propietarias).

La descomposición del SSP permite que la automatización de la prueba se interconecte con el SSP en diferentes niveles de prueba. Es posible automatizar las pruebas en un nivel específico (p. ej., a nivel de componente y sistema), pero solo cuando el SSP lo admite adecuadamente. Por ejemplo, a nivel de componentes, puede que no haya una interfaz de usuario que pueda usarse para realizar pruebas, por lo que es necesario que existan diferentes interfaces de software (también llamados anzuelos de prueba) disponibles, posiblemente personalizadas.

- Software de terceros

A menudo, el SSP no solo consiste en software escrito en la organización de origen, sino que también puede incluir software proporcionado por terceros. En algunos contextos, este software de terceros puede necesitar pruebas, y si la automatización de pruebas está justificada, puede necesitar una solución de automatización de la prueba diferente, como el uso de una API.

- Niveles de intrusión

Diferentes enfoques de automatización de pruebas (que utilizan diferentes herramientas) tienen diferentes niveles de intrusión. Cuanto mayor sea el número de cambios que se deben realizar en el SSP específicamente para las pruebas automatizadas, mayor será el nivel de intrusión. El uso de interfaces de software dedicadas requiere un alto nivel de intrusión, mientras que el uso de elementos existentes de la interfaz de usuario tiene un nivel más bajo de intrusión. El uso de elementos de hardware del SSP (como teclados, interruptores manuales, pantallas táctiles, interfaces de comunicación) tiene un nivel de intrusión aún mayor.

El problema con niveles más elevados de intrusión es el riesgo de falsas alarmas. La SAP puede presentar fallos que pueden deberse al nivel de intrusión impuesto por las pruebas, pero es probable que no ocurran cuando el sistema de software se está utilizando en un entorno real. Las pruebas con un alto nivel de intrusión suelen ser una solución más sencilla para el enfoque de automatización de la prueba.

- Diferentes arquitecturas del SSP

Diferentes arquitecturas del SSP pueden requerir diferentes soluciones de automatización de la prueba. Se necesita un enfoque diferente para un SSP escrito en C++ que usa tecnología COM que para un SSP escrito en Python. Es posible que estas diferentes arquitecturas se manejen con la misma estrategia de automatización de la prueba, pero eso requiere una estrategia híbrida con la capacidad de soportarlas.

- Tamaño y complejidad del SSP

Tenga en cuenta el tamaño y la complejidad del SSP actual y los planes para el desarrollo futuro. Para un SSP pequeño y simple, es posible que no se justifique un enfoque de automatización de pruebas complejo y ultra flexible. Un enfoque simple puede ser más adecuado. Por el contrario,

puede que no sea prudente implementar un enfoque pequeño y simple para un SSP muy grande y complejo. A veces, sin embargo, es apropiado comenzar con uno pequeño y simple, incluso para un SSP complejo, pero este debe ser un enfoque temporal (consulte el Capítulo 3 para obtener más detalles).

Se conocen varios factores descritos aquí (p. ej., tamaño y complejidad, interfaces de software disponibles) cuando el SSP ya está disponible, pero la mayoría de las veces el desarrollo de la automatización de la prueba debe comenzar antes de que el SSP esté disponible. Cuando esto sucede, es necesario estimar varias cosas o el IAP puede especificar las interfaces de software que se necesitan. (Consulte la Sección 2.3 para más detalles).

Incluso cuando el SSP aún no existe, puede comenzar la planificación de la automatización de pruebas. Por ejemplo:

- Cuando se conocen los requisitos (funcionales o no funcionales), los candidatos para la automatización se pueden seleccionar de esos requisitos junto con la identificación de los medios para probarlos. La planificación para la automatización puede comenzar para esos candidatos, incluida la identificación de los requisitos para la automatización y la determinación de la estrategia de automatización de la prueba.
- Cuando se desarrolla la arquitectura y el diseño técnico, se puede llevar a cabo el diseño de interfaces de software que soporten las pruebas.

2.2 Evaluación y Selección de Herramientas

La responsabilidad principal del proceso de selección y evaluación de la herramienta le corresponde al Jefe de Automatización de la Prueba (JAP). Sin embargo, el IAP participará en el suministro de información al JAP y en la realización de muchas de las actividades de evaluación y selección. El concepto del proceso de selección y evaluación de la herramienta se introdujo en el Nivel Básico y se describen más detalles de este proceso en el Nivel Avanzado - Programa de Estudio de Jefe de Pruebas [ISTQB-AL-TM].

El IAP participará a lo largo del proceso de selección y evaluación de la herramienta, pero tendrá contribuciones particulares que hacer en las siguientes actividades:

- Evaluar la madurez de la organización e identificación de oportunidades para el soporte de la herramienta de prueba
- Evaluar los objetivos apropiados para el soporte de la herramienta de prueba
- Identificar y recopilar información sobre herramientas potencialmente adecuadas
- Análisis de información de herramientas contra objetivos y restricciones de proyecto
- Estimar la relación costo-beneficio basada en un caso de negocios sólido
- Hacer una recomendación sobre la herramienta adecuada
- Identificar la compatibilidad de la herramienta con componentes del SSP

Las herramientas de automatización de pruebas funcionales con frecuencia no pueden satisfacer todas las expectativas o las situaciones que se presentan en un proyecto de automatización. El siguiente es un conjunto de ejemplos de estos tipos de problemas (pero definitivamente no es una lista completa):

Encontrar	Ejemplos	Soluciones posibles
La interfaz de la herramienta no funciona con otras herramientas que ya están establecidas	<ul style="list-style-type: none"> La herramienta de gestión de pruebas se ha actualizado y la interfaz de conexión ha cambiado La información del soporte de preventa fue incorrecta y no todos los datos pueden transferirse a la herramienta de gestión de información 	<ul style="list-style-type: none"> Preste atención a las notas de la entrega antes de cualquier actualización, y para pruebas de grandes migraciones antes de migrarlas a producción Intente obtener una demostración en el sitio de la herramienta que usa el SSP real Busque soporte en el vendedor y/o en los foros de la comunidad
Algunas dependencias del SSP se cambian a unas que no son compatibles con la herramienta de prueba	<ul style="list-style-type: none"> El departamento de desarrollo ha actualizado a la versión más reciente de Java 	<ul style="list-style-type: none"> Sincronice las actualizaciones para el entorno de desarrollo/de prueba y herramienta de automatización de la prueba
El objeto en la GUI no pudo ser capturado	<ul style="list-style-type: none"> El objeto es visible pero la herramienta de automatización de prueba no puede interactuar con él 	<ul style="list-style-type: none"> Trate de usar solo tecnologías bien conocidas u objetos en desarrollo Haga un proyecto piloto antes de comprar una herramienta de automatización de prueba Haga que los desarrolladores definan estándares para objetos
La herramienta se ve muy complicada	<ul style="list-style-type: none"> La herramienta tiene un gran conjunto de prestaciones, pero solo se utilizará una parte 	<ul style="list-style-type: none"> Trate de encontrar una manera de limitar el conjunto de prestaciones eliminando las no deseadas de la barra de herramientas Seleccione una licencia para satisfacer sus necesidades Trate de encontrar herramientas alternativas que estén más enfocadas en la funcionalidad requerida
Conflictos con otros sistemas	<ul style="list-style-type: none"> Después de la instalación de otro software, la herramienta de automatización de pruebas ya no funcionará o viceversa 	<ul style="list-style-type: none"> Lea las notas de la entrega o los requisitos técnicos antes de instalar Obtenga la confirmación del proveedor de que no repercutirá en otras herramientas Pregunte a los foros de la comunidad de usuarios

Probador Certificado

Programa de Estudio de Nivel Avanzado – Ingeniero de
Automatización de la Prueba



Junta Internacional de
Calificaciones de
Pruebas de Software

Impacto en el SSP	<ul style="list-style-type: none">• Durante/después del uso de la herramienta de automatización de prueba, el SSP reacciona de manera diferente (p. ej., mayor tiempo de respuesta)	<ul style="list-style-type: none">• Utilice una herramienta que no necesite cambiar el SSP (p. ej., instalación de bibliotecas, etc.)
Acceso al código	<ul style="list-style-type: none">• La herramienta de automatización de la prueba cambiará partes del código fuente	<ul style="list-style-type: none">• Utilice una herramienta que no necesite cambiar el código fuente (p. ej., instalación de bibliotecas, etc.)



Encontrar	Ejemplos	Soluciones posibles
Recursos limitados (principalmente en entornos integrados)	<ul style="list-style-type: none"> El entorno de prueba tiene recursos libres limitados o se queda sin recursos (p. ej., memoria) 	<ul style="list-style-type: none"> Lea las notas de la entrega y discuta el entorno con el proveedor de la herramienta para obtener confirmación de que esta no dará lugar a problemas Pregunte a los foros de la comunidad de usuarios
Actualizaciones	<ul style="list-style-type: none"> La actualización no migrará todos los datos ni dañará los guiones automatizados existentes, los datos o las configuraciones La actualización necesita un entorno diferente (mejor) 	<ul style="list-style-type: none"> Pruebe la actualización en el entorno de prueba y obtenga confirmación del proveedor de que la migración funcionará Lea los requisitos previos de actualización y decida si la actualización merece la pena Busque ayuda en los foros de la comunidad
Seguridad	<ul style="list-style-type: none"> La herramienta de automatización de pruebas requiere información que no está disponible para el ingeniero de automatización de pruebas 	<ul style="list-style-type: none"> El ingeniero de automatización de pruebas necesita que se le otorgue acceso
Incompatibilidad entre diferentes entornos y plataformas	<ul style="list-style-type: none"> La automatización de pruebas no funciona en todos los entornos/plataformas 	<ul style="list-style-type: none"> Implementar pruebas automatizadas para maximizar la independencia de la herramienta, minimizando así el costo de utilizar múltiples herramientas

2.3 Diseño para la Capacidad de ser Probado y la Automatización

La capacidad de ser probado del SSP (disponibilidad de interfaces de software que admitan pruebas, por ejemplo, que permitan el control y la observabilidad del SSP) debe diseñarse e implementarse en paralelo con el diseño y la implementación de las otras prestaciones del SSP. Esto lo puede hacer el arquitecto de software (ya que la capacidad de ser probado es solo uno de los requisitos no funcionales del sistema), pero a menudo esto se realiza por medio de, o con la participación de, un IAP.

El diseño para la capacidad de ser probado consta de varias partes:

- Observabilidad: El SSP debe proporcionar interfaces que brinden información sobre el sistema. Los casos de prueba pueden usar estas interfaces para verificar, por ejemplo, si el comportamiento esperado es igual al comportamiento real.
- Control(abilidad): El SSP debe proporcionar interfaces que puedan usarse para realizar acciones en el SSP. Pueden ser elementos de la interfaz de usuario, llamadas de función, elementos de comunicación (p. ej., TCP/IP o protocolo USB), señales electrónicas (para interruptores físicos), etc.
- Arquitectura claramente definida: La tercera parte importante del diseño para la capacidad de ser

probado es una arquitectura que proporcione interfaces claras y comprensibles que brinden control y visibilidad en todos los niveles de prueba.

El IAP considera formas en las que se puede probar el SSP, incluidas las pruebas automatizadas, de una manera eficaz (probando las áreas correctas y encontrando errores críticos) y eficiente (sin tomar demasiado esfuerzo). Siempre que se necesiten interfaces de software específicas, deben ser especificadas por el IAP e implementadas por el desarrollador. Es importante definir la capacidad de ser probado y, si es necesario, las interfaces de software adicionales al principio del proyecto, para que el trabajo de desarrollo se pueda planificar y presupuestar.

Algunos ejemplos de interfaces de software que admiten pruebas incluyen:

- Las potentes capacidades de escritura de guiones de las hojas de cálculo modernas.
- La aplicación de stubs o simulacros para imitar software y/o hardware (p. ej., transacciones financieras electrónicas, servicio de software, servidor dedicado, tarjeta electrónica, componente mecánico) que aún no está disponible o es demasiado caro para comprarlo, permite probar el software en ausencia de esa interfaz específica.
- Las interfaces de software (o stubs y controladores) se pueden usar para probar las condiciones de error. Considere un dispositivo con un disco duro interno (HDD). El software que controla este HDD (llamado controlador) debe probarse para detectar fallos o desgaste del HDD. Hacerlo esperando que un HDD falle no es muy eficiente (o confiable). La implementación de interfaces de software que simulan HDD defectuosos o lentos puede verificar que el software del controlador funciona correctamente (p. ej., proporciona un mensaje de error, reintenta).
- Se pueden usar interfaces de software alternativas para probar un SSP cuando todavía no hay una interfaz de usuario disponible (y esto a menudo se considera un mejor enfoque). El software integrado en sistemas técnicos a menudo necesita monitorear la temperatura en el dispositivo y activar una función de enfriamiento para comenzar cuando la temperatura sube por encima de un nivel dado. Esto podría probarse sin que el hardware utilice una interfaz de software para especificar la temperatura.
- Las pruebas de transición de estado se utilizan para evaluar el comportamiento del estado del SSP. Una forma de verificar si el SSP está en el estado correcto es mediante una consulta a través de una interfaz de software personalizada diseñada para este propósito (aunque esto también incluye un riesgo, consulte el nivel de intrusión en la Sección 2.1).

El diseño para la automatización debe tener presente que:

- La compatibilidad con las herramientas de prueba existentes debe establecerse desde el principio.
- El problema de la compatibilidad de la herramienta de prueba es crítico, ya que puede afectar la capacidad de automatizar las pruebas de funcionalidad importante (p. ej., la incompatibilidad con un control de cuadrícula evita que todas las pruebas usen ese control).
- Las soluciones pueden requerir el desarrollo de un código de programa y llamadas a las API.

El diseño para la capacidad de ser probado es de la mayor importancia para un buen enfoque de automatización de la prueba y también puede beneficiar la ejecución manual de la prueba.

3. La Arquitectura de Automatización de la Prueba Genérica -

Palabras clave

captura/reproducción, pruebas guiadas por datos, arquitectura de automatización de pruebas genéricas, pruebas guiadas por palabras clave, escritura de guiones lineales, pruebas basadas en modelos, escritura de guiones guiados por procesos, escritura de guiones estructurados, capa de adaptación de pruebas, arquitectura de automatización de la prueba, marco de trabajo de automatización de la prueba, solución de automatización de la prueba, capa de definición de prueba, capa de ejecución de prueba, capa de generación de prueba

Objetivos de Aprendizaje para la Arquitectura de Automatización de la Prueba Genérica

3.1 Introducción a la AAPg

ALTA-E-3.1.1 (K2) Explicar la estructura de la AAPg

3.2 Diseño de AAP

ALTA-E-3.2.1 (K4) Diseñar la AAP apropiada para un proyecto dado ALTA-E-3.2.2 (K2) Explicar la función que desempeñan las capas dentro de una AAP ALTA-E-3.2.3 (K2) Comprender las consideraciones de diseño para una AAP

ALTA-E-3.2.4 (K4) Analizar los factores de implementación, uso y requisitos de mantenimiento para una SAP determinada

3.3 Desarrollo de SAP

ALTA-E-3.3.1 (K3) Aplicar componentes de la AAP genérica (AAPg) para construir una AAP especializada

ALTA-E-3.3.2 (K2) Explicar los factores a tener en cuenta al identificar la reusabilidad de los componentes

3.1 Introducción a la AAPg

Un ingeniero de automatización de la prueba (IAP) tiene la función de diseñar, desarrollar, implementar y mantener soluciones de automatización de pruebas (SAPs). A medida que se desarrolla cada solución, se deben realizar tareas similares, se deben responder preguntas similares y se deben abordar y priorizar problemas similares. Estos conceptos, pasos y enfoques recurrentes en la automatización de pruebas se convierten en la base de la arquitectura de automatización de pruebas genéricas, llamada en pocas palabras AAPg.

La AAPg presenta las capas, componentes e interfaces de una AAPg, que luego se redefinen en la AAP concreta para un SAP específica. Permite un enfoque estructurado y modular para construir una solución de automatización de prueba al:

- Definir el espacio conceptual, las capas, los servicios y las interfaces de un SAP para permitir la realización de los SAP tanto por parte de la empresa como por componentes desarrollados externamente.
- Soportar componentes simplificados para el desarrollo eficaz y eficiente de la automatización de pruebas.
- Reutilizar componentes de automatización de pruebas para SAP diferentes o en evolución para líneas de productos de software y familias y para tecnologías y herramientas de software.
- Facilitar el mantenimiento y evolución de las SAP
- Definir las características esenciales para un usuario de una SAP

Una SAP consiste en el entorno de prueba (y sus artefactos) y los juegos de prueba (un conjunto de casos de prueba que incluyen datos de prueba). Se puede utilizar un marco de trabajo de automatización de la prueba (TAF) para realizar una SAP. Proporciona soporte para la realización del entorno de prueba y proporciona herramientas, arneses de prueba o bibliotecas de soporte.

Se recomienda que la AAP de una SAP cumpla con los principios siguientes que apoyan el desarrollo fácil, la evolución y el mantenimiento de la SAP:

- Responsabilidad única: Cada componente de la SAP debe tener una responsabilidad única, y esa responsabilidad debe estar encapsulada completamente en el componente. En otras palabras, cada componente de una SAP debe estar a cargo de exactamente una cosa, p. ej., generar palabras clave o datos, crear escenarios de prueba, ejecutar casos de prueba, registrar resultados, generar informes de ejecución.
- Extensión (Consulte, p. ej., el principio de abierto/cerrado de B. Myer): Cada componente de la SAP debe estar abierto a la extensión, pero cerrado a la modificación. Este principio significa que debe ser posible modificar o enriquecer el comportamiento de los componentes sin romper la funcionalidad compatible con versiones anteriores.
- Sustitución (consulte, p. ej., el principio de sustitución de B. Liskov): Cada componente de la SAP debe ser reemplazable sin afectar el comportamiento general de la SAP. El componente puede ser reemplazado por uno o más componentes sustitutos, pero el comportamiento mostrado debe ser el mismo.
- Segregación de componentes (Consulte, p. ej., el principio de segregación de interfaces de R.C. Martin): Es mejor tener componentes más específicos que un componente general de usos múltiples. Esto facilita la sustitución y el mantenimiento al eliminar las dependencias innecesarias.
- Inversión de dependencia: Los componentes de una SAP deben depender de abstracciones en lugar de detalles de bajo nivel. En otras palabras, los componentes no deben depender de escenarios de prueba automatizados específicos.

Normalmente, una SAP basada en la AAPg se implementará mediante un conjunto de herramientas, sus complementos y/o componentes. Es importante tener en cuenta que el AAPg es neutral con respecto a los proveedores: No predefina método concreto, tecnología o herramienta para la realización de una SAP.

La AAPg se puede implementar mediante cualquier enfoque de ingeniería de software, p. ej., estructurado, orientado a objetos, orientado a servicios, guiado por modelos, así como por cualquier tecnología y herramienta de software. De hecho, una SAP a menudo se implementa utilizando herramientas estándar, pero normalmente necesitará más adiciones y/o adaptaciones específicas del SSP.

Otras pautas y modelos de referencia relacionados con la SAP son estándares de ingeniería de software para el SDLC seleccionado (Ciclo de Vida del Desarrollo de Software), tecnologías de programación, estándares de formato, etc. No está dentro del alcance de este programa enseñar ingeniería de software en general, sin embargo, se espera que un IAP tenga habilidades, experiencia y pericia en ingeniería de software.

Además, un IAP debe conocer los estándares de codificación y documentación de la industria y las mejores prácticas para hacer uso de ellos al desarrollar una SAP. Estas prácticas pueden aumentar la mantenibilidad, la confiabilidad y la seguridad de la SAP. Tales estándares son típicamente específicos del dominio. Los estándares populares incluyen:

- MISRA para C o C++
- Estándar de codificación JSF para C++
- Reglas de AUTOSAR para MathWorks Matlab/Simulink®

3.1.1 Descripción General de la AAPg

La AAPg está estructurada en capas horizontales para lo siguiente:

- Generación de prueba
- Definición de prueba
- Ejecución de prueba
- Adaptación de la prueba

La AAPg (ver Figura 1: Arquitectura de automatización de prueba genérica) abarca lo siguiente:

- La capa de generación de prueba que admite el diseño manual o automatizado de casos de prueba. Proporciona los medios para diseñar casos de prueba.
- Capa de definición de prueba que admite la definición e implementación de casos y juegos de prueba. Separa la definición de prueba de las tecnologías y herramientas del SSP y/o del sistema de prueba. Contiene medios para definir pruebas de alto y bajo nivel, que se manejan en los datos de prueba, casos de prueba, procedimientos de prueba y componentes de la biblioteca de prueba o combinaciones de los mismos.
- La capa de ejecución de prueba que admite la ejecución de casos de prueba y el registro de prueba. Proporciona una herramienta de ejecución de pruebas para ejecutar las pruebas seleccionadas automáticamente y un componente de registro y gestión de información.
- La capa de adaptación de prueba que proporciona el código necesario para adaptar las pruebas automatizadas para los distintos componentes o interfaces del SSP. Proporciona diferentes adaptadores para conectarse al SSP a través de APIs, protocolos, servicios y otros.
- También tiene interfaces para la gestión de proyectos, la gestión de la configuración y la gestión de pruebas en relación con la automatización de pruebas. Por ejemplo, la interfaz entre la gestión de prueba y la capa de adaptación de prueba se ocupa de la selección y configuración de los adaptadores adecuados relacionados con la configuración de prueba seleccionada.

Las interfaces entre las capas de AAPg y sus componentes son típicamente específicas y, por lo tanto, no dan más explicaciones aquí.

Es importante entender que estas capas pueden estar presentes o ausentes en cualquier SAP dada. Por ejemplo:

- Si la ejecución de la prueba se va a automatizar, es necesario utilizar la ejecución de la prueba y las capas de adaptación de la prueba. No es necesario que se separen y se pueden realizar juntas, p. ej., en marcos de trabajo de prueba unitaria.

- Si la definición de prueba se va a automatizar, es necesaria la capa de definición de prueba.
- Si la generación de prueba se va a automatizar, es necesaria la capa de generación de la prueba.

La mayoría de las veces, uno comenzaría con la implementación de una SAP de abajo hacia arriba, pero otros enfoques como la generación de pruebas automatizadas para pruebas manuales también pueden ser útiles. En general, se recomienda implementar la SAP en pasos incrementales (p. ej., en sprints) para usar la SAP tan pronto como sea posible y para demostrar el valor agregado de la SAP. Además, las pruebas de concepto se recomiendan como parte del proyecto de automatización de la prueba.

Cualquier proyecto de automatización de la prueba debe entenderse, configurarse y gestionarse como un proyecto de desarrollo de software y requiere una gestión de proyectos dedicada. La gestión de proyectos para el desarrollo del TAF (es decir, el soporte de automatización de pruebas para toda una empresa, familias de productos o líneas de productos) se puede separar de la gestión de proyectos para la SAP (es decir, automatización de pruebas para un producto concreto).

