

Probador Certificado

Programa de Estudio de Nivel Avanzado

Ingeniero de Automatización de Pruebas

Versión ES 001.08 basada en la Versión 2016

Spanish Software Testing
Qualifications Board

International Software
Testing Qualifications
Board



Nota sobre derechos de propiedad intelectual

El presente documento podrá ser copiado parcial o íntegramente siempre y cuando se cite la fuente.

Probador Certificado

Programa de Estudio - Nivel Avanzado

Ingeniero de Automatización de Pruebas



Copyright © International Software Testing Qualifications Board (en adelante denominado ISTQB®).

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



Historial de Revisiones

Versión	Fecha	Observaciones
Borrador inicial	13AGO2015	Borrador inicial.
Segundo borrador	05NOV2015	Mapeo y reubicación de Objetivos de Aprendizaje (OA).
Tercer borrador	17DIC2015	OA refinados.
Borrador Beta	11ENE2016	Borrador editado.
Beta	18MAR2016	Versión Beta.
Programa de Estudio 2016	21OCT2016	Lanzamiento en Asamblea General.
Traducción		
Versión	Fecha	Observaciones
ES.V1.00	12SEP2018	Versión de lanzamiento. Está prevista una próxima versión en la que se incluya el índice temático.

Índice general

Historial de Revisiones.....	3
Índice general.....	4
Agradecimientos	7
1.1 Versión Española.....	7
0. Introducción a este Programa de Estudio	8
0.1 Objetivo de este documento	8
0.2 Alcance de este documento.....	8
0.2.1 Dentro de alcance	8
0.2.2 Fuera de alcance	9
0.3 Probador Certificado de Nivel Avanzado - Ingeniero de Automatización de Pruebas	9
0.3.1 Expectativas	9
0.3.2 Requisitos de Acceso y Renovación	9
0.3.3 Nivel de conocimiento	10
0.3.4 Examen	10
0.3.5 Acreditación.....	10
0.4 Partes Normativas frente a Partes Informativas	10
0.5 Nivel de Detalle.....	11
0.6 Organización del Programa de Estudio	11
0.7 Términos, Definiciones y Acrónimos.....	12
1. Introducción y Objetivos para la Automatización de Pruebas - 30 minutos	13
1.1 Objetivo de la Automatización de la Prueba	14
1.2 Factores de Éxito en la Automatización de Pruebas	16
2. Preparación para la Automatización de la Prueba - 165 minutos	19
2.1 Factores del SSP que Influyen en la Automatización de Pruebas.....	20
2.2 Evaluación y Selección de Herramientas	21
2.3 Diseño para Capacidad de ser Probado y Automatización.....	24
3. La Arquitectura de Automatización de Pruebas Genérica - 270 minutos.....	27
3.1 Introducción a la AAPg	28
3.1.1 Descripción general de la AAPg.....	29
3.1.2 Capa de Generación de Pruebas	31
3.1.3 Capa de Definición de Pruebas	32
3.1.4 Capa de Ejecución de Pruebas	32
3.1.5 Capa de Adaptación de Pruebas.....	33
3.1.6 Gestión de la Configuración de una SAP	34
3.1.7 Gestión de Proyecto de una SAP	34

3.1.8 Soporte de la SAP para la Gestión de Pruebas	34
3.2 Diseño de la AAP	35
3.2.1 Introducción al Diseño de la AAP	35
3.2.2 Enfoques para la Automatización de Casos de Prueba	40
3.2.3 Consideraciones técnicas con respecto al SSP	47
3.2.4 Consideraciones para los Procesos de Desarrollo/Aseguramiento de la Calidad	48
3.3 Desarrollo de la SAP	49
3.3.1 Introducción al desarrollo de la SAP	49
3.3.2 Compatibilidad entre la SAP y el SSP	50
3.3.3 Sincronización entre la SAP y el SSP	51
3.3.4 Construcción de la reutilización en la SAP	55
3.3.5 Soporte para una Variedad de Sistemas Objetivo	55
4. Riesgos y Contingencias del Despliegue - 150 minutos	57
4.1 Selección del Enfoque de Automatización de Pruebas y Planificación del Despliegue