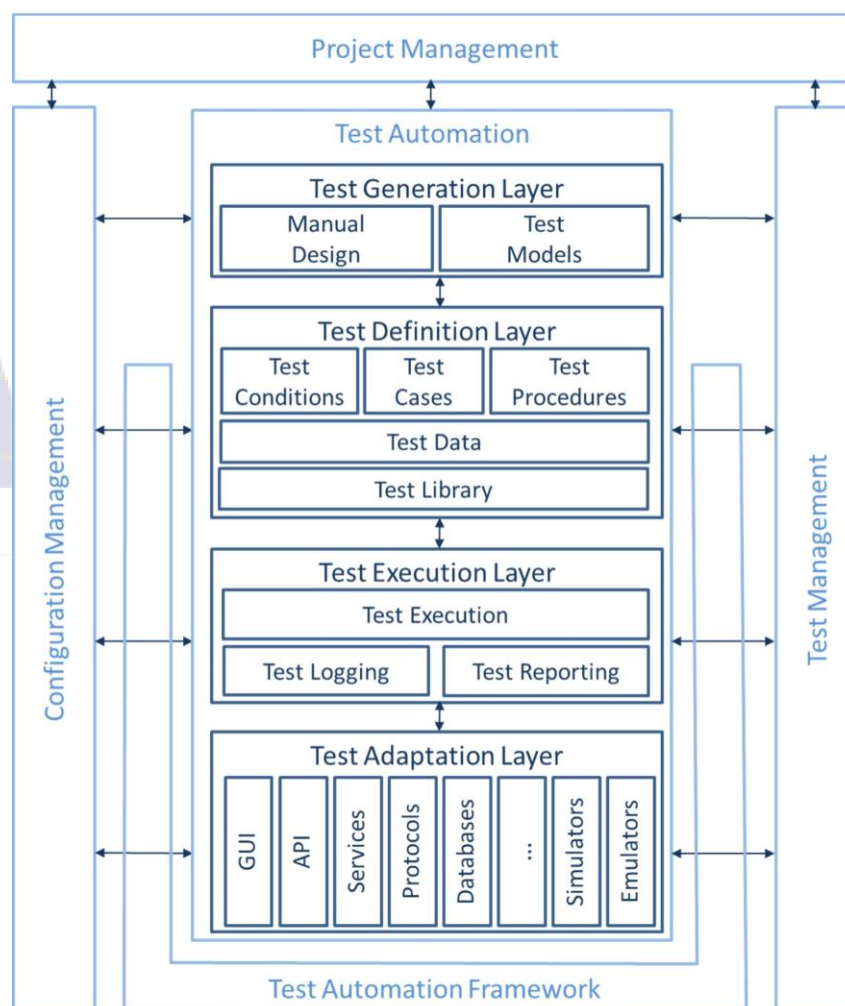


Figura 1: La Arquitectura de Automatización de la Prueba Genérica

3.1.2 Capa de Generación de la Prueba

La capa de generación de la prueba consta de soporte de herramientas para lo siguiente:

- Diseño manual de casos de prueba
- Desarrollar, capturar o derivar datos de prueba
- Generación automática de casos de prueba a partir de modelos que definen el SSP y/o su entorno (es decir, pruebas automatizadas basadas en modelos)

Los componentes de esta capa se utilizan para:

- Editar y navegar por las estructuras del juego de prueba
- Relacionar casos de prueba con objetivos de prueba o requisitos del SSP
- Documentar el diseño de prueba

Para la generación automatizada de pruebas, también se pueden incluir las siguientes prestaciones:

- Capacidad para modelar el SSP, su entorno y/o el sistema de prueba
- Capacidad para definir directivas de prueba y para configurar/parametrizar algoritmos de generación de prueba
- Capacidad para rastrear las pruebas generadas hasta el modelo (elementos)

3.1.3 Capa de Definición de la Prueba

La capa de definición de la prueba consta de soporte de herramientas para lo siguiente:

- Especificar casos de prueba (a nivel alto y/o bajo)
- Definir datos de prueba para casos de prueba de bajo nivel
- Especificar procedimientos de prueba para un caso de prueba o un conjunto de casos de prueba
- Definir guiones para la ejecución de los casos de prueba
- Proporcionar acceso a las bibliotecas de prueba según sea necesario (p. ej., en enfoques guiados por palabras clave)

Los componentes de esta capa se utilizan para:

- Segmentar/restringir, parametrizar o instanciar datos de prueba
- Especificar secuencias de prueba o comportamientos de prueba perfeccionados (incluidas las sentencias de control y las expresiones), para parametrizarlas y/o agruparlas
- Documentar los datos de prueba, casos de prueba y/o procedimientos de prueba.

3.1.4 Capa de Ejecución de la Prueba

La capa de ejecución de la prueba consta de soporte de herramientas para lo siguiente:

- Ejecutar de casos de prueba automáticamente
- Registrar las ejecuciones de casos de prueba
- Gestionar información de los resultados de la prueba

La capa de ejecución de la prueba puede estar integrada por componentes que brinden las siguientes capacidades:

- Configurar y desarmar el SSP para la ejecución de la prueba
- Configurar y desarmar juegos de pruebas (es decir, un conjunto de casos de prueba que incluye datos de prueba)
- Configurar y parametrizar la configuración de prueba
- Interpretar los datos de prueba y los casos de prueba y transformarlos en guiones ejecutables
- Instalar el sistema de prueba y/o el SSP para el registro (filtrado) de la ejecución de la prueba y/o para la inyección de fallos

- Analizar las respuestas del SSP durante la ejecución de la prueba para dirigir las ejecuciones de prueba subsiguientes
- Validar las respuestas del SSP (comparación de resultados esperados y reales) para resultados de ejecución de casos de prueba automatizados
- Controlar la ejecución automatizada de las pruebas a tiempo

3.1.5 Capa de Adaptación de la Prueba

La capa de adaptación de la prueba consta de soporte de herramientas para lo siguiente:

- Controlar el arnés de prueba
- Interactuar con el SSP
- Monitorizar el SSP
- Simular o emular el entorno del SSP

La capa de adaptación de la prueba proporciona la siguiente funcionalidad:

- Mediar entre las definiciones de prueba neutrales tecnológicamente y los requisitos tecnológicos específicos del SSP y los dispositivos de prueba
- Aplicar diferentes adaptadores específicos de tecnología para interactuar con el SSP
- Distribuir la ejecución de prueba en múltiples dispositivos de prueba/interfaces de prueba o ejecutar pruebas localmente

3.1.6 Gestión de Configuración de una SAP

Comúnmente, una SAP se está desarrollando en varias iteraciones/versiones y debe ser compatible con las iteraciones/versiones del SSP. La administración de configuración de una SAP puede necesitar incluir:

- Modelos de prueba
- Definiciones/especificaciones de prueba, incluidos datos de prueba, casos de prueba y bibliotecas
- Guiones de prueba
- Motores de ejecución de la prueba y herramientas y componentes complementarios
- Adaptadores de prueba para el SSP
- Simuladores y emuladores para el entorno del SSP
- Resultados de la prueba y gestión de información de la prueba

Estos elementos constituyen el software de prueba y deben estar en la versión correcta para que coincida con la versión del SSP. En algunas situaciones, puede ser necesario volver a las versiones anteriores de la SAP, p. ej., en caso de que los problemas de campo deban reproducirse con versiones anteriores del SSP. Una buena gestión de la configuración permite esta capacidad.

3.1.7 Gestión de Proyecto de una SAP

Como cualquier proyecto de automatización de la prueba es un proyecto de software, requiere la misma gestión de proyectos que cualquier otro proyecto de software. Un IAP debe realizar las tareas para todas las fases de la metodología SDLC establecida al desarrollar la SAP. Además, un IAP debe comprender que el entorno de desarrollo de la SAP debe diseñarse de manera tal que la información de estado (métrica) se pueda extraer fácilmente o que se informe automáticamente a la administración del proyecto de la SAP.

3.1.8 Soporte de SAP para la Gestión de Prueba

Una SAP debe soportar la gestión de pruebas para el SSP. Los informes de prueba, incluidos los registros de prueba y los resultados de las pruebas, deben extraerse de forma fácil o automática para la gestión de prueba (personas o sistema) del SSP.

3.2 Diseño de AAP

3.2.1 Introducción al Diseño de AAP

Hay una serie de actividades principales requeridas para diseñar una AAP, que se pueden ordenar de acuerdo con las necesidades del proyecto de automatización de pruebas u organización. Estas actividades se discuten en las siguientes secciones. Se pueden necesitar más o menos actividades dependiendo de la complejidad de la AAP.

Requisitos de captura necesarios para definir una AAP apropiada

Los requisitos para un enfoque de automatización de prueba deben considerar lo siguiente:

- Qué actividad o fase del proceso de prueba debe ser automatizada, p. ej., gestión de prueba, diseño de prueba, generación de prueba o ejecución de prueba. Tenga en cuenta que la automatización de prueba refina el proceso de prueba fundamental al insertar la generación de prueba entre el diseño de prueba y la implementación de prueba.
- Qué nivel de prueba se debe admitir, p. ej., nivel de componente, nivel de integración, nivel de sistema
- Qué tipo de prueba se debe admitir, p. ej., pruebas funcionales, pruebas de conformidad, pruebas de interoperabilidad
- Qué función de prueba debe ser compatible, p. ej., ejecutor de prueba, analista de prueba, arquitecto de prueba, jefe de prueba
- Qué producto de software, línea de productos de software, familia de productos de software debe ser compatible, p. ej., para definir la duración y la vida útil de la AAP implementada
- Qué tecnologías de SSP deben ser compatibles, p. ej., para definir la AAP en función de la compatibilidad con las tecnologías de SSP

Comparar y contrastar diferentes enfoques de diseño/arquitectura

El IAP debe analizar los pros y los contras de los diferentes enfoques al diseñar las capas seleccionadas de la AAP. Estos incluyen pero no se limitan a:

Consideraciones para la capa de generación de prueba:

- Selección de generación de pruebas manuales o automatizadas
- Selección de, p. ej., generación de pruebas basadas en requisitos, basadas en datos, basadas en escenarios o basadas en el comportamiento
- Selección de estrategias de generación de pruebas (p. ej., cobertura de modelos, como los árboles de clasificación para enfoques basados en datos, cobertura de casos de uso/excepción para enfoques basados en escenarios, cobertura de transición/estado/ruta para enfoques basados en el comportamiento, etc.)
- Selección de la estrategia de selección de prueba. En la práctica, la generación de pruebas combinatorias completas es inviable, ya que puede conducir a la explosión de un caso de prueba. Por lo tanto, los criterios de cobertura práctica, las ponderaciones, las evaluaciones de riesgo, etc. se deben utilizar para guiar la generación de pruebas y la posterior selección de pruebas.

Consideraciones para la capa de definición de la prueba:

- Selección de definición de prueba guiada por datos, por palabras clave, basada en patrones o en modelos
- Selección de la notación para la definición de la prueba (p. ej., tablas, notación basada en el estado, notación estocástica, notación de flujo de datos, notación de proceso de negocio, notación basada en escenarios, etc. mediante el uso de hojas de cálculo, lenguajes de prueba específicos para el dominio, las Pruebas y la Notación del Control de Pruebas (TTCN-3), el perfil de prueba UML (UTP), etc.)
- Selección de guías de estilo y pautas para la definición de pruebas de alta calidad

- Selección de repositorios de casos de prueba (hojas de cálculo, bases de datos, archivos, etc.)
- Consideraciones para la capa de ejecución de la prueba:
 - Selección de la herramienta de ejecución de pruebas
 - Selección de la interpretación (mediante el uso de una máquina virtual) o enfoque de compilación para implementar procedimientos de prueba; esta opción generalmente depende de la herramienta de ejecución de prueba elegida
 - Selección de la tecnología de implementación para implementar procedimientos de prueba (imperativo, como C; funcional, como Haskell o Erlang; orientado a objetos, como C++, C#, Java; escritura de guiones, como Python o Ruby, o una Tecnología específica para herramientas); esta elección suele depender de la herramienta de ejecución de prueba seleccionada
 - Selección de bibliotecas auxiliares para facilitar la ejecución de la prueba (p. ej., bibliotecas de dispositivos de prueba, bibliotecas de codificación/decodificación, etc.)

Consideraciones para la capa de adaptación de la prueba:

- Selección de interfaces de prueba para el SSP
- Selección de herramientas para estimular y observar las interfaces de prueba
- Selección de herramientas para monitorizar el SSP durante la ejecución de la prueba
- Selección de herramientas para rastrear la ejecución de la prueba (p. ej., incluyendo el tiempo de la ejecución de la prueba)

Identificar áreas donde la abstracción puede producir beneficios

La abstracción en una AAP permite la independencia tecnológica, ya que el mismo conjunto de pruebas se puede utilizar en diferentes entornos de prueba y en diferentes tecnologías de destino. La portabilidad de los artefactos de prueba se incrementa. Además, se garantiza la neutralidad del proveedor, lo que evita los efectos de bloqueo para una SAP. La abstracción también mejora la mantenibilidad y la adaptabilidad a las tecnologías del SSP nuevas o en evolución. Además, la abstracción ayuda a hacer que una AAP (y sus ejemplificaciones por las SAP) sea más accesible para los no técnicos, ya que los juegos de prueba pueden documentarse (incluidos los medios gráficos) y explicarse a un nivel superior, lo que mejora la legibilidad y la comprensibilidad.

El IAP debe discutir con las partes interesadas en el desarrollo de software, el aseguramiento de la calidad y las pruebas, sobre qué nivel de abstracción usar en qué área de la SAP. Por ejemplo, ¿Qué interfaces de la adaptación de la prueba y/o capa de ejecución de la prueba necesitan ser externalizadas, definidas formalmente y mantenidas estables durante toda la vida útil de la AAP? También se debe discutir si se está utilizando una definición de la prueba abstracta o si la AAP utiliza una capa de ejecución de la prueba solo con guiones. Del mismo modo, debe entenderse si la generación de pruebas se abstrae mediante el uso de modelos de prueba y enfoques de prueba basados en modelos. El IAP debe ser consciente de que existen compromisos entre implementaciones sofisticadas y sencillas de una AAP con respecto a la funcionalidad general, la mantenibilidad y la capacidad de expansión. Una decisión sobre qué abstracción usar en una AAP debe tener en cuenta estas compensaciones.

Mientras más abstracción se utilice para una AAP, más flexible es con respecto a una mayor evolución o transición a nuevos enfoques o tecnologías. Esto alcanza el costo de inversiones iniciales más grandes (p. ej., arquitectura y herramientas de automatización de pruebas más complejas, mayores requisitos de conjunto de habilidades, curvas de aprendizaje más grandes), lo que retrasa el punto de equilibrio inicial pero puede ser rentable a largo plazo. También puede llevar a un menor rendimiento de la SAP.

Si bien las consideraciones detalladas sobre el retorno de la inversión (ROI) son responsabilidad del JAP, el IAP debe proporcionar aportes al análisis del retorno de la inversión al proporcionar evaluaciones técnicas y comparaciones de diferentes arquitecturas y enfoques de automatización de pruebas con respecto a los plazos, costos, esfuerzos y beneficios.

Entender las tecnologías del SSP y cómo éstas se interconectan con la SAP

El acceso a las interfaces de prueba del SSP es fundamental para cualquier ejecución de prueba

automatizada. El acceso puede estar disponible en los siguientes niveles:

- El nivel de software, p. ej., el SSP y el software de prueba están vinculados
- Nivel de API, p. ej., la SAP invoca las funciones/operaciones/métodos proporcionados en una interfaz de programación de aplicaciones (remota)
- Nivel de protocolo, p. ej., la SAP interactúa con el SSP a través de HTTP, TCP, etc.
- Nivel de servicio, p. ej., la SAP interactúa con los servicios del SSP a través de servicios web, servicios RESTful, etc.

Además, el IAP debe decidir sobre el paradigma de interacción de la AAP que se utilizará para la interacción entre la SAP y el SSP, siempre que la SAP y el SSP estén separados por el API, protocolos o servicios. Estos paradigmas incluyen los siguientes:

- Paradigma guiado por eventos, que guía la interacción a través de eventos que se intercambian en un bus de eventos.
- Paradigma cliente-servidor, que guía la interacción a través de la invocación del servicio de los solicitantes de servicios al proveedor de servicios.
- Paradigma de igual a igual, que guía la interacción a través de la invocación del servicio desde cualquier igual.

A menudo, la selección del paradigma depende de la arquitectura del SSP y puede tener implicaciones en la arquitectura del SSP. La interconexión entre el SSP y la AAP debe analizarse y diseñarse cuidadosamente para elegir una arquitectura segura para el futuro entre los dos sistemas.

Entender el entorno del SSP

Un SSP puede ser un software independiente o software que funciona solo en relación con otro software (p. ej., sistemas de sistemas), hardware (p. ej., sistemas integrados) o componentes ambientales (p. ej., sistemas cibernéticos). Una SAP simula o emula el entorno del SSP como parte de una configuración de prueba automatizada.

Los ejemplos de entornos de prueba y usos de muestras incluyen los siguientes:

- Una computadora con SSP y SAP: útil para probar una aplicación de software
- Computadoras en red individuales para un SSP y SAP respectivamente: útiles para probar el software del servidor
- Dispositivos de prueba adicionales para estimular y observar las interfaces técnicas de un SSP: útiles para probar el software, p. ej., en un decodificador
- Dispositivos de prueba conectados en red para emular el entorno operativo del SSP: útil para probar el software de un enrutador de red
- Simuladores para simular el entorno físico del SSP: útil para probar el software de una unidad de control integrada

Tiempo y complejidad para una implementación determinada de la arquitectura de productos de prueba

Si bien la estimación del esfuerzo para un proyecto de SAP es responsabilidad de un JAP, un IAP debe apoyar al JAP en esta tarea, proporcionando buenas estimaciones para el tiempo y la complejidad de un diseño de AAP. Los métodos y ejemplos de estimaciones incluyen los siguientes:

- Estimación basada en analogías, como son los puntos de funciones, estimación de tres puntos, delphi de banda ancha y estimación por expertos
- Estimación por el uso de estructuras de descomposición del trabajo, como las que se encuentran en el software de administración o en las plantillas de proyectos.
- Estimación paramétrica como el Modelo de Costo Constructivo (COCOMO por sus siglas en inglés)
- Estimaciones basadas en el tamaño, como el Análisis de puntos de función, el Análisis de puntos de historias o el Análisis de casos de uso
- Estimaciones grupales como la Planificación póker

Facilidad de uso para una implementación determinada de la arquitectura de software de prueba

Además de la funcionalidad de la SAP, su compatibilidad con el SSP, su estabilidad y capacidad de evolución a largo plazo, sus requisitos de esfuerzo y consideraciones de ROI, un IAP tiene la responsabilidad específica de abordar los problemas de usabilidad para una SAP. Esto incluye, pero no se limita a:

- Diseño orientado al probador
- Facilidad de uso de la SAP
- Soporte de SAP para otras funciones en el desarrollo de software, aseguramiento de la calidad y gestión de proyectos.
- Organización efectiva, navegación y búsqueda en/con la SAP
- Documentación útil, manuales y texto de ayuda para la SAP
- Informes prácticos por y sobre la SAP
- Diseños iterativos para abordar la retroalimentación de la SAP y las perspectivas empíricas

3.2.2 Enfoques para Automatizar los Casos de Prueba

Los casos de prueba deben traducirse en secuencias de acciones que se ejecutan contra un SSP. Esa secuencia de acciones se puede documentar en un procedimiento de prueba y/o se puede implementar en un guión de prueba. Además de las acciones, los casos de prueba automatizados también deben definir datos de prueba para la interacción con el SSP e incluir pasos de verificación para comprobar que el SSP logró el resultado esperado. Se puede utilizar varios enfoques para crear la secuencia de acciones:

1. El IAP implementa casos de prueba directamente en guiones automatizados. Esta opción es la menos recomendada, ya que carece de abstracción y aumenta la carga del mantenimiento.
2. El IAP diseña procedimientos de prueba y los transforma en guiones automatizados. Esta opción tiene abstracción pero carece de automatización para generar los guiones.
3. El IAP utiliza una herramienta para traducir los procedimientos de prueba a guiones automatizados. Esta opción combina tanto la abstracción como la generación automatizada de guiones.
4. El IAP utiliza una herramienta que genera procedimientos de prueba automatizados y/o traduce los guiones directamente de los modelos. Esta opción tiene el más alto grado de automatización.

Tenga en cuenta que las opciones dependen en gran medida del contexto del proyecto. También puede ser eficiente iniciar la automatización de pruebas aplicando una de las opciones menos avanzadas, ya que estas suelen ser más fáciles de implementar. Esto puede proporcionar valor agregado a corto plazo, aunque resultará en una solución menos mantenible.

Los enfoques bien establecidos para automatizar los casos de prueba incluyen:

- Enfoque de captura/reproducción, que se puede utilizar para la opción 1
- Enfoque estructurado de escritura de guiones, enfoque guiado por datos y enfoque guiado por palabras clave, que se puede utilizar para la opción 2 o 3
- Pruebas basadas en modelos (incluido el enfoque guiado por procesos), que pueden

utilizarse para la opción 4. Estos enfoques se explican posteriormente en términos de los conceptos principales y las ventajas y desventajas.

Enfoque de captura/reproducción

Concepto principal

En los enfoques de captura/reproducción, las herramientas se utilizan para capturar interacciones con el SSP mientras se realiza la secuencia de acciones tal como se define en un procedimiento de prueba. Las entradas son capturadas; las salidas también se pueden registrar para

verificaciones posteriores. Durante la reproducción de eventos, hay varias posibilidades de verificación de salida manual y automatizada:

- Manual: el probador debe observar las salidas del SSP para detectar anomalías
- Completo: todas las salidas del sistema que se registraron durante la captura deben ser reproducidas por el SSP
- Exacto: todas las salidas del sistema que se grabaron durante la captura deben ser reproducidas por el SSP al nivel de detalle de la grabación
- Completo: todas las salidas del sistema que se registraron durante la captura deben ser reproducidas por el SSP
- Puntos de control: solo los resultados seleccionados del sistema se verifican en ciertos puntos para valores específicos

A favor

El enfoque de captura / reproducción se puede utilizar para SSP en la GUI y/o nivel de API. Inicialmente, es fácil de configurar y usar.

En contra

Los guiones de captura/reproducción son difíciles de mantener y evolucionar porque la ejecución del SSP capturada depende en gran medida de la versión del SSP de la que se tomó la captura. Por ejemplo, cuando se graba en el nivel de la GUI, los cambios en el diseño de la GUI pueden afectar al guión de prueba, incluso si solo es un cambio en la posición de un elemento de la GUI. Por lo tanto, los enfoques de captura/reproducción siguen siendo vulnerables a los cambios

La implementación de los casos de prueba (guiones) solo puede comenzar cuando el SSP está disponible.

Escritura de guiones lineales

Concepto principal

Al igual que con todas las técnicas de escritura de guiones, la escritura de guión lineal comienza con algunos procedimientos de prueba manuales. Sin embargo, tenga en cuenta que estos pueden no ser documentos escritos: el conocimiento sobre qué pruebas ejecutar y cómo ejecutarlos puede ser "conocido" por uno o más Analistas de Pruebas.

Cada prueba se ejecuta manualmente, mientras que la herramienta de prueba graba la secuencia de acciones y, en algunos casos, captura la salida visible del SSP a la pantalla. Esto generalmente resulta en una secuencia de comandos (generalmente grande) para cada procedimiento de prueba. Los guiones grabados pueden editarse para mejorar la legibilidad (p. ej., agregando comentarios para explicar lo que está sucediendo en puntos clave) o agregar controles adicionales utilizando el lenguaje de escritura de guiones de la herramienta.

La herramienta puede volver a reproducir los guiones, lo que hace que la herramienta repita las mismas acciones que realiza el probador cuando se grabó el guión. Aunque esto se puede usar para automatizar las pruebas de GUI, no es una buena técnica para utilizar donde se debe automatizar un gran número de pruebas y se requiere para muchas entregas del software. Esto se debe al alto costo de mantenimiento que generalmente se debe a los cambios en el SSP (cada cambio en el SSP puede requerir muchos cambios en los guiones grabados).

A favor

Las ventajas de los guiones lineales se centran en el hecho de que se requiere poco o ningún trabajo de preparación antes de que pueda comenzar a automatizar. Una vez que haya aprendido a usar la herramienta, es simplemente una cuestión de grabar una prueba manual y volver a reproducirla (aunque la parte de la grabación puede requerir una interacción adicional con la

Probador Certificado

Programa de Estudio de Nivel Avanzado – Ingeniero de
Automatización de la Prueba



Junta Internacional de
Calificaciones de
Pruebas de Software

herramienta de prueba para solicitar que se realicen comparaciones de la salida real con la esperada para verificar el software está funcionando correctamente). Las habilidades de programación no son necesarias, pero suelen ser útiles.



En contra

Las desventajas de los guiones lineales son numerosas. La cantidad de esfuerzo requerido para automatizar cualquier procedimiento de prueba determinado dependerá principalmente del tamaño (número de pasos o acciones) requerido para realizarlo. Por lo tanto, el procedimiento de prueba número 1000 que se automatizará requerirá una cantidad proporcional de esfuerzo similar al procedimiento de prueba número 100. En otras palabras, no hay mucho margen para reducir el costo de compilar nuevas pruebas automatizadas.

Asimismo, si hubiera un segundo guión que realizara una prueba similar aunque con diferentes valores de entrada, ese guión contendría la misma secuencia de instrucciones que el primer guión; solo la información incluida con las instrucciones (conocidas como argumentos o parámetros de instrucciones) diferiría. Si hubiera varias pruebas (y, por lo tanto, guiones), todas tendrían la misma secuencia de instrucciones, las cuales deberían mantenerse siempre que el software cambiara de una manera que afectara a los guiones.

Debido a que los guiones están en un lenguaje de programación, en lugar de un lenguaje natural, los no programadores pueden encontrarlos difíciles de entender. Algunas herramientas de prueba utilizan lenguajes propietarios (exclusivos de la herramienta), por lo que lleva tiempo aprender el lenguaje y dominarlo.

Los guiones grabados contienen solo sentencias generales en los comentarios, si las hay. Los guiones largos en particular se anotan mejor con comentarios para explicar lo que sucede en cada paso de la prueba. Esto facilita el mantenimiento. Los guiones pronto pueden volverse muy grandes (conteniendo muchas instrucciones) cuando la prueba abarca muchos pasos.

Los guiones son no modulares y difíciles de mantener. La escritura de guiones lineales no sigue los paradigmas comunes de reutilización y modularidad del software y está estrechamente relacionada con la herramienta que se está utilizando.

Guión estructurado

Concepto principal

La principal diferencia entre la técnica de guión estructurado y la técnica de escritura de guión lineal es la introducción de una biblioteca de guiones. Esta contiene guiones reutilizables que realizan secuencias de instrucciones que generalmente se requieren en una serie de pruebas. Buenos ejemplos de dichos guiones son aquellos que se relacionan, por ejemplo, con las operaciones de las interfaces del SSP.

A favor

Los beneficios de este enfoque incluyen una reducción significativa en los cambios de mantenimiento requeridos y el costo reducido de automatizar nuevas pruebas (porque pueden usar guiones que ya existen en lugar de tener que crearlos todos desde cero).

Las ventajas del guión estructurado se obtienen en gran medida mediante la reutilización de los guiones. Se pueden automatizar más pruebas sin tener que crear el volumen de guiones que requeriría un enfoque de escritura de guión lineal. Esto tiene un impacto directo en los costos de compilación y mantenimiento. La segunda prueba y las posteriores no tomarán tanto esfuerzo para automatizarse porque algunos de los guiones creados para implementar la primera prueba se pueden reutilizar.

En contra

El esfuerzo inicial para crear los guiones compartidos puede verse como una desventaja, pero esta inversión inicial debería rendir grandes dividendos si se aborda de manera adecuada. Se requerirá habilidades de programación para crear todos los guiones, ya que una simple grabación por sí sola no será suficiente. La biblioteca de guiones debe ser bien gestionada, es decir, los guiones deben estar documentados y los Analistas de Pruebas Técnicas deben encontrarlas fácilmente (de manera que una convención de nomenclatura razonable ayude aquí).

Pruebas guiadas por datos

Concepto principal

La técnica de escritura de guiones guiadas por datos se basa en la técnica de guión estructurado. La diferencia más significativa es cómo se manejan las entradas de prueba. Las entradas se extraen de los guiones y se colocan en uno o más archivos separados (generalmente llamados archivos de datos).

Esto significa que el guión de prueba principal se puede reutilizar para implementar una serie de pruebas (en lugar de una sola prueba). Normalmente, el guión de prueba principal "reutilizable" se denomina guión de "control". El guión de control contiene la secuencia de instrucciones necesarias para realizar las pruebas, pero lee los datos de entrada de un archivo de datos. Se puede usar una prueba de control para muchas pruebas, pero generalmente no es suficiente para automatizar una amplia gama de pruebas. Por lo tanto, se requerirá una cantidad de guiones de control, pero eso es solo una fracción del número de pruebas que se automatizan.

A favor

El costo de agregar nuevas pruebas automatizadas se puede reducir significativamente con esta técnica de escritura de guiones. Esta técnica se utiliza para automatizar muchas variaciones de una prueba útil, proporcionando pruebas más profundas en un área específica y puede aumentar la cobertura de la prueba.

Tener las pruebas "descritas" por los archivos de datos significa que los analistas de pruebas pueden especificar pruebas "automatizadas" simplemente al introducir uno o más archivos de datos. Esto le da a los Analistas de Pruebas más libertad para especificar pruebas automatizadas sin tanta dependencia de los Analistas de Pruebas Técnicas (que pueden ser un recurso escaso).

En contra

La necesidad de administrar los archivos de datos y asegurarse de que sean legibles por la SAP es una desventaja, pero se puede abordar de manera adecuada.

Además, los casos de pruebas negativas importantes pueden pasarse por alto. Las pruebas negativas son una combinación de procedimientos de prueba y datos de prueba. En un enfoque dirigido principalmente a los datos de prueba, se pueden pasar por alto los "procedimientos de prueba negativos".

Pruebas guiadas por palabras clave

Concepto principal

La técnica de escritura de guiones guiadas por palabras clave se basa en la técnica de escritura de guiones guiada por datos. Hay dos diferencias principales: (1) los archivos de datos ahora se llaman archivos de "definición de prueba" o algo similar (p. ej., archivos de palabras de acción); y (2) solo hay un guión de control.

Un archivo de definición de prueba contiene una descripción de las pruebas de una manera que debería ser más fácil de entender para los analistas de pruebas (más fácil que el archivo de datos equivalente). Por lo general, contendrá datos al igual que los archivos de datos, pero los archivos de palabras clave también contienen instrucciones de alto nivel (las palabras clave o "palabras de acción").

Las palabras clave deben elegirse de modo que sean significativas para el analista de pruebas, las pruebas que se describen y la aplicación que se está probando. Estos se utilizan principalmente (pero no exclusivamente) para representar interacciones comerciales de alto nivel con un sistema (p. ej., "realizar pedidos"). Cada palabra clave representa una serie de interacciones detalladas con el sistema bajo prueba. Las secuencias de palabras clave (incluidos los datos de prueba relevantes) se utilizan para especificar los casos de prueba. Se pueden usar palabras clave especiales para los pasos de verificación, o las palabras clave pueden contener tanto las acciones como los pasos de verificación.



El alcance de la responsabilidad de los Analistas de Prueba incluye la creación y el mantenimiento de los archivos de palabras clave. Esto significa que una vez que se implementan las guiones de soporte, los analistas de prueba pueden agregar pruebas "automatizadas" simplemente especificándolas en un archivo de palabras clave (como en la escritura de guiones guiados por datos).

A favor

Una vez que se haya escrito el guión de control y los guiones de soporte para las palabras clave, esta técnica de escritura de guiones reducirá mucho el costo de agregar nuevas pruebas automatizadas.

Tener las pruebas "descritas" por los archivos de palabras clave significa que los analistas de pruebas pueden especificar pruebas "automatizadas" simplemente describiendo las pruebas utilizando las palabras clave y los datos asociados. Esto le da a los Analistas de Pruebas más libertad para especificar pruebas automatizadas sin tanta dependencia de los Analistas de Pruebas Técnicas (que pueden ser un recurso escaso). El beneficio del enfoque guiado por palabras clave sobre el enfoque guiado por datos en este sentido es el uso de las palabras clave. Cada palabra clave debe representar una secuencia de acciones detalladas que producen algún resultado significativo. Por ejemplo, "crear cuenta", "realizar pedido", "verificar el estado del pedido" son todas acciones posibles para una aplicación de compra en línea que involucran una serie de pasos detallados. Cuando un Analista de Pruebas le describe una prueba del sistema a otro analista de pruebas, es probable que hablen en términos de estas acciones de alto nivel, no de los pasos detallados. El objetivo del enfoque guiado por palabras clave es, entonces, implementar estas acciones de alto nivel y permitir que las pruebas se definan en términos de las acciones de alto nivel sin hacer referencia a los pasos detallados.

Estos casos de prueba son más fáciles de mantener, leer y escribir, ya que la complejidad se puede ocultar en las palabras clave (o en las bibliotecas, en caso de un enfoque de guión estructurado). Las palabras clave pueden ofrecer una abstracción de las complejidades de las interfaces del SSP.

En contra

Implementar las palabras clave sigue siendo una gran tarea para los ingenieros de automatización de pruebas, especialmente si se emplea una herramienta que no ofrece soporte para esta técnica de escritura de guiones. Para sistemas pequeños puede ser demasiado costoso de implementar y los costos serían mayores que los beneficios.

Se debe tener cuidado para garantizar que se implementen las palabras clave correctas. Las palabras clave buenas se usarán a menudo con muchas pruebas diferentes, mientras que las palabras clave pobres probablemente se usen solo una vez o solo unas pocas veces.

Escritura de guiones guiada por procesos

Concepto principal

El enfoque guiado por procesos se basa en la técnica de escritura de guiones guiados por palabras clave, con la diferencia de que los escenarios, que representan los casos de uso del SSP y sus variantes, constituyen los guiones que están parametrizadas con datos de prueba o combinados en definiciones de prueba de nivel superior.

Tales definiciones de prueba son más fáciles de manejar, ya que se puede determinar la relación lógica entre acciones, p. ej., "verificar el estado del pedido" después de "hacer el pedido" en la prueba de prestaciones o "verificar el estado del pedido" sin el "orden de colocación" anterior en la prueba de robustez. *A favor*

El uso de una definición de casos de prueba basada en un escenario, similar a un proceso, permite definir los procedimientos de prueba desde una perspectiva de flujo de trabajo. El objetivo del enfoque guiado por procesos es implementar estos flujos de trabajo de alto nivel mediante el uso de bibliotecas de prueba que representan los pasos detallados de la prueba (consulte también el enfoque guiado por palabras clave).

En contra

Es posible que los procesos de un SSP no sean fáciles de comprender por un Analista de Pruebas Técnicas, al igual que la implementación de los guiones guiados por procesos, especialmente si la herramienta no admite ninguna lógica de procesos de negocios.

También se debe tener cuidado para garantizar que se implementen los procesos correctos, mediante el uso de palabras clave correctas. Otros procesos harán referencia a los buenos procesos y darán como resultado muchas pruebas relevantes, mientras que los procesos deficientes no darán resultados en términos de relevancia, capacidad de detección de errores, etc.

Pruebas basadas en modelos

Concepto principal

Las pruebas basadas en modelos se refieren a la generación automatizada de casos de prueba (consulte también el Programa de Estudio de Prueba Basadas de Modelos por la ISTQB), a diferencia de la ejecución automatizada de casos de prueba, mediante el uso de captura/reproducción, escritura de guiones lineales, escritura de guiones guiados por datos o guiados por procesos. Las pruebas basadas en modelos utilizan modelos (semi) formales que se abstraen de las tecnologías de escritura de guiones de AAP. Se pueden utilizar diferentes métodos de generación de pruebas para derivar pruebas para cualquiera de los marcos de escritura de guiones analizados anteriormente.

A favor

Las pruebas basadas en modelos permiten que la abstracción se concentre en la esencia de las pruebas (en términos de lógica de negocios, datos, escenarios, configuraciones, etc. que deben probarse). También permite generar pruebas para diferentes sistemas de destino y tecnologías específicas, de modo que los modelos utilizados para la generación de pruebas constituyen una representación segura del software de prueba que se puede reutilizar y mantener a medida que la tecnología evoluciona.

En caso de cambios en los requisitos, sólo se debe adaptar el modelo de prueba; automáticamente se genera un conjunto completo de casos de prueba. Las técnicas de diseño de casos de prueba se incorporan en los generadores de casos de prueba.

En contra

Se requiere experiencia en modelado para ejecutar un enfoque de prueba basado en modelos de manera efectiva. La tarea de modelar abstrayendo las interfaces, los datos y/o el comportamiento de un SSP puede ser difícil. Además, el modelado y las herramientas de prueba basadas en modelos aún no son una corriente principal, pero están madurando. Los enfoques de prueba basados en modelos requieren ajustes en los procesos de prueba. Por ejemplo, es necesario establecer el papel del diseñador de pruebas. Además, los modelos utilizados para la generación de pruebas constituyen artefactos importantes para el aseguramiento de la calidad de un SSP y también deben garantizarse y mantenerse con la calidad.

3.2.3 Consideraciones Técnicas del SSP

Además, los aspectos técnicos de un SSP se deben tener en cuenta al diseñar una AAP. Algunos de ellos se discuten a continuación, aunque esta no es una lista completa pero debe servir como una muestra de los aspectos importantes.

Interfaces del SSP

Un SSP tiene interfaces internas (dentro del sistema) e interfaces externas (para el entorno del sistema y sus usuarios o por componentes expuestos). Una AAP debe ser capaz de controlar y/o observar todas las interfaces del SSP que pueden verse afectadas por los procedimientos de prueba (es decir, las interfaces deben ser comprobables). Además, también puede ser necesario registrar las interacciones entre el SSP y la SAP con diferentes niveles de detalle, que suelen incluir sellos de hora.

El enfoque de la prueba (p. e., una prueba) es necesario al inicio del proyecto (o de manera continua en entornos ágiles) durante la definición de la arquitectura para verificar la disponibilidad de las interfaces de prueba o las instalaciones de prueba necesarias para que el SSP sea verificable (diseño para la capacidad de ser probado).

Datos del SSP

Un SSP utiliza datos de configuración para controlar su creación de ejemplos, configuración, administración, etc. Además, utiliza datos de usuario que procesa. Un SSP también puede usar datos externos de otros sistemas para completar sus tareas. Dependiendo de los procedimientos de prueba para un SSP, todos estos tipos de datos deben ser definibles, configurables y capaces de ser ejemplificados por la AAP. La forma específica de confrontar los datos del SSP se decide en el diseño de la AAP. Dependiendo del enfoque, los datos pueden manejarse como parámetros, hojas de datos de prueba, bases de datos de prueba, datos reales, etc.

Configuraciones del SSP

Un SSP se puede desplegar en diferentes configuraciones, por ejemplo, en diferentes sistemas operativos, en diferentes dispositivos de destino o con diferentes configuraciones de idioma. Dependiendo de los procedimientos de prueba, es posible que la AAP tenga que abordar diferentes configuraciones del SSP. Los procedimientos de prueba pueden necesitar diferentes configuraciones de prueba (en un laboratorio) o configuraciones de prueba virtuales de la AAP (en la nube), en combinación con una configuración de los datos del SSP. También puede necesitar agregar simuladores y/o emuladores de componentes del SSP seleccionados para aspectos del SSP seleccionados.

Normas del SSP y marcos legales

Además de los aspectos técnicos de un SSP, es posible que el diseño de la AAP deba respetar los requisitos legales y/o normas para diseñar la AAP de una manera compatible. Los ejemplos incluyen los requisitos de privacidad para los datos de prueba o los requisitos de confidencialidad que afectan las capacidades de registro y gestión de información de la AAP.

Herramientas y entornos de herramientas utilizados para desarrollar el SSP

Junto con el desarrollo de un SSP, se pueden utilizar diferentes herramientas para la ingeniería de

requisitos, diseño y modelado, codificación, integración y despliegue del SSP. La AAP, junto con sus propias herramientas, debe tener en cuenta el panorama de herramientas del SSP que permitan la compatibilidad, el rastreo de las herramientas y/o la reutilización de artefactos.

Interfaces de prueba en el producto de software

Se recomienda encarecidamente no eliminar todas las interfaces de prueba antes de la entrega del producto. En la mayoría de los casos, estas interfaces se pueden dejar en el SSP sin que causen problemas al producto final. Cuando se dejan instaladas, los ingenieros de servicio y soporte pueden usar las interfaces para el diagnóstico de problemas y para probar las entregas de mantenimiento. Es importante verificar que las interfaces no presenten riesgos de seguridad. Si es necesario, los desarrolladores generalmente pueden deshabilitar estas interfaces de prueba de manera que no puedan usarse fuera del departamento de desarrollo.

3.2.4 Consideraciones para el Desarrollo/Procesos de Aseguramiento de la Calidad

Los aspectos de procesos de desarrollo y aseguramiento de la calidad de un SSP se deben tener en cuenta al diseñar una AAP. Algunos de ellos se discuten a continuación, aunque esta no es una lista completa pero debe servir como una muestra de los aspectos importantes. Algunos de ellos se discuten a continuación, aunque esta no es una lista completa pero debe servir como una muestra de los aspectos importantes.

Requisitos del control de ejecución de la prueba

Dependiendo del nivel de automatización requerido por la AAP, es posible que la AAP deba soportar la ejecución de la prueba interactiva, la ejecución de la prueba en modo de lotes o la ejecución de la prueba completamente automatizada.

Requisitos de gestión de información

Dependiendo de los requisitos de gestión de la información, incluidos los tipos de informes y sus estructuras, la AAP debe poder admitir informes de prueba fijos, parametrizados o definidos en diferentes formatos y diseños.

Función y los derechos de acceso

Dependiendo de los requisitos de seguridad, es posible que se necesite que la AAP proporcione un sistema de funciones y derechos de acceso.

Entorno de la herramienta establecida

La gestión de proyectos de SSP, la gestión de pruebas, el repositorio de códigos y pruebas, el seguimiento de defectos, la gestión de incidentes, el análisis de riesgos, etc., pueden ser compatibles con herramientas que componen el entorno de la herramienta establecida. La AAP también es compatible con una herramienta o conjunto de herramientas que debe integrarse a la perfección con las otras herramientas en el entorno. Además, los guiones deben almacenarse y versionarse como código del SSP para que las revisiones sigan el mismo proceso para ambos.

3.3 Desarrollo de la SAP

3.3.1 Introducción al Desarrollo de SAP

El desarrollo de una SAP es comparable a otros proyectos de desarrollo de software. Puede seguir los mismos procedimientos y procesos, incluidas las revisiones por pares realizadas por desarrolladores y probadores. Específicos para una SAP son su compatibilidad y sincronización con el SSP. Estos requieren consideración en el diseño de la AAP (consulte la Sección 3.2) y en el desarrollo de la SAP. Además, el SSP se ve afectado por la estrategia de prueba, p. ej., al tener que hacer que las interfaces de prueba estén disponibles para la SAP.

Esta sección utiliza el ciclo de vida de desarrollo de software (SDLC) para explicar el proceso de desarrollo de la SAP y los aspectos relacionados con el proceso de compatibilidad y sincronización con el SSP. Estos aspectos son igualmente importantes para cualquier otro proceso de desarrollo que se haya elegido o esté establecido para el desarrollo del SSP y/o de la SAP; deben adaptarse adecuadamente.

El SDLC básico para la SAP se muestra en la Figura 2.

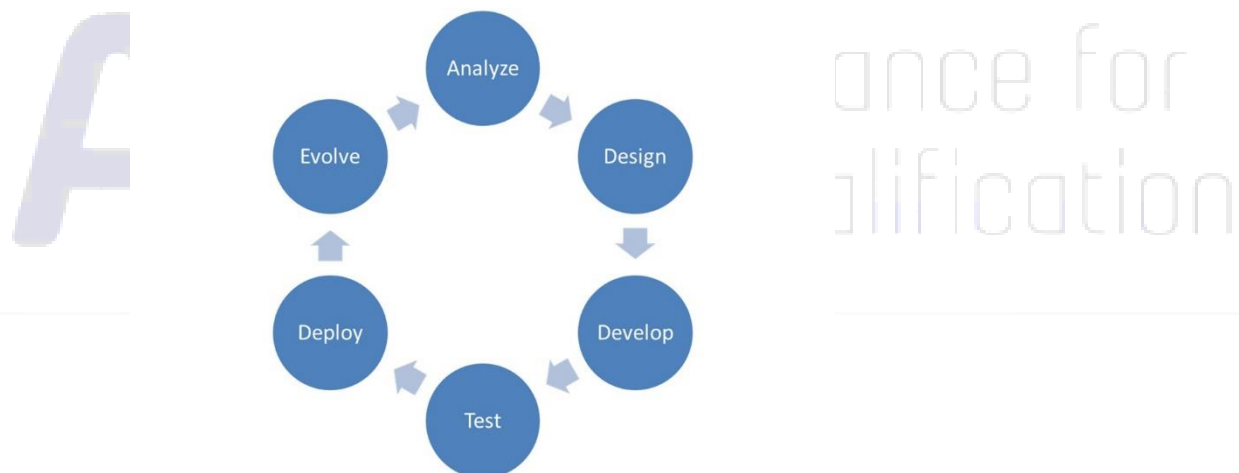


Figura 2: SDLC básico para la SAP

El conjunto de requisitos para una SAP necesita ser analizado y recopilado (véa la Figura 2). Los requisitos guían el diseño de la SAP según lo define su AAP (consulte la Sección 3.2). El diseño se convierte en software mediante enfoques de ingeniería de software. Tenga en cuenta que una SAP también puede utilizar hardware de dispositivo de prueba dedicado, que está fuera de consideración para este programa. Como cualquier otro software, una SAP necesita ser probada. Esto se hace normalmente mediante pruebas de capacidad básica para la SAP que son seguidas por una interacción entre la SAP y el SSP. Después de la implementación y el uso de una SAP, a menudo se necesita una evolución de la SAP para agregar más capacidad de prueba, cambiar las pruebas o actualizar la SAP para que coincida con el cambio de SSP. La evolución de la SAP requiere una nueva ronda de desarrollo de la SAP de acuerdo con el SDLC.

Tenga en cuenta también que el SDLC no muestra la copia de seguridad, el almacenamiento y el desmontaje de una SAP. Al igual que con el desarrollo de SAP, estos procedimientos deben seguir los

métodos establecidos en una organización.

3.3.2 Compatibilidad entre la SAP y el SSP

Compatibilidad de procesos

Las pruebas de un SSP se deben sincronizar con su desarrollo y, en el caso de la automatización de pruebas, se deben sincronizar con el desarrollo de la SAP. Por lo tanto, es ventajoso coordinar los procesos para el desarrollo del SSP, el desarrollo de la SAP y para las pruebas. Se puede lograr una gran ganancia cuando el desarrollo del SSP y de la SAP son compatibles en términos de estructura de procesos, gestión de procesos y soporte de herramientas.

Compatibilidad de equipos

La compatibilidad de equipos es otro aspecto de la compatibilidad entre la SAP y el desarrollo de SSP. Si se utiliza un modo de pensar compatible para abordar y administrar el desarrollo de la SAP y del SSP, ambos equipos se beneficiarán al revisar los requisitos, diseños y/o artefactos de desarrollo de cada uno, al discutir problemas y al encontrar soluciones compatibles. La compatibilidad de equipos también ayuda en la comunicación y la interacción entre sí.

Compatibilidad de tecnologías

Además, debe considerarse la compatibilidad tecnológica entre la SAP y el SSP. Es beneficioso diseñar e implementar una interacción perfecta entre la SAP y el SSP desde el principio. Incluso si eso no es posible (p. ej., debido a que no hay soluciones técnicas disponibles para la SAP o el SSP), puede ser posible una interacción homogénea mediante el uso de adaptadores, envoltorios u otras formas de intermediarios.

Compatibilidad de herramientas

Se debe considerar la compatibilidad de la herramienta entre la gestión de la SAP y del SSP, el desarrollo y el aseguramiento de la calidad. Por ejemplo, si se utilizan las mismas herramientas para la gestión de requisitos y/o la gestión de problemas, el intercambio de información y la coordinación del desarrollo de la SAP y del SSP será más fácil.

3.3.3 Sincronización entre la SAP y el SSP

Sincronización de requisitos

Después de la obtención de requisitos, se deben desarrollar los requisitos del SSP y de la SAP. Los requisitos de la SAP se pueden clasificar en dos grupos principales de requisitos: (1) los requisitos que abordan el desarrollo de la SAP como un sistema basado en software, como los requisitos de las características de la SAP para el diseño de prueba, especificación de prueba, análisis de resultados de prueba, etc. y (2) los requisitos que abordan la prueba del SSP por medio de la SAP. Los llamados requisitos de prueba corresponden a los requisitos del SSP y reflejan todas las características y propiedades del SSP que deben ser probadas por la SAP. Siempre que se actualicen los requisitos del SSP o de la SAP, es importante verificar la coherencia entre los dos y verificar que todos los requisitos del SSP que deben ser probados por la SAP tengan requisitos de prueba definidos.

Sincronización de fases de desarrollo

Para tener la SAP lista cuando sea necesario para probar el SSP, las fases de desarrollo deben coordinarse. Es más eficiente cuando los requisitos, diseños, especificaciones e implementaciones del SSP y de la SAP están sincronizados.

Sincronización del seguimiento de defectos

Los defectos pueden estar relacionados con el SSP, con la SAP o con los requisitos/diseños/especificaciones. Debido a la relación entre los dos proyectos, siempre que se corrija un defecto dentro de uno, la acción correctiva puede impactar al otro. El seguimiento de defectos y las pruebas de confirmación deben abordar tanto la SAP como el SSP.

Sincronización de la evolución del SSP y de la SAP

Tanto el SSP como la SAP pueden evolucionar para adaptarse a nuevas prestaciones o deshabilitar funciones, corregir defectos o abordar cambios en su entorno (incluidos los cambios en el SSP y la SAP, respectivamente, ya que uno es un componente del entorno para el otro). Cualquier cambio aplicado a un SSP o a una SAP puede afectar al otro, por lo que la administración de estos cambios debe abordar tanto al SSP como a la SAP.

Dos enfoques de sincronización entre el SSP y procesos de desarrollo de la SAP se representan en la Figura 3 y la Figura 4.

En la Figura 3 y la Figura 4 se muestran dos enfoques de sincronización entre los procesos de desarrollo para el SSP y para la SAP. (1) el análisis de la SAP se basa en el diseño del SSP, que a su vez se basa en el análisis del SSP y (2) la prueba del SSP hace uso de la SAP desplegada.

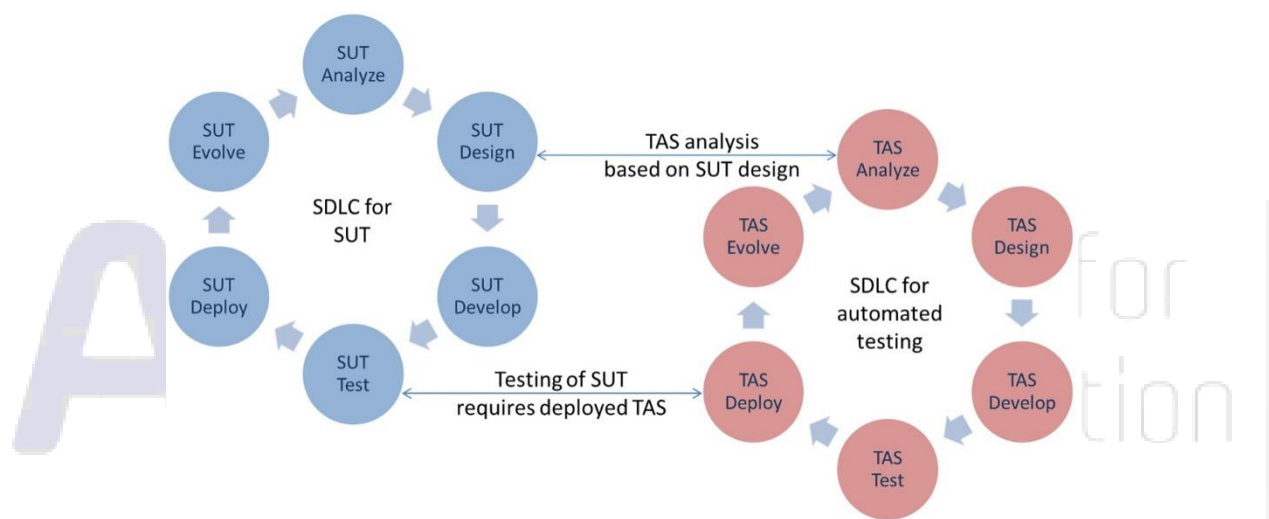


Figura 3: Ejemplo 1 de sincronización de los procesos de desarrollo de la SAP y del SSP

La Figura 4 muestra un enfoque híbrido con pruebas tanto manuales como automatizadas. Siempre que se usen las pruebas manuales antes de que las pruebas sean automatizadas o siempre que las pruebas manuales y automáticas se usen juntas, el análisis de la SAP debe basarse tanto en el diseño del SSP como en las pruebas manuales. De esta manera, la SAP se sincroniza con ambos. El segundo punto de sincronización importante para tal enfoque es como el anterior: la prueba del SSP requiere pruebas implementadas, que en el caso de las pruebas manuales podrían ser los procedimientos de prueba manuales que se deben seguir.

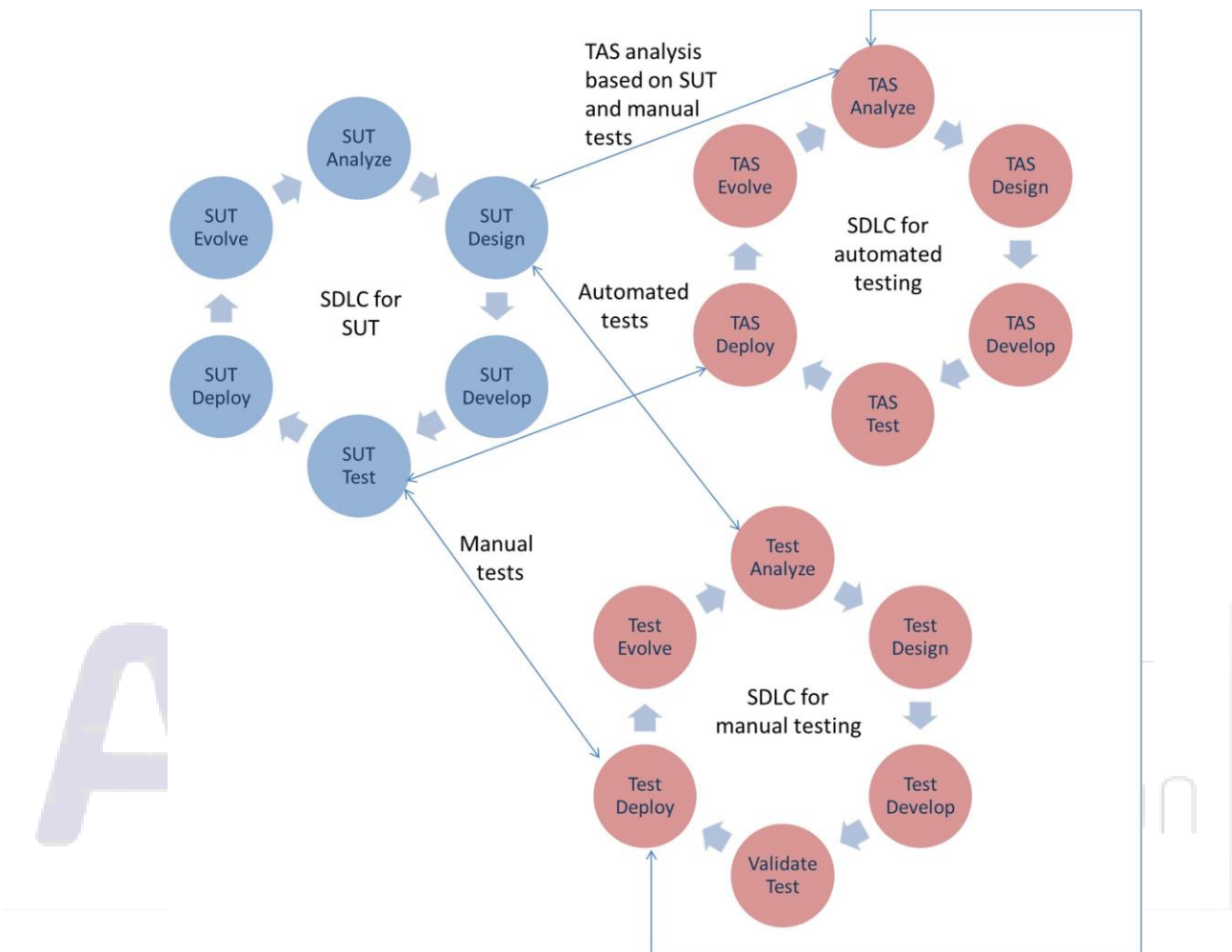


Figura 4: Ejemplo 2 de sincronización de los procesos de desarrollo de la SAP y del SSP

3.3.4 Reutilización de la Compilación en la SAP

La reutilización de una SAP se refiere a la reutilización de los artefactos de la SAP (de cualquier nivel de su arquitectura) en líneas de productos, marco de trabajo de productos, dominios de productos y/o familias de proyectos. Los requisitos para la reutilización resultan de la relevancia de los artefactos de la SAP para las otras variantes de productos, productos y/o proyectos. Los artefactos de la SAP reutilizables pueden incluir:

- (Partes de) modelos de prueba de objetivos de prueba, escenarios de prueba, componentes de prueba o datos de prueba
- (Partes de) casos de prueba, datos de prueba, procedimientos de prueba o bibliotecas de prueba
- El motor de prueba y/o marco de trabajo de gestión de información de la prueba
- Los adaptadores a los componentes y/o interfaces del SSP

Si bien los aspectos de reutilización ya están resueltos cuando se define la AAP, la SAP puede ayudar a aumentar la capacidad de reutilización al:

- Seguir la AAP o revisarla y actualizarla cuando sea necesario
- Documentar los artefactos de la SAP para que se entiendan fácilmente y puedan incorporarse a nuevos contextos

- Asegurar la corrección de cualquier artefacto de la SAP para que el uso en nuevos contextos sea compatible con su alta calidad

Es importante tener en cuenta que si bien el diseño para la reutilización es principalmente un asunto de la AAP, el mantenimiento y las mejoras para la reutilización son una preocupación a lo largo del ciclo de vida de la SAP. Requiere una consideración y un esfuerzo continuos para que la reutilización ocurra, para medir y demostrar el valor agregado de la reutilización y para convencer a otros a reutilizar las SAP existentes.

3.3.5 Soporte para una Variedad de Sistemas Destino

El soporte de la SAP para una variedad de sistemas de destino se refiere a la capacidad de una SAP para probar diferentes configuraciones de un producto de software. Diferentes configuraciones se refiere a cualquiera de las siguientes:

- Número e interconexión de componentes del SSP
- Entornos (software y hardware) en los que se ejecutan los componentes del SSP
- Tecnologías, lenguajes de programación o sistemas operativos utilizados para implementar los componentes del SSP
- Bibliotecas y paquetes que utilizan los componentes del SSP
- Herramientas utilizadas para implementar los componentes del SSP

Si bien los primeros cuatro aspectos afectan al SAP en cualquier nivel de prueba, el último se aplica principalmente a las pruebas de nivel de componente e integración.

La capacidad de una SAP para probar diferentes configuraciones de productos de software se determina cuando se define la AAP. Sin embargo, la SAP debe implementar la capacidad de manejar la variación técnica y debe permitir la administración de las características y componentes de la SAP necesarios para las diferentes configuraciones de un producto de software.

El manejo de la variedad de SAP en relación con la variedad del producto de software se puede tratar de manera diferente:

- La administración de versión/configuración para la SAP y para el SSP se puede usar para proporcionar las respectivas versiones y configuraciones de la SAP y del SSP que se adecuan entre sí.
- La parametrización de la SAP se puede utilizar para ajustar una SAP a una configuración del SSP

Es importante tener en cuenta que si bien el diseño para la reutilización es ante todo un asunto de la AAP, el mantenimiento y las mejoras para la reutilización son una preocupación a lo largo del ciclo de vida de la SAP. Requiere consideración y esfuerzos continuos para revisar, agregar e incluso eliminar opciones y formas de variabilidad.

4 Riesgos de Implementación y Contingencias - 150 minutos

Palabras clave

riesgos, mitigación de riesgos, evaluación de riesgos, riesgo del producto

Objetivos de Aprendizaje para Riesgos de Implementación y Contingencias

4.1 Selección del Enfoque de Automatización de la Prueba y Planificación del Despliegue/Introducción

ALTA-E-4.1.1 (K3) Aplicar directrices que apoyen la prueba piloto eficaz de la herramienta de prueba y las actividades de despliegue

4.2 Evaluación de Riesgos y Estrategias de Mitigación

ALTA-E-4.2.1 (K4) Analizar los riesgos de implementación e identificar los problemas técnicos que podrían conducir al fallo del proyecto de automatización de la prueba, y planificar estrategias de mitigación

4.3 Mantenimiento de la Automatización de Pruebas

Comprender qué factores ayudan y afectan la mantenibilidad de la SAP



4.1 Selección del Enfoque de Automatización de la Prueba y Planificación del Despliegue/Introducción

Hay dos actividades principales involucradas en la implementación e introducción de una SAP: piloto y despliegue. Los pasos que comprenden estas dos actividades variarán según el tipo de SAP y la situación específica.

Para el piloto, al menos deben considerarse los siguientes pasos:

- Identificar un proyecto adecuado
- Planificar el piloto
- Realizar el piloto
- Evaluar el piloto

Para el despliegue, al menos deben considerarse los siguientes pasos:

- Identificar proyecto(s) objetivo inicial(es)
- Implementar la SAP en los proyectos seleccionados
- Monitorizar y evaluar la SAP en los proyectos después de un período predefinido
- Introducirla al resto de la organización/proyectos

4.1.1 Proyecto Piloto

La implementación de herramientas normalmente comienza con un proyecto piloto. El objetivo del proyecto piloto es garantizar que la SAP pueda utilizarse para lograr los beneficios planificados. Los objetivos del proyecto piloto incluyen:

- Aprender más detalles sobre la SAP.
- Vea cómo la SAP concuerda con los procesos, procedimientos y herramientas existentes; identificar cómo podrían necesitar cambiar. (Por lo general, se prefiere modificar la SAP para que se ajuste a los procesos/procedimientos existentes. Si es necesario ajustarlos para ("apoyar la SAP", esto debe ser al menos una mejora de los procesos en sí).
- Diseñar la interfaz de automatización para satisfacer las necesidades de los probadores
- Decidir formas estándar de utilizar, administrar, almacenar y mantener la herramienta y los activos de prueba (p. ej., decidir sobre convenciones de nombres para archivos y pruebas, crear bibliotecas y definir la modularidad de juegos de prueba).
- Identificar métricas y métodos de medición para monitorizar la automatización de pruebas en uso, incluida la usabilidad, la mantenibilidad y la expansibilidad.
- Evaluar si los beneficios se pueden obtener a un costo razonable. Esta será una oportunidad para restablecer las expectativas una vez que se hayan utilizado las SAP.
- Determinar qué habilidades se requieren y cuáles están disponibles y cuáles faltan

Identificar un proyecto adecuado

El proyecto piloto debe seleccionarse cuidadosamente utilizando las siguientes pautas:

- No seleccionar un proyecto crítico. Cuando el despliegue de la SAP cause un retraso, este no debe tener un impacto importante sobre proyectos críticos. El despliegue de la SAP tomará tiempo al principio. El equipo del proyecto debe ser consciente de esto.
- No seleccionar un proyecto trivial. Un proyecto trivial no es un buen candidato ya que el éxito de la implementación no implica el éxito en proyectos no triviales y, por lo tanto, agrega menos información a la necesaria para la implementación.
- Involucrar a las partes interesadas necesarias (incluida la gerencia) en el proceso de selección.

- El SSP del proyecto piloto debe ser una buena referencia para los otros proyectos de la organización, p. ej., el SSP debe contener componentes de la GUI representativos que deben ser automatizados.

Planificar el piloto

El piloto debe tratarse como un proyecto de desarrollo normal: hacer un plan, reservar presupuesto y recursos, informar sobre el avance, definir hitos, etc. Un punto de atención adicional es asegurarse de que las personas que trabajan en el despliegue de la SAP (es decir, un campeón) puede dedicar suficiente esfuerzo al despliegue, incluso cuando otros proyectos exigen los recursos para sus actividades. Es importante tener un compromiso de la administración, especialmente sobre cualquier recurso compartido. Es probable que estas personas no puedan trabajar a tiempo completo en la implementación.

Cuando el proveedor no ha proporcionado la SAP, pero se ha desarrollado internamente, los desarrolladores correspondientes deberán participar en las actividades de implementación.

Realizar el piloto

Realizar el piloto del despliegue y prestar atención a los siguientes puntos:

- ¿La SAP proporciona la funcionalidad esperada (y prometida por el proveedor)? Si no es así, esto debe abordarse lo antes posible. Cuando la SAP se desarrolla internamente, los desarrolladores correspondientes deben ayudar a la implementación proporcionando cualquier funcionalidad faltante.
- ¿La SAP y el proceso existente se apoyan mutuamente? Si no es así, deben estar alineados.

Evaluar el piloto

Utilice a todas las partes interesadas para la evaluación.

4.1.2 Despliegue

Una vez que el piloto ha sido evaluado, la SAP solo se debe implementar en el resto del departamento/organización si el piloto se ha considerado exitoso. La introducción debe realizarse de forma incremental y gestionarse de manera eficiente. Los factores de éxito para la implementación incluyen:

- Una introducción incremental: Realice la introducción al resto de la organización en pasos, en incrementos. De esta manera, el apoyo a los nuevos usuarios viene en "olas" en lugar de todos a la vez. Esto permite que el uso de la SAP aumente en pasos. Los posibles cuellos de botella se pueden identificar y resolver antes de que se conviertan en problemas reales. Las licencias se pueden agregar cuando sea necesario.
- Adaptar y mejorar los procesos para adecuarlos al uso de la SAP: Cuando diferentes usuarios usan la SAP, diferentes procesos entran en contacto con la SAP y deben ajustarse a la SAP, o la SAP puede necesitar adaptaciones (pequeñas) a los procesos.
- Proporcionar capacitación y entrenamiento/tutoría para nuevos usuarios: Los nuevos usuarios necesitan capacitación y entrenamiento en el uso de la nueva SAP. Asegúrese de que esta esté establecida. Se debe proporcionar capacitación/talleres a los usuarios antes de que realmente usen la SAP.
- Definir pautas de uso: Es posible escribir pautas, listas de verificación y preguntas frecuentes para el uso de la SAP. Esto puede evitar extensas preguntas de apoyo.
- Implementación de una forma de recopilar información sobre el uso real: Debe haber una forma automatizada de recopilar información sobre el uso real de la SAP. Lo ideal es no solo el uso en sí, sino también las partes de la SAP (ciertas funcionalidades) que se están utilizando. De esta manera, el uso de la SAP puede ser monitorizado fácilmente.
- Monitorización del uso de la SAP, beneficios y costos: La monitorización del uso de la SAP durante un cierto período de tiempo indica si la SAP se usa efectivamente. Esta información también se puede usar para volver a calcular el caso de negocio (p. ej., cuánto tiempo se ha ahorrado, cuántos problemas se han evitado).
- Brindar apoyo al equipo de prueba para una SAP dada.

- Recopilar las lecciones aprendidas de todos los equipos: Realizar reuniones de evaluación/retrospectivas con los diferentes equipos que utilizan la SAP. De esta manera, se pueden identificar las lecciones aprendidas. Los equipos sentirán que su aporte es necesario y querían mejorar el uso de la SAP.
- Identificar e implementar mejoras: A partir de la retroalimentación del equipo y la monitorización de la SAP, identifique e implemente los pasos para la mejora. También comuníquese esto claramente a las partes interesadas.

4.1.3 Despliegue de la SAP dentro del Ciclo de Vida del Software

El despliegue de una SAP depende en gran medida de la fase de desarrollo del proyecto de software que será probado por la SAP.

Por lo general, se implementa una nueva SAP o una nueva versión al inicio del proyecto o al alcanzar un hito, como la congelación del código o el final de un sprint. Esto se debe a que las actividades de despliegue, con todas las pruebas y modificaciones involucradas, requieren tiempo y esfuerzo. Además, esta es una buena manera de mitigar el riesgo de que la SAP no funcione y cause interrupciones en el proceso de automatización de la prueba. Sin embargo, si hay problemas críticos que deben solucionarse para la SAP o si se debe reemplazar un componente del entorno en el que se ejecuta, entonces la implementación se realizará independientemente de la fase de desarrollo del SSP.

4.2 Evaluación de Riesgos y Estrategias de Mitigación

Los problemas técnicos pueden llevar a riesgos de producto o proyecto. Los problemas técnicos típicos incluyen:

- Demasiada abstracción puede llevar a dificultades para entender lo que realmente sucede (p. ej., con palabras clave)
- Guiado por datos: Las tablas de datos pueden ser demasiado grandes/complejas/engorrosas
- Dependencia de la SAP al uso de ciertas bibliotecas del sistema operativo u otros componentes que pueden no estar disponibles en todos los entornos de destino del SSP

Los riesgos típicos de un proyecto de despliegue incluyen:

- Problemas de personal: Conseguir que las personas adecuadas mantengan el código base puede ser difícil
- Las nuevas entregas del SSP pueden hacer que la SAP funcione incorrectamente
- Retrasos en la introducción de la automatización
- Retrasos en la actualización de la SAP en función de los cambios realizados en el SSP
- La SAP no puede capturar los objetos (no estándar) que se pretende rastrear

Los puntos potenciales de fallo del proyecto de SAP incluyen:

- Migración a un entorno diferente
- Despliegue en el entorno de destino
- Nueva entrega de desarrollo

Hay una serie de estrategias de mitigación de riesgos que pueden emplearse para hacer frente a estas áreas de riesgo. Estas se discuten a continuación.

La SAP tiene un ciclo de vida de software propio, ya sea desarrollado internamente o como una solución adquirida. Una cosa que se debe recordar es que la SAP, como cualquier otro software, debe estar bajo el control de la versión y sus prestaciones documentadas. De lo contrario, se vuelve muy difícil desplegar diferentes partes de ella y hacer que trabajen juntas, o que funcionen en ciertos entornos.

Además, tiene que haber un procedimiento de despliegue documentado, claro y fácil de seguir. Este procedimiento es dependiente de la versión; por lo tanto, debe incluirse también en el control de versiones.

Hay dos casos distintos al desplazar una SAP:

1. Despliegue inicial
2. Despliegue de mantenimiento: La SAP ya existe y debe mantenerse.

Antes de comenzar con la primera implementación de una SAP, es importante asegurarse de que se puede ejecutar en su propio entorno, está aislada de los cambios aleatorios y los casos de prueba se pueden actualizar y administrar. Tanto la SAP como su infraestructura deben mantenerse.

En el caso del primer despliegue, se necesitan los siguientes pasos básicos:

- Definir la infraestructura en la que se ejecutará la SAP
- Crear la infraestructura para la SAP
- Crear un procedimiento para mantener la SAP y su infraestructura
- Crear un procedimiento para mantener el conjunto de pruebas que ejecutará la SAP

Los riesgos relacionados con el despliegue por primera vez incluyen:

- El tiempo total de ejecución del conjunto de pruebas puede ser más largo que el tiempo de ejecución planificado para el ciclo de prueba. En este caso, es importante asegurarse de que el conjunto de pruebas tenga el tiempo suficiente para ejecutarse por completo antes de que comience el siguiente ciclo de prueba programado.
- Existen problemas de instalación y configuración con el entorno de prueba (p. ej., configuración de la base de datos y carga inicial, inicio/detención de servicios). En general, la SAP necesita una forma efectiva de configurar las condiciones previas necesarias para los casos de prueba automatizados dentro del entorno de prueba.

Para los despliegues de mantenimiento, hay consideraciones adicionales. La SAP en sí misma necesita evolucionar, y las actualizaciones para ella deben desplegarse en producción. Antes de desplegar una versión actualizada de la SAP en producción, debe probarse como cualquier otro software. Por lo tanto, es necesario verificar la nueva funcionalidad para verificar que el conjunto de pruebas se pueda ejecutar en la SAP actualizada, que se puedan enviar informes y que no haya problemas de rendimiento u otras regresiones funcionales. En algunos casos, es posible que deba cambiarse todo el conjunto de pruebas para que se ajuste a la nueva versión de la SAP.

Cuando se produce el despliegue de mantenimiento, se necesitan los siguientes pasos:

- Haga una evaluación de los cambios en la nueva versión de la SAP en comparación con la anterior
- Pruebe la SAP para nuevas funcionalidades y regresiones
- Compruebe si el conjunto de pruebas debe adaptarse a la nueva versión de la SAP

Una actualización también incurre en los siguientes riesgos y acciones de mitigación correspondientes:

- El juego de pruebas debe cambiar para ejecutarse en la SAP actualizado: Haga los cambios necesarios en el conjunto de pruebas y pruébelos antes de implementarlos en la SAP.
- Los stubs, los controladores y las interfaces que se usan en las pruebas deben modificarse para que se ajusten a la SAP actualizada: Haga los cambios necesarios en el arnés de prueba y pruébelo antes de implementarlos en la SAP.
- La infraestructura debe cambiar para adaptarse a la SAP actualizada: Haga una evaluación de los componentes de la infraestructura que deben modificarse, realice los cambios y pruébelos con la SAP actualizada.
- La SAP actualizada tiene defectos adicionales o problemas de rendimiento: Haga un análisis de riesgos vs. beneficios. Si los problemas detectados 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 problemas son insignificantes en comparación con los beneficios, la SAP aún puede actualizarse. Asegúrese de crear una nota de entrega con problemas conocidos para notificarlo a los ingenieros de automatización de pruebas y otras partes interesadas y tratar de obtener una estimación de cuándo se solucionarán los problemas.

4.3 Mantenimiento de la Automatización de Pruebas

Implementar soluciones de automatización de pruebas no es trivial. Deben ser modulares, escalables, comprensibles, confiables y comprobables. Para agregar aún más complejidad, las soluciones de automatización de pruebas, como cualquier otro sistema de software, deben evolucionar. Ya sea debido a cambios internos o cambios en el entorno en el que operan, el mantenimiento es un aspecto importante de la arquitectura de una SAP. Mantener la SAP al adaptarla a los nuevos tipos de sistemas que se van a probar, acomodar el soporte para nuevos entornos de software o cumplir con las nuevas leyes y regulaciones, ayuda a garantizar un funcionamiento confiable y seguro de la SAP. También optimiza la vida útil y el rendimiento de la SAP.

4.3.1 Tipos de Mantenimiento

El mantenimiento se realiza en una SAP operativo existente y se desencadena por modificaciones, migración o retiro del sistema. Este proceso se puede estructurar en las siguientes categorías:

- **Mantenimiento preventivo:** Se realizan cambios para hacer que la SAP sea compatible con más tipos de prueba, realice pruebas en múltiples interfaces, pruebe varias versiones del SSP o admita la automatización de pruebas para un nuevo SSP.
- **Mantenimiento correctivo:** Se realizan cambios para corregir fallos de la SAP. La mejor manera de mantener una SAP en operación, reduciendo así el riesgo de usarla, es a través de la ejecución de pruebas de mantenimiento regulares.
- **Mantenimiento perfectivo:** La SAP está optimizada y los problemas no funcionales están resueltos. Pueden abordar el rendimiento de la SAP, su usabilidad, robustez o fiabilidad.
- **Mantenimiento adaptativo:** A medida que se entregan nuevos sistemas de software en el mercado (sistemas operativos, administradores de bases de datos, navegadores web, etc.), puede ser necesario que la SAP los admita. Además, puede darse el caso de que la SAP deba cumplir con las nuevas leyes, regulaciones o requisitos específicos de la industria. En este caso, se realizan cambios en la SAP para adaptarlo según el caso. Nota: por lo general, el cumplimiento de las leyes y regulaciones crea un mantenimiento obligatorio con reglas, requisitos específicos y, en ocasiones, requisitos de auditoría. Además, a medida que se actualizan las herramientas de integración y se crean nuevas versiones, los puntos finales de integración de herramientas deben mantenerse y ser funcionales.

4.3.2 Alcance y Enfoque

El mantenimiento es un proceso que puede afectar a todas las capas y componentes de una SAP. El alcance del mismo depende de:

- El tamaño y complejidad de la SAP
- El tamaño del cambio
- El riesgo del cambio

Dado que el mantenimiento se refiere a la SAP en operación, es necesario un análisis del impacto para determinar cómo el sistema puede verse afectado por los cambios. Dependiendo del impacto, los cambios deben introducirse de forma incremental y las pruebas deben realizarse después de cada paso para garantizar el funcionamiento continuo de la SAP. Nota: mantener la SAP puede ser difícil si sus especificaciones y documentación están desactualizadas.

Debido a que la eficiencia del tiempo es el principal factor que contribuye al éxito de la automatización de

pruebas, es fundamental contar con buenas prácticas para mantener la SAP, que incluyen:

- Los procedimientos de implementación y el uso de la SAP deben ser claros y estar documentados
- Las dependencias de terceros deben estar documentadas, junto con inconvenientes y problemas conocidos
- La SAP debe ser modular, de modo que sus partes se puedan reemplazar fácilmente
- La SAP debe ejecutarse en un entorno que sea reemplazable o con componentes reemplazables
- La SAP debe separar los guiones del propio TAF
- La SAP debe ejecutarse aislada del entorno de desarrollo, de modo que los cambios en la SAP no afecten negativamente al entorno de prueba
- Las SAP junto con el entorno, el conjunto de pruebas y los artefactos de prueba deben estar bajo la administración de la configuración

También hay consideraciones para el mantenimiento de los componentes de terceros y otras bibliotecas de la siguiente manera:

- Muy a menudo es el caso de que la SAP utilizará componentes de terceros para ejecutar las pruebas. También puede darse el caso de que la SAP dependa de bibliotecas de terceros (p. ej., las bibliotecas de automatización de la interfaz de usuario). Todos los componentes de terceros de la SAP deben estar documentados y bajo administración de configuración.
- Es necesario tener un plan en caso de que estos componentes externos deban modificarse o repararse. La persona responsable del mantenimiento de la SAP necesita saber con quién comunicarse o dónde enviar un problema.
- Debe haber documentación con respecto a la licencia bajo la cual se utilizan los componentes de terceros, de modo que haya información sobre si se pueden modificar, en qué grado y por quién.
- Para cada uno de los componentes de terceros, es necesario obtener información sobre actualizaciones y nuevas versiones. Mantener actualizados los componentes y las bibliotecas de terceros es una acción preventiva que amortiza la inversión a largo plazo.

Las consideraciones para nombrar normas y otras convenciones incluyen:

- La idea de nombrar normas y otras convenciones tiene una razón simple: el conjunto de pruebas y la propia SAP deben ser fáciles de leer, comprender, cambiar y mantener. Esto ahorra tiempo en el proceso de mantenimiento y también minimiza el riesgo de introducir regresiones o correcciones incorrectas que podrían evitarse fácilmente.
- Es más fácil introducir a nuevas personas en el proyecto de automatización de prueba cuando se utilizan las convenciones de nomenclatura estándar.
- Las normas de denominación pueden referirse a variables y archivos, escenarios de prueba, palabras clave y parámetros de palabras clave. Otras convenciones se refieren a requisitos previos y acciones posteriores para la ejecución de la prueba, el contenido de los datos de la prueba, el entorno de la prueba, el estado de la ejecución de la prueba y los registros e informes de ejecución.
- Todas las normas y convenciones deben acordarse y documentarse al iniciar un proyecto de automatización de prueba.

Las consideraciones de documentación incluyen:

- La necesidad de una documentación buena y actual tanto para los escenarios de prueba como para la SAP es bastante clara, pero hay dos problemas relacionados con esto: alguien tiene que escribirlo y alguien debe mantenerlo.
- Si bien el código de la herramienta de prueba puede ser autodocumentado o documentado semiautomáticamente, cualquier persona debe documentar todo el diseño, los componentes, las integraciones con terceros, las dependencias y los procedimientos de implementación.
- Es una buena práctica introducir la escritura de la documentación como parte del proceso de desarrollo. Una tarea no debe considerarse como realizada a menos que esté documentada o la documentación esté actualizada.

Las consideraciones del material para capacitación incluyen:

- Si la documentación para la SAP está bien escrita, se puede utilizar como base para el material de capacitación de la SAP.
- El material de capacitación es una combinación de especificaciones funcionales de la SAP, diseño y arquitectura de la SAP, implementación y mantenimiento del SAP, uso del SAP (manual del usuario), ejemplos prácticos y ejercicios, y sugerencias y trucos.
- El mantenimiento del material de capacitación consiste en escribirlo inicialmente y luego revisarlo periódicamente. Se realiza en la práctica por los miembros del equipo designados como capacitadores en la SAP y es muy probable que ocurra al final de una iteración del ciclo de vida del SSP (al final de los sprints, por ejemplo).



5 Gestión de Información de Automatización de la Prueba y

Palabras clave

código de automatización densidad de defectos, cobertura, matriz de trazabilidad, esfuerzo de prueba manual equivalente, métricas, registro de pruebas, gestión de información de la prueba

Objetivos de Aprendizaje para la Gestión de Información de la Automatización de la Prueba y Métricas

5.1 Selección de Métricas de SAP

ALTA-E-5.1.1 (K2) Clasificar las métricas que se pueden usar para monitorear la estrategia de automatización de la prueba y la efectividad

5.2 Implementación de la Medición

ALTA-E-5.2.1 (K3) Implementar métodos de recolección de métricas para apoyar los requisitos técnicos y de gestión Explicar cómo se puede implementar la medición de la automatización de la prueba.

5.3 Registro de la SAP y del SSP

ALTA-E-5.3.1 (K4) Analizar el registro de prueba de los datos de la SAP y del SSP

5.4 Gestión de Información de Automatización de la Prueba

ALTA-E-5.4.1 (K2) Explicar cómo se construye y publica una gestión de información de la ejecución de la prueba

5.1 Selección de Métricas de SAP

Esta sección se enfoca en las métricas que se pueden usar para monitorizar la estrategia de automatización de pruebas y la efectividad y eficiencia de la SAP. Estas son independientes de las métricas relacionadas con el SSP que se utilizan para monitorizar el SSP y las pruebas (funcionales y no funcionales) del SSP. Estas son seleccionadas por el Jefe de Prueba general del proyecto. Las métricas de automatización de pruebas le permiten al JAP y al IAP realizar un seguimiento del avance hacia los objetivos de la automatización de pruebas y monitorear el impacto de los cambios realizados en la solución de automatización de pruebas.

Las métricas de la SAP se pueden dividir en dos grupos: externas e internas. Las métricas externas son las que se utilizan para medir el impacto de la SAP en otras actividades (en particular, las actividades de prueba). Las métricas internas son aquellas que se utilizan para medir la efectividad y la eficiencia de la SAP en el cumplimiento de sus objetivos.

Las métricas de la SAP medidas típicamente incluyen lo siguiente:

- Métricas externas de la SAP
 - Beneficios de la automatización
 - Esfuerzo para construir pruebas automatizadas
 - Esfuerzo para analizar incidencias de las pruebas automatizadas
 - Esfuerzo para manter pruebas automatizadas
 - Índice de fallos ante defectos
 - Tiempo para ejecutar pruebas automatizadas
 - Número de casos de prueba automatizadas
 - Número de resultados de paso y fallo
 - Número de resultados falso-fallo y falso-paso
 - Cobertura de código
- Métricas internas de la SAP
 - Métricas de escritura de guiones de herramientas
 - Densidad de defectos del código de automatización
 - Velocidad y eficiencia de los

componentes de la SAP

Beneficios de la automatización

Es particularmente importante medir e informar los beneficios de una SAP. Esto se debe a que los costos (en términos de la cantidad de personas involucradas durante un período de tiempo determinado) son fáciles de ver. Las personas que trabajan fuera de las pruebas podrán formarse una impresión del costo general, pero es posible que no vean los beneficios alcanzados.

Cualquier medida de beneficio dependerá del objetivo de la SAP. Por lo general, esto puede ser un ahorro de tiempo o esfuerzo, un aumento en la cantidad de pruebas realizadas (amplitud o profundidad de la cobertura, o la frecuencia de ejecución), o alguna otra ventaja, como mayor repetibilidad, mayor uso de recursos o menos errores manuales. Las posibles mediciones incluyen:

- Número de horas ahorradas de esfuerzo de prueba manual
- Reducción en el tiempo para realizar pruebas de regresión
- Número de ciclos adicionales de ejecución de pruebas logrados
- Número o porcentaje de pruebas adicionales ejecutadas
- Porcentaje de casos de prueba automatizados relacionados con el conjunto completo de casos de prueba (aunque los automatizados no se pueden comparar fácilmente con casos de prueba manuales)

- Aumento de la cobertura (requisitos, funcionalidad, estructural)
- Número de defectos encontrados anteriormente debido a la SAP (cuando se conoce el beneficio promedio de los defectos encontrados anteriormente, esto se puede "calcular" a una suma de costos evitados)
- Número de defectos encontrados debido a la SAP que no se habría encontrado mediante pruebas manuales (p. ej., defectos de confiabilidad)

Tenga en cuenta que la automatización de pruebas generalmente ahorra esfuerzo de prueba manual. Este esfuerzo se puede dedicar a otros tipos de pruebas (manuales) (p. ej., pruebas exploratorias). Los defectos encontrados por estas pruebas adicionales también pueden verse como beneficios indirectos de la SAP, ya que la automatización de pruebas permitió que se ejecutaran estas pruebas manuales. Sin la SAP, estas pruebas no se habrían ejecutado y, posteriormente, no se habrían encontrado otros defectos.

Esfuerzo para compilar pruebas automatizadas

El esfuerzo por automatizar las pruebas es uno de los costos clave asociados con la automatización de las pruebas. A menudo, este es más que el costo de ejecutar la misma prueba manualmente y, por lo tanto, puede ser un daño para la extensión del uso de la automatización de pruebas. Si bien el costo de implementar una prueba automatizada específica dependerá en gran medida de la prueba en sí, otros factores, como el enfoque de scripting utilizado, la familiaridad con la herramienta de prueba, el entorno y el nivel de habilidad del ingeniero de automatización de pruebas también tendrán un impacto.

Debido a que pruebas más grandes o más complejas generalmente toman más tiempo para automatizarse que las pruebas cortas o simples, el cálculo del costo de compilación para la automatización de pruebas puede basarse en un tiempo de compilación promedio. Esto se puede refinar aún más considerando el costo promedio de un conjunto específico de pruebas, como las que se dirigen a la misma función o las de un nivel de prueba determinado. Otro enfoque es expresar el costo de compilación como un factor del esfuerzo requerido para ejecutar la prueba manualmente (esfuerzo de prueba manual equivalente, EPME). Por ejemplo, puede ser que se requiera dos veces el esfuerzo de prueba manual para automatizar un caso de prueba, o dos veces el EPME.

Esfuerzo para analizar fallos del SSP

El análisis de los fallos en el SSP detectados a través de la ejecución automatizada de la prueba puede ser significativamente más complejo que para una prueba ejecutada manualmente porque los probadores que ejecutan la prueba a menudo conocen los eventos que conducen al fallo de una prueba manual. Esto puede mitigarse como se describe en el nivel de diseño en el Capítulo 3.1.4 y en el nivel de gestión de información en los Capítulos 5.3 y 5.4. Esta medida puede expresarse como un promedio por caso de prueba fallida o puede expresarse como un factor de EPME. Este último es particularmente adecuado cuando las pruebas automatizadas varían significativamente en complejidad y duración de ejecución.

El registro disponible del SSP y la SAP juegan un papel crucial en el análisis de fallos. El registro debe proporcionar suficiente información para realizar este análisis de manera eficiente. Las prestaciones importantes de registro incluyen:

- El registro del SSP y el registro de la SAP deben estar sincronizados
- La SAP debe registrar el comportamiento esperado y el real
- La SAP debe registrar las acciones a realizar

El SSP, por otro lado, debe registrar todas las acciones que se realizan (independientemente de si la acción es el resultado de pruebas manuales o automáticas). Se debe registrar cualquier error interno y deben estar disponibles los volcados de memoria y los seguimientos de la pila.

Esfuerzo para mantener pruebas automatizadas

El esfuerzo de mantenimiento requerido para mantener las pruebas automatizadas sincronizadas con el SSP puede ser muy importante y, en última instancia, puede superar los beneficios logrados por la SAP. Esto ha sido la causa del fracaso de muchos esfuerzos de automatización. Por lo tanto, es importante

hacer un seguimiento del esfuerzo de mantenimiento cuando es necesario tomar medidas para reducir el esfuerzo de mantenimiento o al menos evitar que crezca sin control.

Las mediciones del esfuerzo de mantenimiento pueden expresarse como un total para todas las pruebas automatizadas que requieren mantenimiento para cada nueva entrega del SSP. También pueden expresarse como un promedio por prueba automatizada actualizada o como un factor de EPME.

Una métrica relacionada es el número o porcentaje de pruebas que requieren trabajo de mantenimiento.

Cuando se conoce (o se puede derivar) el esfuerzo de mantenimiento para las pruebas automatizadas, esta información puede desempeñar un papel crucial en la decisión de implementar o no cierta funcionalidad o de corregir un defecto determinado. El esfuerzo requerido para mantener el caso de prueba debido al software modificado debe considerarse con cambio del SSP.

Índice de fallos ante defectos

Un problema común con las pruebas automatizadas es que muchas de ellas pueden fallar por el mismo motivo: un solo defecto en el software. Si bien el propósito de las pruebas es resaltar los defectos en el software, tener más de una prueba que falla por el mismo defecto es antieconómico. Este es particularmente el caso de las pruebas automatizadas, ya que el esfuerzo requerido para analizar cada prueba fallida puede ser significativo. Medir la cantidad de pruebas automatizadas que fallan para un defecto dado puede ayudar a indicar dónde puede haber un problema. La solución radica en el diseño de las pruebas automatizadas y su selección para su ejecución.

Tiempo para ejecutar pruebas automatizadas

Una de las métricas más fáciles de determinar es el tiempo que lleva ejecutar las pruebas automatizadas. Al comienzo de la SAP, esto podría no ser importante, pero a medida que aumenta el número de casos de prueba automatizados, esta métrica puede ser muy importante.

Número de casos de prueba automatizadas

Esta métrica se puede utilizar para mostrar la progresión realizada por el proyecto de automatización de la prueba. Pero hay que tener en cuenta que solo el número de casos de prueba automatizados no revela mucha información; por ejemplo, no indica que la cobertura de prueba haya aumentado.

Número de resultados de paso y fallo

Esta es una métrica común y realiza un seguimiento de cuántas pruebas automatizadas pasaron y de cuántas no lograron el resultado esperado. Los fallos deben analizarse para determinar si el fallo se debió a un defecto en el SSP o se debió a problemas externos, como un problema con el entorno o con la propia SAP.

Número de resultados falso-fallo y falso-paso

Como se vio en varias métricas anteriores, puede llevar bastante tiempo analizar los fallos de las pruebas. Esto es aún más frustrante cuando resulta ser una falsa alarma. Esto sucede cuando el problema está en la SAP o en el caso de prueba, pero no en el SSP. Es importante que el número de falsas alarmas (y el esfuerzo potencialmente desperdiciado) se mantengan bajos. Los falsos fallos pueden reducir la confianza en la SAP. Por el contrario, los resultados de falso paso pueden ser más peligrosos. Cuando se produce un falso paso, hubo un fallo en el SSP, pero la automatización de prueba no lo identificó, por lo que se informó un resultado de paso. En este caso, un defecto potencial puede escapar a la detección. Esto puede ocurrir porque la verificación del resultado no se realizó correctamente, se usó un oráculo de prueba no válido o el caso de prueba esperaba el resultado incorrecto.

Tenga en cuenta que las falsas alarmas pueden ser causadas por defectos en el código de prueba (consulte la métrica "Densidad de defectos del código de automatización"), pero también pueden ser causadas por un SSP inestable que se comporta de manera impredecible (p. ej., tiempo agotado). Los anzuelos de prueba también pueden causar falsas alarmas debido al nivel de intrusión que están causando.

Cobertura del código

Conocer la cobertura del código del SSP proporcionada por los diferentes casos de prueba puede revelar información útil. Esto también puede medirse a un alto nivel, p. ej., la cobertura de código del conjunto de pruebas de regresión. No existe un porcentaje absoluto que indique una cobertura adecuada, y la cobertura de código del 100% es inalcanzable en cualquier otra cosa que no sea la más simple de las aplicaciones de software. Sin embargo, generalmente se acepta que una mayor cobertura es mejor, ya que reduce el riesgo general de la implementación del software. Esta métrica puede indicar actividad en el SSP también. Por ejemplo, si la cobertura del código cae, lo más probable es que esto signifique que la funcionalidad se haya agregado al SSP, pero no se haya agregado el caso de prueba correspondiente al conjunto de pruebas automatizado.

Métricas de escritura de guiones de herramientas

Hay muchas métricas que se pueden usar para monitorizar el desarrollo de la escritura de guiones de automatización. La mayoría de ellas son similares a la métrica del código fuente para el SSP. Las líneas de código (LDC) y la complejidad ciclomática pueden ser usadas para destacar escrituras demasiado grandes o complejas (es necesario sugerir un posible rediseño).

La proporción de comentarios a sentencias ejecutables se puede utilizar para dar una posible indicación de la extensión de la documentación del guión y la anotación. El número de no conformidades con las normas de escritura de guión puede dar una indicación de hasta qué punto se están siguiendo esas normas.

Densidad de defectos del código de automatización

El código de automatización no es diferente del código del SSP, ya que es un software y contendrá defectos. El código de automatización no debe considerarse menos importante que el código del SSP. Se deben aplicar buenas prácticas y estándares de codificación, y el resultado de estos debe ser controlado por métricas como la densidad de defectos de código. Estos serán más fáciles de recopilar con el soporte de un sistema de administración de configuración.

Velocidad y eficiencia de los componentes de la SAP

Las diferencias en el tiempo que toma realizar los mismos pasos de prueba en el mismo entorno pueden indicar un problema en el SSP. Si el SSP no está realizando la misma funcionalidad en el mismo tiempo transcurrido, se necesita una investigación. Esto puede indicar una variabilidad en el sistema que no es aceptable y que podría empeorar con el aumento de carga. La SAP debe estar funcionando lo suficientemente bien como para que no obstaculice el rendimiento del SSP.

Métricas de tendencia

Con muchas de estas métricas son las tendencias (es decir, la forma en que las medidas cambian con el tiempo) las que pueden ser más valiosas para el informe que el valor de una medida en un momento específico. Por ejemplo, saber que el costo promedio de mantenimiento por prueba automatizada que requiere mantenimiento es mayor de lo que era para las dos entregas anteriores del SSP puede provocar una acción para determinar la causa del aumento y emprender pasos para revertir la tendencia.

El costo de la medición debe ser lo más bajo posible y esto puede lograrse a menudo automatizando la recopilación y la gestión de información.

5.2 Implementación de la Medición

Dado que una estrategia de automatización de pruebas tiene un software de pruebas automatizado en su núcleo, se puede mejorar el software de pruebas automatizado para registrar información sobre su uso. Cuando la abstracción se combina con el software de prueba estructurado, todas las guiones automatizadas de nivel superior pueden utilizar las mejoras realizadas en el software de prueba subyacente. Por ejemplo, la mejora del software de prueba subyacente para registrar el tiempo de inicio y finalización de una prueba puede aplicarse a todas las pruebas.

Prestaciones de automatización que soportan la medición y generación de reportes

Los lenguajes de escritura de guiones de muchas herramientas de prueba admiten la medición y la generación de informes a través de prestaciones que pueden usarse para grabar y registrar información antes, durante y después de la ejecución de pruebas individuales, juegos de prueba y un juego de pruebas completo.

La gestión de información sobre cada una de una serie de ejecuciones de prueba debe tener implementada una función de análisis para tener en cuenta los resultados de las ejecuciones de prueba anteriores para que pueda resaltar tendencias (como los cambios en la tasa de éxito de la prueba).

La automatización de las pruebas generalmente requiere la automatización tanto de la ejecución de la prueba como de la verificación de la prueba, esto último se logra al comparar elementos específicos del resultado de la prueba con un resultado esperado predefinido. Esta comparación generalmente se realiza mejor mediante una herramienta de prueba. Se debe considerar el nivel de información que se informa como resultado de esta comparación. Es importante que el estado de la prueba se determine correctamente (p. ej., paso, fallo). En el caso de un estado fallido, se requerirá más información sobre la causa del error (p. ej., capturas de pantalla).

Distinguir entre las diferencias esperadas en el resultado real y el esperado de una prueba no siempre es trivial, aunque el apoyo de la herramienta puede ser de gran ayuda para definir comparaciones que ignoran las diferencias esperadas (como fechas y horas) al tiempo que resalta cualquier diferencia inesperada.

Integración con otras herramientas de terceros (hojas de cálculo, XML, documentos, bases de datos, herramientas de gestión de información, etc.)

Cuando la información de la ejecución de casos de prueba automatizada se utiliza en otras herramientas (para el seguimiento y la generación de informes, p. ej., la actualización de la matriz de trazabilidad), es posible proporcionar la información en un formato adecuado para estas herramientas de terceros. Esto se logra a menudo mediante la funcionalidad de la herramienta de prueba existente (exportar formatos para informes) o creando informes personalizados que se generan en un formato compatible con otros programas (".xls" para Excel, ".doc" para Word, ".html" para Web, etc.).

Visualización de resultados (paneles de control, cuadros, gráficos, etc.)

Los resultados de las pruebas deben hacerse visibles en tablas. Considere usar colores para indicar problemas en la ejecución de la prueba, como los semáforos para indicar el avance de la ejecución/automatización de la prueba, de modo que las decisiones se puedan tomar en base a la información reportada. La gerencia está particularmente interesada en los resúmenes visuales para ver el resultado de la prueba de un vistazo; en caso de que se necesite más información, todavía pueden sumergirse en los detalles.

5.3 Registro de la SAP y del SSP

El registro es muy importante en la SAP, incluido el registro tanto para la automatización de pruebas como para el SSP. Los registros de prueba son una fuente que se utiliza con frecuencia para analizar problemas potenciales. En la siguiente sección hay ejemplos de registro de prueba, categorizados por SAP o SSP.

El registro de la SAP (ya sea que el TAF o el propio caso de prueba registren la información no es tan importante y depende del contexto) debe incluir lo siguiente:

- El caso de prueba que se está ejecutando actualmente, incluida la hora de inicio y finalización
- El estado de la ejecución del caso de prueba porque, si bien los fallos se pueden identificar fácilmente en los archivos de registro, el marco de trabajo también debe tener esta información y se debe informar a través de un panel de control. El estado de ejecución del caso de prueba puede ser paso, fallo o error de la SAP. El resultado del error de la SAP se usa para situaciones donde el problema no está en el SSP.
- Detalles del registro de prueba a un nivel alto (registro de pasos significativos) que incluye información sobre el tiempo.
- Información dinámica sobre el SSP (p. ej., fugas de memoria) que el caso de prueba pudo identificar con la ayuda de herramientas de terceros. Los resultados reales y los fallos de estas mediciones dinámicas deben registrarse con el caso de prueba que se estaba ejecutando cuando se detectó el incidente.
- En el caso de pruebas de confiabilidad/pruebas de estrés (donde se realizan numerosos ciclos) se debe registrar un contador, de modo que se pueda determinar fácilmente cuántas veces se han ejecutado los casos de prueba.
- Cuando los casos de prueba tienen partes aleatorias (p. ej., parámetros aleatorios o pasos aleatorios en la prueba de la máquina de estados), el número/selección aleatoria debe registrarse.
- Todas las acciones que realiza un caso de prueba deben registrarse de tal manera que el archivo de registro (o partes de él) se pueda reproducir para volver a ejecutar la prueba con exactamente los mismos pasos y el mismo tiempo. Esto es útil para verificar la reproducibilidad de un fallo identificado y para capturar información adicional. La información sobre la acción del caso de prueba también se puede registrar en el propio SSP para su uso cuando se reproducen los problemas identificados por el cliente (el cliente ejecuta el escenario, la información de registro se captura y el equipo de desarrollo puede volver a reproducirla para solucionar el problema).
- Las capturas de pantalla y otras capturas visuales se pueden guardar durante la ejecución de la prueba para su uso posterior durante el análisis de fallos.
- Cuando un caso de prueba encuentra un fallo, la SAP debe asegurarse de que toda la información necesaria para analizar el problema esté disponible/almacenada, así como cualquier información relacionada con la continuación de las pruebas, si corresponde. La SAP debe guardar en un lugar seguro cualquier volcado de caída y seguimiento de pila asociados. Además, todos los archivos de registro que podrían sobrescribirse (las memorias intermedias cíclicas se utilizan a menudo para los archivos de registro en el SSP) deben copiarse en esta ubicación donde estarán disponibles para su posterior análisis.
- El uso del color puede ayudar a distinguir diferentes tipos de información registrada (p. ej., errores en rojo, información de avance en verde).

Registro del SSP:

- Cuando el SSP identifica un problema, se debe registrar toda la información necesaria para analizar el problema, incluidas las marcas de tiempo y fecha, la ubicación de origen del problema, los mensajes de error, etc.
- El SSP puede registrar toda la interacción del usuario (directamente a través de la interfaz de usuario disponible, pero también a través de interfaces de red, etc.). De esta manera, los problemas identificados por los clientes se pueden analizar correctamente y el desarrollo puede

intentar reproducir el problema.

- Al iniciar el sistema, la información de configuración debe registrarse en un archivo, que consiste en las diferentes versiones de software/firmware, la configuración del SSP, la configuración del sistema operativo, etc.

Toda la información de registro diferente debe ser fácil de buscar. Un problema identificado por la SAP en el archivo de registro debe identificarse fácilmente en el archivo de registro del SSP, y viceversa (con o sin herramientas adicionales). La sincronización de varios registros con una marca de tiempo facilita la correlación de lo que ocurrió cuando se informó un error.

5.4 Gestión de Información de la Automatización de la Prueba

Los registros de prueba proporcionan información detallada sobre los pasos de ejecución, las acciones y las respuestas de un caso de prueba y/o conjunto de pruebas. Sin embargo, los registros por sí solos no pueden proporcionar una buena visión general del resultado general de la ejecución. Para esto, es necesario tener implementada la funcionalidad de gestión de información. Después de cada ejecución del juego de pruebas, se debe crear y publicar un informe conciso. Se podría usar un componente generador de informes reutilizable para ello.

Contenido de los informes

El informe de ejecución de la prueba debe contener un resumen que ofrezca una descripción general de los resultados de la ejecución, el sistema que se está probando y el entorno en el que se ejecutaron las pruebas, que es apropiado para cada una de las partes interesadas.

Es necesario saber qué pruebas han fallado y las razones del fallo. Para facilitar la resolución de problemas, es importante conocer el historial de la ejecución de la prueba y quién es responsable de ella (generalmente la persona que la creó o la actualizó por última vez). La persona responsable debe investigar la causa del fallo, informar los problemas relacionados con el, hacer un seguimiento de la solución de los problemas y verificar que la solución se haya implementado correctamente.

Los informes también se utilizan para diagnosticar cualquier fallo de los componentes del TAF (consulte el Capítulo 7).

Publicación de los informes

El informe debe ser publicado para todas las partes interesadas en los resultados de la ejecución. Puede cargarse en un sitio Web, enviarse a una lista de correo o cargarse a otra herramienta, como una herramienta de gestión de pruebas. Desde un punto de vista práctico, es muy probable que las partes interesadas en el resultado de la ejecución lo vean y lo analicen si se les otorga un servicio de suscripción y pueden recibir el informe por correo electrónico.

La opción es identificar las partes problemáticas del SSP, es mantener un historial de los informes, para que se puedan recopilar estadísticas sobre casos de prueba o juegos de prueba con regresiones frecuentes.

6 Transición de las Pruebas Manuales a un Entorno Automatizado - 120 minutos

Palabras clave

pruebas de confirmación, pruebas de regresión

Objetivos de Aprendizaje para la Transición de las Pruebas Manuales a un Entorno Automatizado

6.1 Criterios para la Automatización

ALTA-E-6.1.1 (K3) Aplicar criterios para determinar la adecuación de las pruebas para la automatización

ALTA-E-6.1.2 (K2) Comprender los factores en la transición de pruebas manuales a la automatización

6.2 Identificar los Pasos Necesarios para Implementar la Automatización dentro de las Pruebas de Regresión

ALTA-E-6.2.1 (K2) Explicar los factores a tener en cuenta al implementar pruebas de regresión automatizadas

6.3 Factores a Tener en Cuenta al Implementar la Automatización dentro de las Pruebas de Nuevas Prestaciones

ALTA-E-6.3.1 (K2) Explicar los factores a tener en cuenta al implementar pruebas de regresión automatizadas

6.4 Factores a Tener en Cuenta al Implementar la Automatización dentro de las Pruebas de Confirmación

ALTA-E-6.4.1 (K2) Explicar los factores a tener en cuenta al implementar pruebas de confirmación automatizadas

6.1 Criterios para la Automatización

Tradicionalmente, las organizaciones han desarrollado casos de prueba manuales. Al decidir migrar hacia un entorno de prueba automatizado, uno debe evaluar el estado actual de las pruebas manuales y determinar el enfoque más efectivo para automatizar estos recursos de prueba. La estructura existente de una prueba manual puede o no ser adecuada para la automatización, en cuyo caso puede ser necesaria una reescritura completa de la prueba para respaldar la automatización. Alternativamente, los componentes relevantes de las pruebas manuales existentes (p. ej., valores de entrada, resultados esperados, ruta de navegación) pueden extraerse de las pruebas manuales existentes y reutilizarse para su automatización. Una estrategia de prueba manual que tenga en cuenta la automatización permitirá realizar pruebas cuya estructura facilita la migración a la automatización.

No todas las pruebas pueden o deben ser automatizadas, y algunas veces la primera iteración de una prueba puede ser manual. Por lo tanto, hay dos aspectos de la transición a tener en cuenta: la conversión inicial de las pruebas manuales existentes a la automatización, y la posterior transición de las nuevas pruebas manuales a la automatización.

También tenga presente que ciertos tipos de pruebas solo pueden ejecutarse (efectivamente) de manera automatizada, p. ej., pruebas de confiabilidad, pruebas de estrés o pruebas de rendimiento.

Con la automatización de pruebas es posible probar aplicaciones y sistemas sin una interfaz de usuario. En este caso, las pruebas se pueden realizar en el nivel de integración a través de interfaces en el software. Si bien este tipo de casos de prueba también podrían ejecutarse manualmente (usando comandos ingresados manualmente para activar las interfaces), esto puede no ser práctico. Por ejemplo, con la automatización puede ser posible insertar mensajes en un sistema de cola de mensajes. De esta manera, las pruebas pueden comenzar antes (y pueden identificar defectos antes), cuando las pruebas manuales aún no son posibles.

Antes de comenzar un esfuerzo de prueba automatizado, uno debe considerar la aplicabilidad y la viabilidad de crear pruebas automatizadas vs. manuales. Los criterios de idoneidad pueden incluir, pero no se limitan a:

- Frecuencia de uso
- Complejidad para automatizar
- Compatibilidad del soporte de herramientas
- Madurez de los procesos de prueba
- Adecuación de la automatización para la etapa del ciclo de vida del producto de software
- Sostenibilidad del entorno automatizado
- Controlabilidad del SSP

Cada uno de ellos se explica con más detalle a continuación.

Frecuencia de uso

La frecuencia con la que se debe ejecutar una prueba es una consideración en cuanto a la viabilidad de automatizar o no. Las pruebas que se realizan con mayor frecuencia, como parte de un ciclo de entrega mayor o menor, son mejores candidatos para la automatización, ya que se utilizarán con frecuencia. Como regla general, cuanto mayor sea el número de entregas de aplicaciones y, por lo tanto, los ciclos de prueba correspondientes, mayor será el beneficio de automatizar las pruebas.

A medida que las pruebas funcionales se automatizan, se pueden usar en entregas posteriores como parte de las pruebas de regresión. Las pruebas automatizadas utilizadas en las pruebas de regresión proporcionarán un alto retorno de la inversión (ROI) y mitigación de riesgos para el código base existente.

Si se ejecuta un guión de prueba una vez al año y el SSP cambia dentro del año, puede que no sea factible o eficiente crear una prueba automatizada. El tiempo que puede tomar adaptar la prueba una vez al año

para cumplir con el SSP se puede hacer mejor de forma manual.

Complejidad para automatizar

En los casos en los que se necesita probar un sistema complejo, puede que la automatización tenga un gran beneficio al ahorrarle al probador manual la difícil tarea de repetir pasos complejos que son tediosos, requieren mucho tiempo y son propensos a errores.

Sin embargo, ciertos guiones pueden ser difíciles o no rentables de automatizar. Existe una variedad de factores que pueden afectar esto, incluyendo: un SSP que no es compatible con las soluciones de prueba automatizadas existentes; el requisito de producir un código de programa sustancial y desarrollar llamadas a las API para automatizar; la multiplicidad de sistemas que deben abordarse como parte de una ejecución de prueba; la interacción con interfaces externas y/o sistemas propietarios; algunos aspectos de las pruebas de usabilidad; la cantidad de tiempo necesario para probar los guiones de automatización, etc.

Compatibilidad y soporte de herramientas

Existe una amplia gama de plataformas de desarrollo utilizadas para crear aplicaciones. El desafío para el probador es saber qué herramientas de prueba disponibles existen (si las hay) que soporten una plataforma determinada y en qué medida la plataforma es soportada. Las organizaciones utilizan una variedad de herramientas de prueba, incluidas las de proveedores comerciales, de código abierto y desarrolladas internamente. Cada organización tendrá diferentes necesidades y recursos para apoyar las herramientas de prueba. Los proveedores comerciales suelen ofrecer asistencia de pago y, en el caso de los líderes del mercado, suelen contar con un ecosistema de expertos que pueden ayudar en la implementación de la herramienta de prueba. Las herramientas de código abierto pueden ofrecer asistencia, como foros en línea desde los cuales los usuarios pueden obtener información y enviar una pregunta a una lista de correo. Las herramientas de prueba desarrolladas internamente dependen del personal existente para brindar apoyo.

El problema de la compatibilidad de la herramienta de prueba no debe ser subestimado. Embarcarse en un proyecto de automatización de pruebas sin comprender completamente el nivel de compatibilidad entre las herramientas de prueba y el SSP puede tener resultados desastrosos. Incluso si la mayoría de las pruebas para el SSP se pueden automatizar, puede haber una situación en la que las pruebas más críticas no puedan automatizarse.

Madurez de los procesos de prueba

Para implementar la automatización de manera efectiva dentro de un proceso de prueba, ese proceso debe ser estructurado, disciplinado y repetible. La automatización conlleva un proceso de desarrollo completo al proceso de prueba existente que requiere la administración del código de automatización y los componentes relacionados.

Adecuación de la automatización para la etapa del ciclo de vida del producto de software

Un SSP tiene un ciclo de vida del producto que puede abarcar desde años hasta décadas. A medida que comienza el desarrollo de un sistema, el sistema cambia y se expande para solucionar defectos y agregar mejoras para satisfacer las necesidades del usuario final. En las primeras etapas del desarrollo de un sistema, el cambio puede ser demasiado rápido para implementar una solución de prueba automatizada. Como los diseños y controles de la pantalla están optimizados y mejorados, la creación de automatización en un entorno que cambia dinámicamente puede necesitar un trabajo continuo, que no es eficiente ni efectivo. Esto sería similar a tratar de cambiar un neumático en un automóvil en movimiento; es mejor esperar a que el auto se detenga. Para sistemas grandes en un entorno de desarrollo secuencial, cuando un sistema se ha estabilizado e incluye un núcleo de funcionalidad, este se convierte en el mejor momento para comenzar la implementación de pruebas automatizadas.

Con el tiempo, los sistemas llegan al final de los ciclos de vida de sus productos y se retiran o se rediseñan para utilizar una tecnología más nueva y más eficiente. La automatización no se recomienda para un sistema que se acerca al final de su ciclo de vida, ya que tendrá poco valor emprender una iniciativa tan breve. Sin embargo, para los sistemas que se están rediseñando utilizando una arquitectura diferente pero

conservando la funcionalidad existente, un entorno de prueba automatizado que define elementos de datos será igualmente útil en los sistemas antiguos y nuevos. En este caso, la reutilización de los datos de prueba sería posible y la recodificación del entorno automatizado para que sea compatible con la nueva arquitectura sería necesaria.

Sostenibilidad del entorno

Un entorno de prueba para la automatización debe ser flexible y adaptable a los cambios que se producirán en el SSP a lo largo del tiempo. Esto incluye la capacidad de diagnosticar rápidamente y corregir problemas con la automatización, la facilidad con la que se pueden mantener los componentes de la automatización y la facilidad con la que se pueden agregar nuevas prestaciones y soporte en el entorno automatizado. Estos atributos son una parte integral del diseño general y la implementación del AAPg.

Controlabilidad del SSP (condiciones previas, configuración y estabilidad)

El IAP debe identificar las características de control y visibilidad en el SSP que ayudarán en la creación de pruebas automatizadas efectivas. De lo contrario, la automatización de la prueba se basa únicamente en las interacciones de la interfaz de usuario, lo que da como resultado una solución de automatización de la prueba menos mantenible. Consulte la Sección 2.3 sobre Diseño para la Capacidad de ser Probado y la Automatización para obtener más información.

Planificación técnica en apoyo al análisis del ROI

La automatización de pruebas puede proporcionarle a un equipo de pruebas diversos grados de beneficio. Sin embargo, un nivel significativo de esfuerzo y costo está asociado con la implementación de una solución de prueba automatizada efectiva. Antes de incurrir en el tiempo y el esfuerzo para desarrollar pruebas automatizadas, se debe realizar una evaluación para conocer cuáles podrían ser los beneficios generales previstos y potenciales y el resultado de la implementación de la automatización de la prueba. Una vez determinado esto, se deben definir las actividades necesarias para llevar a cabo un plan de este tipo y se deben determinar los costos asociados para calcular el ROI.

Para prepararse adecuadamente para la transición a un entorno automatizado, deben abordarse las siguientes áreas:

- Disponibilidad de herramientas en el entorno de prueba para automatización de la prueba
- Exactitud de los datos de prueba y otros recursos necesarios
- Alcance del esfuerzo de automatización de la prueba
- Formación del equipo de prueba para el cambio de paradigma
- Funciones y responsabilidades
- Cooperación entre desarrolladores e ingenieros de automatización de pruebas
- Esfuerzo paralelo
- Gestión de Información de Automatización de la Prueba

Disponibilidad de herramientas en el entorno de prueba para automatización de la prueba

Las herramientas de prueba seleccionadas deben instalarse y confirmarse para que funcionen en el entorno del laboratorio de pruebas. Esto puede implicar la descarga de cualquier paquete de servicio o actualizaciones de las entregas, la selección de la configuración de instalación adecuada, incluidos los complementos, necesarios para respaldar el SSP, y garantizar que las funciones de la SAP en el entorno del laboratorio de pruebas sean correctas en comparación con el entorno de desarrollo de automatización.

Exactitud de los datos de prueba y otros recursos necesarios

La exactitud y la integridad de los datos de prueba manuales y los casos de prueba son necesarios para garantizar que el uso con la automatización proporcionará resultados predecibles. Las pruebas ejecutadas bajo la automatización necesitan datos explícitos para la entrada, navegación, sincronización y validación.

Alcance del esfuerzo de automatización de la prueba

Para mostrar el éxito inicial en la automatización y obtener retroalimentación sobre problemas técnicos que

pueden afectar el avance, comenzar con un alcance limitado facilitará las futuras tareas de automatización. Un proyecto piloto puede estar dirigido a un área de la funcionalidad de un sistema que sea representativa de la interoperabilidad general del sistema. Las lecciones aprendidas del proyecto piloto ayudarán a ajustar las estimaciones y cronogramas de tiempo futuros, e identificar áreas que requieren recursos técnicos especializados. Un proyecto piloto proporciona una forma rápida de mostrar el éxito inicial de la automatización, lo que refuerza el apoyo de la administración.

Para ayudar en esto, los casos de prueba que se van a automatizar deben ser seleccionados acertadamente. Seleccione los casos que requieran poco esfuerzo para automatizarlos, pero que proporcionen un alto valor agregado. Se pueden implementar pruebas automáticas de regresión o humo y agregar un valor considerable, ya que estas pruebas normalmente se ejecutan con bastante frecuencia, incluso a diario. Otro buen candidato para comenzar son las pruebas de confiabilidad. Estas pruebas a menudo se componen de pasos y se ejecutan una y otra vez, revelando problemas que son difíciles de detectar manualmente. Estas pruebas de confiabilidad requieren poco esfuerzo para su implementación, pero pueden mostrar un valor agregado muy pronto.

Estos proyectos piloto ponen la automatización en el punto de mira (esfuerzo de prueba manual ahorrado, o se identifican problemas graves) y despejan el camino para nuevas extensiones (esfuerzo y dinero).

Además, se debe dar prioridad a las pruebas que son críticas para la organización, ya que éstas mostrarán desde el comienzo el mayor valor. Sin embargo, dentro de este contexto, es importante que, como parte de un esfuerzo piloto, se eviten las pruebas más difíciles de automatizar técnicamente. De lo contrario, se dedicará demasiado esfuerzo al desarrollo de la automatización con muy pocos resultados que mostrar. Como regla general, la identificación de pruebas que comparten características con gran parte de la aplicación proporcionará el impulso necesario para mantener vivo el esfuerzo de automatización.

Formación del equipo de prueba para cambio de paradigma

Los probadores son de muchos tipos: algunos son expertos en dominios provenientes de la comunidad de usuarios finales o involucrados como analistas de negocios, mientras que otros tienen habilidades técnicas sólidas que les permiten comprender mejor la arquitectura del sistema subyacente. Para que las pruebas sean efectivas, es preferible una amplia combinación de premisas. A medida que el equipo de pruebas pase a la automatización, las funciones se volverán más especializadas. Cambiar la composición del equipo de prueba es esencial para que la automatización tenga éxito, y formar al equipo mucho antes del cambio intencionado ayudará a reducir la ansiedad sobre las funciones o el posible pensamiento de haberlo hecho redundante. Cuando se aborda correctamente, el cambio hacia la automatización debe hacer que todos en el equipo de prueba estén muy entusiasmados y listos para participar en el cambio organizativo y técnico.

Funciones y responsabilidades

La automatización de pruebas debe ser una actividad en la que todos puedan participar. Sin embargo, eso no equivale a que todos tengan la misma función. Diseñar, implementar y mantener un entorno de prueba automatizado es de naturaleza técnica y, como tal, debe reservarse para personas con sólidas habilidades de programación y trayectoria profesional técnica. Los resultados de un esfuerzo de desarrollo de pruebas automatizadas deben ser un entorno que puedan utilizar tanto las personas técnicas como las no técnicas. Para maximizar el valor de un entorno de prueba automatizado, se necesitan personas con experiencia en el dominio y habilidades de prueba, ya que será necesario desarrollar los guiones apropiados (incluidos los datos de prueba correspondientes). Estos se utilizarán para impulsar el entorno automatizado y proporcionar la cobertura de pruebas específicas. Los expertos en dominios revisan los informes para confirmar la funcionalidad de la aplicación, mientras que los expertos técnicos se aseguran de que el entorno automatizado funcione de manera correcta y eficiente. Estos expertos técnicos también pueden ser desarrolladores interesados en las pruebas. La experiencia en el desarrollo de software es esencial para diseñar software que se pueda mantener, y esto es de suma importancia en la automatización de pruebas. Los desarrolladores pueden centrarse en el marco de trabajo automatización de la prueba o en las bibliotecas de pruebas. La implementación de casos de prueba debe permanecer con los probadores.

Cooperación entre desarrolladores e ingenieros de automatización de pruebas

La automatización exitosa de las pruebas también requiere la participación del equipo de desarrollo de software, así como de los probadores. Los desarrolladores y probadores deben trabajar mucho más estrechamente para la automatización de la prueba, de modo que los desarrolladores puedan proporcionar personal de apoyo e información técnica sobre sus métodos y herramientas de desarrollo. Los ingenieros de automatización de la prueba pueden plantear inquietudes acerca de la capacidad de ser probado de los diseños de sistemas y del código de desarrollador. Este será especialmente el caso si no se siguen las normas, o si los desarrolladores usan bibliotecas/objetos extraños, de cosecha propia o incluso muy nuevos. Por ejemplo, los desarrolladores pueden elegir un control de la GUI de terceros que puede no ser compatible con la herramienta de automatización seleccionada. Finalmente, el equipo gestión de proyectos de una organización debe tener un entendimiento claro sobre los tipos de funciones y responsabilidades requeridas para un esfuerzo de automatización exitoso.

Esfuerzo paralelo

Como parte de las actividades de transición, muchas organizaciones crean un equipo paralelo para comenzar el proceso de automatización de los guiones manuales existentes. Los nuevos guiones automatizados se incorporan luego al esfuerzo de prueba, reemplazando los guiones manuales. Sin embargo, antes de hacerlo, a menudo se recomienda comparar y validar que el guión automatizado está realizando la misma prueba y validación que el guión manual al que reemplaza.

En muchos casos, se realizará una evaluación de los guiones manuales antes de la conversión a la automatización. Como resultado de dicha evaluación, se podría determinar que existe la necesidad de reestructurar los guiones manuales existentes para lograr un enfoque más eficiente y efectivo bajo la automatización.

Gestión de información de la automatización

Hay varios informes que pueden ser generados automáticamente por una SAP. Estos incluyen el estado de paso/fallo de guiones individuales o pasos dentro de un guión, estadísticas generales de ejecución de prueba y rendimiento general de la SAP. Es igualmente importante tener visibilidad del funcionamiento correcto de la SAP para que los resultados específicos de las aplicaciones que se informan puedan considerarse precisos y completos (consulte el Capítulo 7: Verificación de la SAP).

6.2 Identificar los Pasos Necesarios para Implementar la Automatización dentro de las Pruebas de Regresión

Las pruebas de regresión proporcionan una gran oportunidad para utilizar la automatización. Un banco de pruebas de regresión crece a medida que las pruebas funcionales de hoy se convierten en las pruebas de regresión del mañana. Es solo una cuestión de tiempo antes de que el número de pruebas de regresión sea mayor que el tiempo y los recursos disponibles para un equipo de prueba manual tradicional.

En el desarrollo de pasos para prepararse para automatizar las pruebas de regresión, se deben hacer varias preguntas:

- ¿Con qué frecuencia se deben ejecutar las pruebas?
- ¿Cuál es el tiempo de ejecución para cada prueba, para el conjunto de regresión?
- ¿Existe superposición funcional entre las pruebas?
- ¿Las pruebas comparten datos?
- ¿Las pruebas dependen unas de otras?
- ¿Qué condiciones previas se requieren antes de la ejecución de la prueba?
- ¿Qué % de cobertura del SSP representan las pruebas?
- ¿Las pruebas actualmente se ejecutan sin fallos?
- ¿Qué debe suceder cuando las pruebas de regresión toman demasiado tiempo?

Cada uno de ellos se explica con más detalle a continuación.

Frecuencia de ejecución de la prueba

Las pruebas que se ejecutan a menudo como parte de las pruebas de regresión son los mejores candidatas para la automatización. Estas pruebas ya se han desarrollado, ejercen la funcionalidad conocida del SSP y se reducirá enormemente su tiempo de ejecución mediante el uso de la automatización.

Tiempo de ejecución de prueba

El tiempo que se tarda en ejecutar una prueba determinada o un conjunto completo de pruebas es un parámetro importante para evaluar el valor de implementar pruebas automatizadas dentro de las pruebas de regresión. Una opción es comenzar implementando la automatización en pruebas que consumen mucho tiempo. Esto permitirá que cada prueba se ejecute de forma más rápida y eficiente, al tiempo que agrega ciclos adicionales de ejecución de pruebas de regresión automatizadas. El beneficio es una retroalimentación adicional y más frecuente sobre la calidad del SSP y un menor riesgo de implementación.

Superposición funcional

Al automatizar las pruebas de regresión existentes, es una buena práctica identificar cualquier superposición funcional que exista entre los casos de prueba y, cuando sea posible, reducir esa superposición en la prueba automatizada equivalente. Esto resultará en más eficiencia en el tiempo de ejecución de la prueba automatizada, lo que será significativo a medida que se ejecuten más y más casos de prueba automatizados. A menudo, las pruebas desarrolladas utilizando la automatización tomarán una nueva estructura, ya que dependen de componentes reutilizables y repositorios de datos compartidos. No es inusual descomponer las pruebas manuales existentes en varias pruebas automatizadas más pequeñas. Del mismo modo, la consolidación de varias pruebas manuales en una prueba automatizada más grande puede ser la solución adecuada. Las pruebas manuales deben evaluarse individualmente, y como grupo, para que se pueda desarrollar una estrategia de conversión efectiva.

Compartir datos

Las pruebas a menudo comparten datos. Esto puede ocurrir cuando las pruebas usan el mismo registro de datos para ejecutar diferentes 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" puede

verificar qué cursos tomó el empleado como parte de sus objetivos de desarrollo profesional. Cada caso de prueba usa al mismo empleado, pero verifica diferentes parámetros. En un entorno de prueba manual, los datos de los empleados normalmente se duplicarían muchas veces en cada caso de prueba manual que verificaba los datos de los empleados utilizando a este empleado. Sin embargo, en una prueba automatizada, los datos que se comparten deben, donde sea posible y factible, ser almacenados y accedidos desde una sola fuente para evitar la duplicación o la introducción de errores.

Interdependencia de las pruebas

Al ejecutar escenarios de pruebas de regresión complejos, una prueba puede depender de una o de otras pruebas más. Esta ocurrencia puede ser bastante común y puede ocurrir, a modo de ejemplo, como resultado de un nuevo "ID de pedido" que se crea como resultado de un paso de prueba. Es posible que las pruebas posteriores deseen verificar que: a) la nueva orden se muestra correctamente en el sistema, b) los cambios en la orden son posibles, o la eliminación de la orden es exitosa. En cada caso, el valor de "ID de pedido" que se crea dinámicamente en la primera prueba debe ser capturado para su reutilización por pruebas posteriores. Dependiendo del diseño de la SAP, este puede ser abordado.

Condiciones previas de prueba

A menudo, una prueba no se puede ejecutar antes de establecer las condiciones iniciales. Estas condiciones pueden incluir seleccionar la base de datos correcta o el conjunto de datos de prueba a partir de los cuales probar, o establecer valores o parámetros iniciales. Muchos de estos pasos de inicialización que se requieren para establecer la condición previa de una prueba se pueden automatizar. Esto permite una solución más confiable y segura cuando estos pasos no se pueden perder antes de ejecutar las pruebas. Como las pruebas de regresión se convierten en automatización, estas condiciones previas deben ser parte del proceso de automatización.

Cobertura del SSP

Cada vez que se ejecutan las pruebas, se ejerce parte de la funcionalidad de un SSP. Para determinar la calidad general del SSP, las pruebas deben diseñarse para que tengan la cobertura más amplia y profunda. Además, las herramientas de cobertura de código pueden usarse para monitorizar la ejecución de pruebas automatizadas para ayudar a cuantificar la efectividad de las pruebas. A través de las pruebas de regresión automatizadas, con el tiempo podemos esperar que las pruebas adicionales proporcionen cobertura adicional. La medición de esta proporciona un medio eficaz para cuantificar el valor de las pruebas en sí.

Pruebas ejecutables

Antes de convertir una prueba de regresión manual en una prueba automatizada, es importante verificar que la prueba manual funciona correctamente. Esto proporciona el punto de partida correcto para garantizar una conversión exitosa a una prueba de regresión automatizada. Si la prueba manual no se ejecuta correctamente, ya sea porque fue mal escrita, usa datos no válidos, está desactualizada o no está sincronizada con el SSP actual, o como resultado de un defecto del SSP, convertirlo a automatización antes de comprenderlo y/o resolver la causa raíz del fallo creará una prueba automatizada que no funciona y que es inútil e improductiva.

Pruebas de regresión grandes

El conjunto de pruebas de regresión para un SSP puede llegar a ser bastante grande, tan grande que el conjunto de prueba no se puede ejecutar completamente durante la noche o durante el fin de semana. En ese caso, la ejecución concurrente de casos de prueba es una posibilidad si hay varios SSP disponibles (para aplicaciones de PC, esto probablemente no plantea un problema, pero cuando el SSP consiste en un avión o cohete espacial, esta es una historia diferente). Los SSP pueden ser escasos y/o costosos, lo que hace que la concurrencia sea una opción poco realista. En este caso, una posibilidad puede ser ejecutar solo partes de la prueba de regresión. Con el tiempo (semanas) se ejecutará el conjunto completo. La elección de qué parte del conjunto de pruebas de regresión ejecutar también puede basarse en un análisis de riesgo (¿qué partes del SSP se han modificado últimamente?).

6.3 Factores a Tener en Cuenta al Implementar la Automatización dentro de las Pruebas de Nuevas Prestaciones

En general, es más fácil automatizar los casos de prueba para una nueva funcionalidad, ya que la implementación aún no está terminada (o mejor aún, aún no se ha iniciado). El ingeniero de pruebas puede utilizar su conocimiento para explicarle a los desarrolladores y arquitectos qué se necesita exactamente en la nueva funcionalidad, de modo que la solución de automatización de pruebas pueda probarla de manera efectiva y eficiente.

A medida que se introducen nuevas prestaciones en un SSP, se requiere que los probadores desarrollen nuevas pruebas contra estas nuevas prestaciones y los requisitos correspondientes. El IAP debe solicitar retroalimentación de los diseñadores de pruebas con experiencia en el dominio y determinar si la SAP actual cumplirá con las necesidades de las nuevas prestaciones. Este análisis incluye, pero no se limita a, el enfoque existente utilizado, herramientas de desarrollo de terceros, herramientas de prueba utilizadas, etc.

Los cambios en la SAP deben evaluarse, comparándolos con los componentes de productos de prueba automatizados existentes para que los cambios o adiciones estén completamente documentados y no afecten el comportamiento (o el rendimiento) de la funcionalidad de la SAP existente.

Si se implementa una nueva prestación con, p. ej., una clase diferente de objeto, puede ser necesario realizar actualizaciones o adiciones a los componentes de productos de prueba. Además, se debe evaluar la compatibilidad con las herramientas de prueba existentes y, cuando sea necesario, se deben identificar soluciones alternativas. Por ejemplo, si utiliza un enfoque guiado por palabras clave, puede ser necesario desarrollar palabras clave adicionales o modificar/expandir las palabras clave existentes para adaptarse a la nueva funcionalidad.

Puede haber un requisito para evaluar herramientas de prueba adicionales que admitan el nuevo entorno en el que existe la nueva funcionalidad. Por ejemplo, una nueva herramienta de prueba podría ser necesaria si la herramienta de prueba existente solo admite HTML.

Los nuevos requisitos de prueba pueden afectar las pruebas automatizadas existentes y los componentes de productos de prueba. Por lo tanto, antes de realizar cualquier cambio, las pruebas automatizadas existentes deben ejecutarse en el SSP nuevo/actualizado para verificar y registrar cualquier cambio en el funcionamiento correcto de las pruebas automatizadas existentes. Esto debe incluir el mapeo de interdependencias a otras pruebas. Cualquier cambio nuevo en la tecnología requerirá la evaluación de los componentes actuales de productos de prueba (incluidas las herramientas de prueba, las bibliotecas de funciones, las API, etc.) y la compatibilidad con la SAP existente.

Cuando los requisitos existentes cambian, el esfuerzo por actualizar los casos de prueba que verifican estos requisitos debe formar parte del cronograma del proyecto (estructura de descomposición del trabajo). La trazabilidad desde los requisitos hasta los casos de prueba indicará qué casos de prueba deben actualizarse. Estas actualizaciones deben ser parte del plan general.

Finalmente, uno necesita determinar si la SAP existente continuará satisfaciendo las necesidades actuales del SSP. ¿Las técnicas de implementación siguen siendo válidas o se requiere una nueva arquitectura? y ¿Se puede hacer esto extendiendo la capacidad actual?

Cuando se introduce una nueva funcionalidad, esta es una oportunidad para que los ingenieros de pruebas se aseguren de que la funcionalidad recién definida se pueda probar. Durante la fase de diseño, las pruebas deben tenerse en cuenta al planear proporcionar interfaces de prueba que pueden ser utilizadas por los lenguajes de escritura de guión o la herramienta de automatización de pruebas para verificar la nueva funcionalidad. Consulte la Sección 2.3, Diseño para la Capacidad de ser Probado y la

Automatización para, obtener más información.

6.4 Factores a Tener en Cuenta al Implementar la Automatización dentro de las Pruebas de Confirmación

La prueba de confirmación se realiza después de una corrección de código que trata un defecto reportado. Un probador generalmente sigue los pasos necesarios para replicar el defecto para verificar que el defecto ya no existe.

Los defectos tienen una forma de reintroducirse en entregas posteriores (esto puede indicar un problema de administración de la configuración) y, por lo tanto, las pruebas de confirmación son las principales candidatas para la automatización. El uso de la automatización ayudará a reducir el tiempo de ejecución para las pruebas de confirmación. La prueba de confirmación se puede agregar y complementar al banco de pruebas de regresión automatizada existente.

La prueba de confirmación automatizada normalmente tiene un alcance limitado de funcionalidad. La implementación puede ocurrir en cualquier momento una vez que se informa un defecto y se comprenden los pasos necesarios para replicarlo. Las pruebas de confirmación automatizadas se pueden incorporar en una serie de regresión automatizada estándar o, cuando sea práctico, se pueden incluir en las pruebas automatizadas existentes. Con cualquiera de los dos enfoques, el valor de la automatización de las pruebas de confirmación de defectos aún se mantiene.

El seguimiento de las pruebas de confirmación automatizadas permite informes adicionales del tiempo y la cantidad de ciclos utilizados en la resolución de defectos.

Además de la confirmación que las pruebas de regresión son necesarias para asegurar que los nuevos defectos no hayan sido introducidos como un efecto secundario del arreglo del defecto. Se puede requerir que el análisis del impacto determine el alcance apropiado de las pruebas de regresión.

7 Verificación de la SAP - 120 minutos

Palabras clave

verificación

Objetivos de Aprendizaje de la Verificación de la SAP

7.1 Verificación de los Componentes del Entorno de Prueba Automatizado

ALTA-E-7.1.1 (K3) Verificar la exactitud de un entorno de prueba automatizado, incluida la configuración de la herramienta de prueba

7.2 Verificación del Juego de Pruebas Automatizadas

ALTA-E-7.2.1 (K3) Verificar el comportamiento correcto para un guión de prueba automatizado dado y/o juego de pruebas



7.1 Verificación de los Componentes del Entorno de Prueba Automatizado

El equipo de automatización de la prueba tiene que verificar que el ambiente de prueba automatizado trabaja según lo esperado. Estas verificaciones se hacen, por ejemplo, antes de comenzar las pruebas automatizadas.

Hay varios pasos que se pueden realizar para verificar los componentes del entorno de prueba automatizado. Cada uno de ellos se explica con más detalle a continuación.

Instalación de la herramienta de prueba, montaje, configuración y personalización

La SAP está integrada por muchos componentes. Cada uno de ellos necesita ser explicado para asegurar su rendimiento confiable y repetible. En el núcleo de una SAP se encuentran los componentes ejecutables, bibliotecas funcionales correspondientes, y datos de apoyo y archivos de configuración. El proceso de configuración de una SAP puede variar, desde el uso de guiones de instalación automatizados a colocar manualmente los archivos en las carpetas correspondientes. Las herramientas de prueba, más parecidas a sistemas operativos y otras aplicaciones, con regularidad tienen service packs o pueden tener programas de extensión opcionales o necesarios para asegurar la compatibilidad con cualquier entorno de SSP dado.

La instalación automatizada (o copia) desde un repositorio central tiene ventajas. Se puede garantizar que las pruebas sobre SSPs diferentes se han realizado con la misma versión de la SAP y la misma configuración de la SAP, donde sea pertinente. Las actualizaciones a la SAP se pueden hacer a través del repositorio. El uso del repositorio y el proceso para actualizar a una nueva versión de la SAP deben ser iguales que para herramientas de desarrollo estándares.

Escritura de guiones de prueba con pasos y fallos conocidos

Cuándo el paso de casos de prueba conocidos fallan, está inmediatamente claro que algo es esencialmente incorrecto y debe ser corregido cuanto antes. Por el contrario, cuando los casos de prueba pasan aunque debieran haber fallado, tenemos que identificar el componente que no funcionó correctamente. Es importante verificar la generación correcta de archivos de registro y métricas de rendimiento así como el montaje y desmontaje automatizado del caso/guión de prueba. También es útil ejecutar unas pruebas de los diferentes tipos y niveles de prueba (pruebas funcionales, pruebas de rendimiento, pruebas de componentes, etc.). Esto también se debe hacer a nivel del marco de trabajo.

Repetibilidad en la configuración/desmontaje del entorno de prueba

Se implementará una SAP en una variedad de sistemas y servidores. Para garantizar que la SAP funcione correctamente en cada entorno, es necesario tener un enfoque sistemático para cargar y descargar la SAP desde cualquier entorno dado. Esto se logra con éxito cuando la compilación y recompilación de la SAP no proporciona una diferencia perceptible en la forma en que opera dentro y entre múltiples entornos. La administración de la configuración de los componentes de la SAP garantiza que se pueda crear una configuración determinada de manera confiable.

Configuración del entorno de prueba y de los componentes

Comprender y documentar los diversos componentes que integran la SAP proporciona el conocimiento necesario sobre qué aspectos de la SAP pueden verse afectados o requieren un cambio cuando cambia el entorno del SSP.

Conectividad contra sistemas/interfaces internos y externos

Una vez que se instala una SAP en un entorno del SSP dado, y antes del uso real contra un SSP, se debe administrar un conjunto de verificaciones o condiciones previas para garantizar que la conectividad a sistemas internos y externos, interfaces, etc., esté disponible. Establecer las condiciones previas para la automatización es esencial para garantizar que la SAP se haya instalado y configurado correctamente.

Intrusión de herramientas de prueba automatizadas

La SAP a menudo se acoplará estrechamente con el SSP. Esto es por diseño, por lo que hay un alto nivel de compatibilidad, especialmente en lo que respecta a las interacciones a nivel de GUI. Sin embargo, esta estrecha integración también puede tener efectos negativos. Estos pueden incluir: Un SSP se comporta de manera diferente cuando la SAP reside dentro del entorno del SSP; el SSP tiene un comportamiento diferente al utilizado manualmente; el rendimiento del SSP se ve afectado con la SAP en el entorno o al ejecutar la SAP contra el SSP.

El nivel de intrusión/agresividad difiere con el enfoque de prueba automatizado elegido. Por ejemplo:

- Al interactuar con el SSP desde interfaces externas, el nivel de intrusión será muy bajo. Las interfaces externas pueden ser señales electrónicas (para interruptores físicos), señales USB para dispositivos USB (como teclados). Con este enfoque, el usuario final es simulado de la mejor manera. En este enfoque, el software del SSP no se modifica en absoluto para pruebas. El comportamiento y la sincronización del SSP no están influenciados por el enfoque de prueba. La interacción con el SSP de esta manera puede ser muy compleja. Podría ser necesario un hardware dedicado, se necesitan lenguajes de descripción de hardware para interactuar con el SSP, etc. Para los sistemas de solo software, este no es un enfoque típico, pero para los productos con software integrado este enfoque es más común.
- Al interactuar con el SSP a nivel de GUI, el entorno del SSP se adapta para inyectar comandos de IU y extraer la información que necesitan los casos de prueba. El comportamiento del SSP no se cambia directamente, pero el tiempo se ve afectado, lo que puede tener un impacto sobre el comportamiento. El nivel de intrusión es más elevado que en el punto anterior, pero la interfaz con el SSP de esta manera es menos compleja. A menudo se pueden utilizar herramientas comerciales disponibles para este tipo de automatización.
- La interfaz con el SSP se puede realizar a través de las interfaces de prueba en el software o utilizando las interfaces ya existentes, proporcionadas por el software. La disponibilidad de estas interfaces (API) es una parte importante del diseño para la capacidad de ser probado. El nivel de intrusión puede ser bastante elevado en este caso. Las pruebas automatizadas utilizan interfaces que los usuarios finales del sistema **no** pueden usar en absoluto (interfaces de prueba) o las interfaces pueden usarse en un contexto diferente al del mundo real. Por otro lado, es muy fácil y económico realizar pruebas automatizadas a través de interfaces (API). Probar el SSP a través de interfaces de prueba puede ser un enfoque sólido siempre que se comprenda el riesgo potencial.

Un alto nivel de intrusión puede mostrar fallos durante las pruebas que no son evidentes bajo las condiciones de uso del mundo real. Si esto causa fallos con las pruebas automatizadas, la confianza en la solución de automatización de pruebas puede disminuir drásticamente. Los desarrolladores pueden necesitar que las fallos identificados por las pruebas automatizadas primero se reproduzcan manualmente, si es posible, para ayudar con el análisis.

Pruebas de Componentes del Marco de Trabajo

Al igual que cualquier proyecto de desarrollo de software, los componentes del marco de trabajo automatizado se deben probar y verificar individualmente. Esto puede incluir pruebas funcionales y no funcionales (rendimiento, utilización de recursos, usabilidad, etc.).

Por ejemplo, los componentes que proporcionan la verificación de objetos en los sistemas de GUI deben probarse para una amplia gama de clases de objetos, para establecer que las funciones de verificación de objetos funcionan correctamente. Del mismo modo, los registros e informes de errores deben generar información precisa sobre el estado de la automatización y el comportamiento del SSP.

Los ejemplos de pruebas no funcionales pueden incluir la comprensión de la degradación del rendimiento del marco de trabajo, la utilización de los recursos del sistema que pueden indicar problemas, como

Probador Certificado

Programa de Estudio de Nivel Avanzado – Ingeniero de
Automatización de la Prueba



Junta Internacional de
Calificaciones de
Pruebas de Software

pérdidas de memoria. Interoperabilidad de componentes dentro y/o fuera del marco de trabajo.



7.2 Verificación del Juego de Pruebas Automatizadas

Las pruebas automatizadas deben probarse para verificar que estén completas, sean coherentes y se comporten correctamente. Se pueden aplicar diferentes tipos de comprobaciones de verificación para asegurarse de que el juego de pruebas automatizadas esté funcionando en un momento dado, o para determinar si es apto para su uso.

Hay varios pasos que se pueden realizar para verificar los componentes del juego de pruebas automatizadas. Éstos incluyen:

- Ejecutar guiones de prueba con pasos y fallos conocidos
- Comprobar el juego de pruebas
- Verificar nuevas pruebas que se centran en las nuevas prestaciones del marco de trabajo
- Tener en cuenta la repetibilidad de las pruebas
- Comprobar que hay suficientes puntos de verificación en el juego de pruebas automatizadas

Cada uno de ellos se explica con más detalle a continuación.

Ejecutar guiones de prueba con pasos y fallos conocidos

Cuando el paso de casos de prueba conocidos fallan, está inmediatamente claro que algo es esencialmente incorrecto y debe ser corregido cuanto antes. Por el contrario, cuando un juego de pruebas pasa aunque debiera haber fallado, es necesario identificar el caso de prueba que no funcionó correctamente. Es importante verificar la generación correcta de archivos de registro, datos de rendimiento, montaje y desmontaje automatizado del caso/guion de prueba. También es útil ejecutar unas pruebas de los diferentes tipos y niveles de prueba (pruebas funcionales, pruebas de rendimiento, pruebas de componentes, etc.).

Comprobar el juego de pruebas

Verifique que el juego de pruebas esté completo (todos los casos de prueba tienen resultados esperados, datos de prueba presentes) y la versión correcta con el marco de trabajo y el SSP.

Verificar nuevas pruebas que se centran en las nuevas prestaciones del marco de trabajo

La primera vez que se usa una nueva prestación de la SAP en los casos de prueba, debe verificarse y monitorizarse de cerca para asegurarse de que la prestación funciona correctamente.

Tener en cuenta repetibilidad de las pruebas

Cuando se repiten las pruebas, el resultado/veredicto de la prueba siempre debe ser el mismo. Los casos de prueba en el juego de pruebas que no dan un resultado confiable (p. ej., condiciones de carrera) podrían trasladarse del juego de pruebas automatizado activo y analizarse por separado para encontrar la causa raíz. De lo contrario, el tiempo se consumirá repetidamente en estas pruebas para analizar el problema.

Los fallos intermitentes deben ser analizados. El problema puede estar en el caso de prueba en sí o en el marco de trabajo (o incluso podría ser un problema en el SSP). El análisis del archivo de registro (del caso de prueba, el marco de trabajo y el SSP) puede identificar la causa raíz del problema. La depuración también puede ser necesaria. El apoyo del analista de pruebas, el desarrollador de software y el experto en dominios puede ser necesario para encontrar la causa raíz.

Comprobar que hay suficientes puntos de verificación en el juego de pruebas automatizadas y/o casos de Prueba

Debe ser posible verificar que el juego de pruebas automatizadas se haya ejecutado y haya logrado los resultados esperados. Se debe proporcionar evidencia para garantizar que el juego de pruebas y/o los casos de prueba se hayan ejecutado como se esperaba. Esta evidencia puede incluir el registro al inicio y al final de cada caso de prueba, el registro del estado de ejecución de la prueba para cada caso de prueba completado, la verificación de que se han alcanzado las condiciones posteriores, etc.

8 Mejora Continua - 150 minutos

Palabras clave

mantenimiento

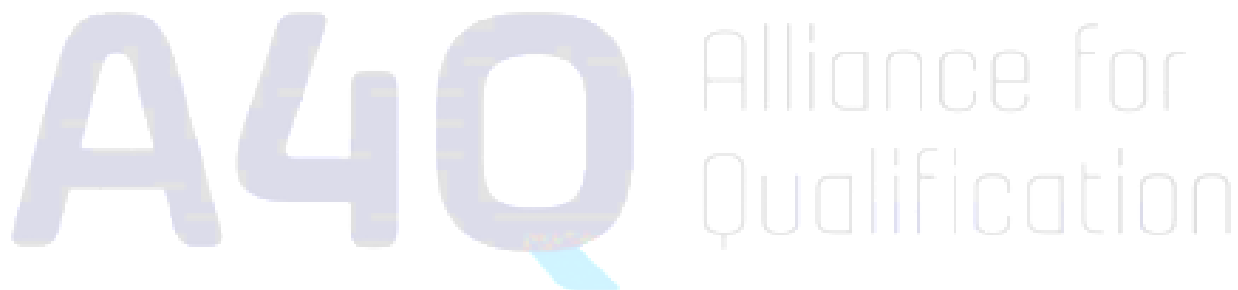
Objetivos de Aprendizaje para la Mejora Continua

8.1 Opciones para Mejorar la Automatización de Pruebas

ALTA-E-8.1.1 (K4) Analizar los aspectos técnicos de una solución de automatización de la prueba desplegada y hacer recomendaciones para su mejora

8.2 Adaptación de la Automatización de la Prueba al Entorno y Cambios del SSP

ALTA-E-8.2.1 (K4) Analizar productos de prueba automatizado, incluidos los componentes, herramientas y bibliotecas de funciones de soporte del entorno de prueba, para comprender dónde deben realizarse las consolidaciones y actualizaciones después de un conjunto determinado de entorno de prueba o cambios del SSP.



8.1 Opciones para Mejorar la Automatización de Pruebas

Además de las tareas de mantenimiento en curso necesarias para mantener la SAP sincronizada con el SSP, generalmente hay muchas oportunidades para mejorar la SAP. Se puede realizar mejoras de la SAP para lograr una gama de beneficios que incluyen una mayor eficiencia (lo que reduce aún más la intervención manual), una mayor facilidad de uso, capacidades adicionales y un mejor soporte para las actividades de prueba. La decisión sobre cómo se mejora la SAP se verá influenciada por los beneficios que agregarán mayor valor a un proyecto.

Las áreas específicas de una SAP que pueden ser consideradas para mejorar incluyen la escritura de guiones, verificación, arquitectura, procesamiento previo y posterior, documentación y soporte de herramientas. Estas están descritas con más detalle a continuación.

Escritura de guiones

Los enfoques de escritura de guiones varían desde el enfoque estructurado simple a los enfoques guiados por datos y a los enfoques más sofisticados basados en palabras clave, como se describe en la Sección 3.2.2. Puede ser apropiado actualizar el enfoque actual de escritura de guiones de la SAP para todas las nuevas pruebas automatizadas. El enfoque se puede adaptar a todas las pruebas automatizadas existentes o al menos a aquellas que impliquen la mayor cantidad de esfuerzo de mantenimiento.

En lugar de cambiar por completo el enfoque de escritura de guiones, las mejoras de la SAP pueden centrarse en la implementación de guiones. Por ejemplo:

- Evaluación de casos de prueba/paso/procedimiento se superponen en un esfuerzo por consolidar pruebas automatizadas.
Los casos de prueba que contienen secuencias de acciones similares no deben implementar estos pasos varias veces. Estos pasos deben convertirse en una función y agregarse a una biblioteca, para que puedan ser reutilizados. Estas funciones de biblioteca pueden ser utilizadas por diferentes casos de prueba. Esto aumenta la mantenibilidad de los productos de prueba. Cuando los pasos de prueba no son idénticos pero similares, puede ser necesaria la parametrización.
Nota: Este es un enfoque típico en las pruebas guiadas por palabras clave.
- Establecer un proceso de recuperación de errores para la SAP y el SSP.
Cuando se produce un error durante la ejecución de los casos de prueba, la SAP debe poder recuperarse de esta condición de error para poder continuar con el siguiente caso de prueba. Cuando se produce un error en el SSP, la SAP debe poder realizar las acciones de recuperación necesarias en el SSP (p. ej., un reinicio del SSP completo).
- Evalúe los mecanismos de espera para asegurarse de que se está utilizando el mejor tipo. Hay tres mecanismos comunes de espera:
 1. Las esperas rígidas (esperan un cierto número de milisegundos) pueden ser la causa raíz de muchos problemas de automatización de pruebas.
 2. La espera dinámica mediante sondeo, p. ej., la comprobación de un determinado cambio de estado o acción ha tenido lugar, es mucho más flexible y eficiente:
 - Espera solo el tiempo necesario y no se pierde tiempo de prueba
 - Cuando, por alguna razón, el proceso toma más tiempo, el sondeo solo esperará hasta que la condición sea verdadera. Recuerde incluir un mecanismo de tiempo de espera, de lo contrario, la prueba puede esperar por siempre en caso de un problema.
 3. Una forma aún mejor es suscribirse al mecanismo de eventos del SSP. Esto es mucho más confiable que las otras dos opciones, pero el lenguaje de escritura de guiones de prueba debe ser compatible con la suscripción a eventos y el SSP debe ofrecerle estos eventos a la aplicación de prueba. Recuerde incluir un mecanismo de tiempo de espera, de lo contrario, la prueba puede esperar por siempre en caso de un problema.
- Trate los productos de prueba como software.

El desarrollo y mantenimiento de productos de prueba es solo una forma de desarrollo de software. Como tales buenas prácticas de codificación (p. ej., utilizando directrices de codificación, análisis estático, revisiones de código) deben aplicarse. Incluso puede ser una buena idea usar desarrolladores de software (en lugar de ingenieros de pruebas) para desarrollar ciertas partes de los productos de prueba (p. ej., bibliotecas).

- Evaluar guiones existentes para su revisión/eliminación.
Varios guiones pueden ser problemáticos (p. ej., fallar de vez en cuando, o altos costos de mantenimiento), y puede ser conveniente rediseñar estos guiones. Otros guiones de prueba pueden eliminarse del juego porque ya no están agregando valor alguno.

Ejecución de prueba

Cuando un juego de pruebas de regresión automatizada no se termina de la noche a la mañana, esto no debe ser una sorpresa. Cuando un juego de pruebas de regresión automatizada no se termina de la noche a la mañana, esto no deb ser una sorpresa. Cuando se usan sistemas costosos (objetivos) para las pruebas, puede ser una limitación que todas las pruebas deban realizarse en un solo objetivo. Puede ser necesario dividir el juego de pruebas de regresión en varias partes, cada una ejecutándose en un período de tiempo definido (p. ej., en una sola noche). Un análisis adicional de la cobertura de prueba automatizada puede revelar duplicación. La eliminación de la duplicación puede reducir el tiempo de ejecución y puede generar más eficiencia. Un análisis adicional de la cobertura de prueba automatizada puede revelar duplicación. La eliminación de la duplicación puede reducir el tiempo de ejecución y puede generar más eficiencia.

Verificación

Antes de crear nuevas funciones de verificación, adopte un conjunto de métodos de verificación estándar para que lo utilicen todas las pruebas automatizadas. Esto evitará la reimplementación de acciones de verificación en múltiples pruebas. Cuando los métodos de verificación no son idénticos pero son similares, el uso de la parametrización ayudará a permitir que una función se use en múltiples tipos de objetos.

Arquitectura

Puede ser necesario cambiar la arquitectura para soportar mejoras de la capacidad de prueba del SSP. Estos cambios pueden realizarse en la arquitectura del SSP y/o en la arquitectura de la automatización. Esto puede proporcionar una mejora importante en la automatización de pruebas, pero puede requerir cambios significativos e inversión en el SSP/la SAP. Por ejemplo, si se va a cambiar el SSP para proporcionar API para las pruebas, la SAP también debe ser refactorizada por ende. Agregar este tipo de prestaciones en una etapa posterior puede ser bastante costoso; es mucho mejor pensar en esto al inicio de la automatización (y en las primeras etapas del desarrollo del SSP, consulte la Sección 2.3 Diseño para la capacidad de Ser Probado y la Automatización).

Procesamiento previo y posterior

Proporcionar tareas de configuración y desmontaje estándar. Estos también se conocen como preprocesamiento (configuración) y postprocesamiento (desmontaje). Esto ahorra las tareas que se implementan repetidamente para cada prueba automatizada, no solo reduciendo los costos de mantenimiento sino también reduciendo el esfuerzo requerido para implementar nuevas pruebas automatizadas.

Documentación

Esto cubre todas las formas de documentación de la documentación de escritura de guiones (qué hacen los guiones, cómo deben usarse, etc.), la documentación del usuario para la SAP y los informes y registros producidos por la SAP.

Prestaciones de la SAP

Agregue prestaciones y funciones adicionales de la SAP, tales como informes detallados, registros, integración a otros sistemas, etc. Solo agregue funciones nuevas cuando éstas sean ciertamente utilizadas. Agregar prestaciones no utilizadas solo aumenta la complejidad y disminuye la confiabilidad y la mantenibilidad.

Actualizaciones y mejoras de la SAP

Al mejorar o actualizar a nuevas versiones de la SAP, pueden estar disponibles nuevas funciones que pueden ser usadas por los casos de prueba (o los fallos pueden ser corregidos). El riesgo es que la actualización del marco de trabajo (ya sea actualizando las herramientas de prueba existentes o introduciendo nuevas) podría tener un impacto negativo sobre los casos de prueba existentes. Pruebe la nueva versión de la herramienta de prueba ejecutando pruebas de muestra antes de implementar la nueva versión. Las pruebas de muestra deben ser representativas de las pruebas automatizadas de diferentes aplicaciones, diferentes tipos de pruebas y, cuando corresponda, diferentes entornos.

8.2 Planificación de la Implementación de la Mejora de la Automatización

Los cambios en una SAP existente requieren una cuidadosa planificación e investigación. Se ha realizado un gran esfuerzo en la creación de una SAP sólida que consiste en un TAF y bibliotecas de componentes. Cualquier cambio, sin importar lo trivial que sea, puede tener un gran impacto en la confiabilidad y el rendimiento de la SAP.

Identificar cambios en los componentes del entorno de prueba

Evaluar qué cambios y mejoras se deben hacer. ¿Estos requieren cambios en el software de prueba, bibliotecas de funciones personalizadas, sistema operativo? Cada uno de ellos tiene un impacto sobre el rendimiento de la SAP. El objetivo general es garantizar que las pruebas automatizadas continúen ejecutándose de manera eficiente. Los cambios deben hacerse de manera incremental para que el impacto sobre la SAP se pueda medir a través de una serie limitada de guiones de prueba. Una vez que se encuentra que no existe un efecto perjudicial, los cambios pueden implementarse completamente. Una regresión completa es el paso final hacia la validación de que el cambio no afectó adversamente a los guiones automatizados. Durante la ejecución de estos guiones de regresión se pueden encontrar errores. La identificación de la causa raíz de estos errores (a través de informes, registros, análisis de datos, etc.) proporcionará un medio para garantizar que no resulten de la actividad de mejora de la automatización.

Aumentar la eficiencia y eficacia de las bibliotecas de funciones de la SAP

A medida que avanza la SAP, se descubren nuevas formas de realizar tareas de manera más eficiente. Estas nuevas técnicas (que incluyen la optimización del código en las funciones, el uso de bibliotecas de sistemas operativos más recientes, etc.) deben incorporarse en las bibliotecas de funciones básicas que se utilizan en el proyecto actual y en todos los proyectos.

Orientar múltiples funciones que actúan en el mismo tipo de control para la consolidación

Una gran parte de lo que ocurre durante una ejecución de prueba automatizada es el examen de los controles en la GUI. Este examen sirve para proporcionar información sobre ese control (p. ej., visible/no visible, habilitado/no habilitado, tamaño y dimensiones, datos, etc.). Con esta información, una prueba automatizada puede seleccionar un elemento de una lista desplegable, ingresar datos en un campo, leer un valor de un campo, etc. Hay varias funciones que pueden actuar sobre los controles para obtener esta información. Algunas funciones son extremadamente especializadas, mientras que otras son de naturaleza más general. Por ejemplo, puede haber una función específica que funcione solo en listas desplegables. Alternativamente, puede haber una función (o una puede ser creada y utilizada dentro de la SAP) que trabaje con varias funciones especificando una función como uno de sus parámetros. Por lo tanto, un IAP puede usar varias funciones que pueden consolidarse en menos funciones, logrando los mismos resultados y minimizando el requisito de mantenimiento.

Refactorizar la AAP para acomodar los cambios en el SSP

A lo largo de la vida de una SAP, se deberán realizar cambios para adaptarse a los cambios en el SSP. A medida que el SSP evolucione y madure, la AAP subyacente también tendrá que evolucionar para garantizar que existe la capacidad para soportar el SSP. Se debe tener cuidado al ampliar las características para que no se implementen de manera complementaria, sino que se analicen y cambien en el nivel arquitectónico de la solución automatizada. Esto asegurará que a medida que la nueva funcionalidad del SSP requiera guiones adicionales, se implementarán componentes compatibles para

Probador Certificado

Programa de Estudio de Nivel Avanzado – Ingeniero de
Automatización de la Prueba



Junta Internacional de
Calificaciones de
Pruebas de Software

acomodar estas nuevas pruebas automatizadas.



Convenciones de nomenclatura y normalización

A medida que se introducen los cambios, las convenciones de nomenclatura para el nuevo código de automatización y las bibliotecas de funciones deben ser coherentes con las normas definidas previamente (consulte la Sección 4.3.2 Alcance y Enfoque).

Evaluación de guiones existentes para la revisión/eliminación del SSP

El proceso de cambio y mejora también incluye una evaluación de los guiones existentes, su uso y valor continuo. Por ejemplo, si ciertas pruebas son complejas y requieren mucho tiempo para ejecutarse, descomponerlas en varias pruebas más pequeñas puede ser más viable y eficiente. Las pruebas de focalización que se ejecutan con poca frecuencia o que no se eliminan en absoluto reducirán la complejidad de la SAP y aportarán mayor claridad a lo que debe mantenerse.



9 Referencias

9.1 Normas

Las normas para la automatización de pruebas incluyen, pero no están limitadas a:

- La Notación de Pruebas y Control de Pruebas (TTCN-3) por el ETSI (Instituto Europeo de Normas de Telecomunicaciones) y la UIT (Unión Internacional de Telecomunicaciones) que está compuesto por
 - ES 201 873-1: TTCN-3 Core Language/*Lenguaje Principal*
 - ES 201 873-2: TTCN-3 Tabular Presentation Format/*Formato de Presentación Tabular (TFT)*
 - ES 201 873-3: TTCN-3 TTCN-3 Graphical Presentation Format/*Formato de Presentación Gráfica (GFT)*
 - ES 201 873-4: TTCN-3 Operational Semantics/*Semántica Operacional*
 - ES 201 873-5: TTCN-3 Runtime Interface/*Interfaz de Tiempo de Ejecución (TRI)*
 - ES 201 873-6: TTCN-3 Control Interface/*Interfaz de Control (TCI)*
 - ES 201 873-7: Using ASN.1 with TTCN-3/*Uso de ASN.1 con TTCN-3*
 - ES 201 873-8: Using IDL with TTCN-3/*Uso de IDL con TTCN-3*
 - ES 201 873-9: Using XML with TTCN-3/*Uso de XML con TTCN-3*
 - ES 201 873-10: TTCN-3 Documentation/*Documentación*
 - ES 202 781: Extensions: Configuration and Deployment Support/*Extensiones: Soporte de Configuración e Implementación*
 - ES 202 782: Extensions: TTCN-3 Performance and Real-Time Testing/*Extensiones: TTCN-3 Rendimiento y Pruebas en Tiempo Real*
 - ES 202 784: Extensions: Advanced Parameterization/*Extensiones: Parametrización Avanzada*
 - ES 202 785: Extensions: Behaviour Types/*Extensiones: Tipos de Comportamiento*
 - ES 202 786: Extensions: Support of interfaces with continuous signals/*Extensiones: Soporte de Interfaces con Señales Continuas*
 - ES 202 789: Extensions: Extended TRI/*Extensiones: TRI Extendido*
- El Lenguaje de marcado de prueba automática (ATML) por el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) que consiste en
 - Norma IEEE 1671.1: Test Description/*Descripción de la Prueba*
 - Norma IEEE 1671.2: Instrument Description/*Descripción del Instrumento*
 - Norma IEEE 1671.3: UUT Description/*Descripción de UUT*
 - Norma IEEE 1671.4: Test Configuration Description/*Descripción de la Configuración de la Prueba*
 - Norma IEEE 1671.5: Test Adaptor Description/*Descripción del Adaptador de la Prueba*
 - Norma IEEE 1671.6: Test Station Description/*Descripción de la Estación de Prueba*
 - Norma IEEE 1641: Signal and Test Definition/*Definición de Señal y Prueba*
 - Norma IEEE 1636,1: Test Results/*Resultados de la Prueba*
- La ISO/IEC/IEEE 29119-3:
- El Perfil de Prueba UML (UTP) por el OMG (Object Management Group) que especifica los conceptos de especificación de prueba para
 - Arquitectura de la Prueba
 - Datos de prueba
 - Comportamiento de la Prueba
 - Registro de Prueba
 - Gestión de Prueba

9.2 Documentos de la ISTQB

Referencia	Identificador
STQB-AL-TM	ISTQB Programa de Estudio de Nivel Avanzado, Jefe de Prueba, Versión 2012, disponible en [ISTQB-Web]
ISTQB-AL-TTA	ISTQB Programa de Estudio de Nivel Avanzado, Analista de Pruebas Técnicas, Versión 2012, disponible en [ISTQB-Web]
ISTQB-EL-CEP	ISTQB Extensión de Certificación de Nivel Avanzado, disponible en [ISTQB-Web]
ISTQB-EL-Módulos	ISTQB Descripción General de los Módulos de Nivel Avanzado, Versión 1.2, 23 de agosto de 2013, disponible en [ISTQB-Web]
ISTQB-EL-TM	ISTQB Nivel Avanzado – Programa de Estudio de Gestión de Prueba, Versión 2011, disponible en [ISTQB-Web]
ISTQB-FL	ISTQB Programa de Estudio de Nivel Básico, Versión 2011, disponible en [ISTQB-Web]
ISTQB-Glosario	STQB Glosario de Términos, Versión 2.4, 4 de julio de 2014, disponible en [ISTQB-Web]

9.3 Marcas Registradas

Las siguientes marcas registradas y marcas de servicio se utilizan en este documento:

ISTQB es una marca registrada de la International Software Testing Qualifications Board/*Junta Internacional de Calificaciones de Pruebas de Software*.

9.4 Libros

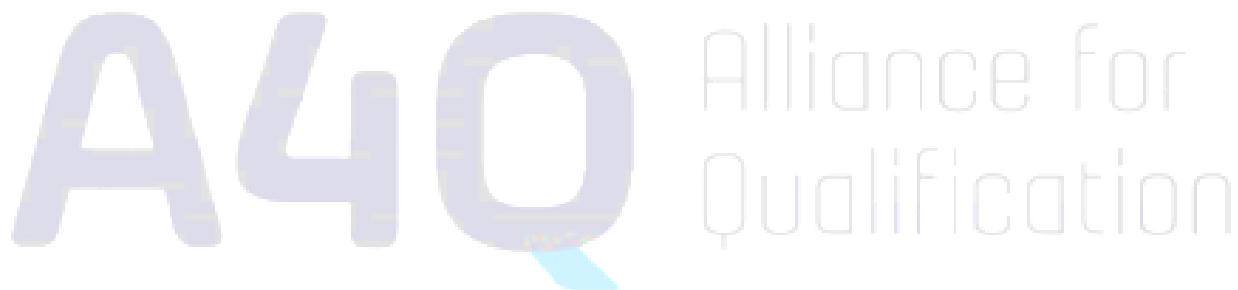
Referencia	Identificador del Libro
[Baker08]	Paul Baker, Zhen Ru Dai, Jens Grabowski and Ina Schieferdecker, "Model-Driven Testing: Using the UML Testing Profile", Springer 2008 edition, ISBN-10: 3540725628, ISBN-13: 978-3540725626)
[Dustin09]	Efriede Dustin, Thom Garrett, Bernie Gauf, "Implementing Automated Software Testing: how to save time and lower costs while raising quality", Addison-Wesley, 2009, ISBN 0-321-58051-6
[Dustin99]	Efriede Dustin, Jeff Rashka, John Paul, "Automated Software Testing: introduction, management, and performance", Addison-Wesley, 1999, ISBN-10: 0201432870, ISBN-13: 9780201432879
[Fewster&Graham12]	Mark Fewster, Dorothy Graham, "Experiences of Test Automation: Case Studies of Software Test Automation", Addison-Wesley, 2012
[Fewster&Graham99]	Mark Fewster, Dorothy Graham, "Software Test Automation: Effective use of test execution tools", ACM Press Books, 1999, ISBN-10: 0201331403, ISBN-13: 9780201331400
[McCaffrey06]	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

[Mosley02]	Daniel J. Mosley, Bruce A. Posey, "Just Enough Software Test Automation", Prentice Hall, 2002, ISBN-10: 0130084689, ISBN-13: 9780130084682
[Willcock11]	Colin Willcock, Thomas Deiß, Stephan Tobies and Stefan Keil, "An Introduction to TTCN-3" Wiley, 2nd edition 2011, ISBN-10: 0470663065, ISBN-13: 978-0470663066)

9.5 Referencias de la Web

Identificador	Referencia
---------------	------------

ISTQB-Web	Sitio Web de la Junta Internacional de Calificaciones de Pruebas de Software. Consulte este sitio Web para obtener el último Glosario y Programas de Estudio de la ISTQB. www.istqb.org
-----------	--



10 Aviso a los Proveedores de Capacitación

10.1 Tiempos de Capacitación

A cada capítulo del programa se le asigna un tiempo en minutos. El propósito de esto es brindar orientación sobre la proporción relativa de tiempo que se asignará a cada sección de un curso acreditado y dar un tiempo mínimo aproximado para la enseñanza de cada sección.

Los proveedores de capacitación pueden pasar más tiempo del indicado y los candidatos pueden dedicar más tiempo nuevamente a la lectura y a la investigación. Las asignaturas de un curso no tienen que seguir el mismo orden que el programa de estudio. No se requiere realizar el curso en un bloque de tiempo continuo.

La siguiente tabla proporciona una guía para los tiempos de enseñanza y ejercicio para cada capítulo (todos los tiempos se muestran en minutos).

Capítulo	Minutos
0. Introducción	0
1. Introducción y Objetivos para la Automatización de la Prueba	30
2. Preparación para la Automatización de la Prueba	165
3. La Arquitectura de Automatización de la Prueba Genérica	270
4. Riesgos de Implementación y Contingencias	150
5. Gestión de Información de Automatización de la Prueba y Métricas	165
6. Transición de las Pruebas Manuales a un Entorno Automatizado	120
7. Verificación de la SAP	120
8. Mejora Continua	150
Total:	1170

El tiempo total del curso en días, basado en un promedio de siete horas por día laborable, es de: 2 días, 5 horas, 30 minutos.

10.2 Ejercicios Prácticos en el Lugar de Trabajo

No hay ejercicios definidos que puedan realizarse en el lugar de trabajo.

10.3 Reglas para el Aprendizaje Virtual

Todas las partes de este programa se consideran apropiadas para su implementación como aprendizaje electrónico.

11 Índice

acreditar proveedores de formación, 7
acreditación de cursos, 8 acrónimos, 9
Pruebas de API, 11, 12, 13
densidad de defectos del código de automatización, 52, 53, 55, 56
resultados comerciales, 8
captura/reproducción, 22, 31, 32, 36
candidatos a la certificación, 7
Pruebas de CLI, 11, 12, 13
Paradigma cliente-servidor, 30
nivel de componentes, 17, 28
pruebas de confirmación, 60
pruebas guiadas por datos, 31
técnica de escritura de guiones guiada por datos, 34 pruebas guiadas por datos, 22
diseño para la capacidad de ser probado, 16, 20, 37, 72
controladores, 16, 21, 48
criterios de entrada, 8
esfuerzo de prueba manual equivalente, 52, 54
estimaciones, 30
Paradigma guiado por eventos, 30
examen, 8
Calificación de nivel experto, 8
métricas externas, 53
marco de trabajo, 13, 42, 57, 64, 71, 73
arquitectura de automatización de la prueba genérica, 22, 23
APg-A, 22, 23, 24, 63
pruebas de GUI, 13
informativa, 9
métricas internas, 53
intrusión, 16, 72
ISO 25000, 13
enfoque de pruebas guiadas por palabras clave, 31
técnica de escritura de guiones guiada por palabras clave, 34, 35
pruebas guiadas por palabras clave, 22, 76
palabras clave, 9, 23, 34, 35, 36, 47, 50, 68
Niveles K, 8
arquitectura en capas, 20 nivel de intrusión, 72 niveles de intrusión, 17
escritura de guiones lineales, 22, 32, 33, 36

registro, 12, 14, 23, 26, 37, 54, 57, 58
Mantenibilidad, 13

Alliance for
Qualification

pruebas basadas en modelos, 22, 36
Pruebas basadas en modelos, 31, 36
normativa, 9
Paradigma de igual a igual, 30
proyecto piloto, 19, 45, 63
enfoque guiado por procesos, 31, 35
escritura de guiones guiada por procesos, 22
gerencia de proyectos, 27
recuperar, 14, 76
pruebas de regresión, 53, 60, 61, 66, 67
gestión de información, 12, 14, 19, 24, 31, 37,
38, 52, 56, 57, 58, 63, 65, 68, 77, 78
evaluación de riesgos, 44
mitigación de riesgos, 44
escritura de guiones, 7, 21, 29, 32, 33, 34, 35,
36, 53, 56, 57, 68, 76
guión estructurado, 22 Enfoque de guión
estructurado, 31
stubs, 14, 16, 21
factores de éxito, 11, 13, 15
arquitectura del SSP, 30
configuraciones del SSP, 37
sistema sujeto a prueba, 12
capa de adaptación de la prueba, 22, 24, 27,
29
Capa de Adaptación de la Prueba, 24, 27
arquitectura de automatización de la prueba,
22
Arquitectura de Automatización de la Prueba
(AAP), 13
**marco de trabajo de automatización de la
prueba**, 11, 22, 23
proyecto de automatización de la prueba,
15, 25
solución de automatización de la prueba, 17,
22
estrategia de automatización de la prueba,
11
estrategia de automatización de la prueba
(TASt), 13
definición de la prueba, 34
capa de definición de la prueba, 22, 24, 26, 28
Capa de Definición de la Prueba, 24, 26
entorno de prueba, 14, 19, 20, 48, 50, 63, 64,
66, 71, 72, 78
capa de ejecución de la prueba, 22, 26, 28, 29
Capa de Ejecución de la Prueba, 24, 26
capa de generación de la prueba, 22, 24, 26,
28
Capa de Generación de la Prueba, 24, 26
anzuelo de prueba, 16
anzuelos de prueba, 17
registro de prueba, 52, 57
capacidad de ser probado, 20
productos de prueba, 11, 12, 14, 27, 30, 36, 50,
56, 67, 68,

76, 77
selección de
herramientas,
18**costo total
de la prueba**,
12 trazabilidad,
14, 37
traducir, 7
solución de problemas, 14, 59
esperas, 76

Alliance for
Qualification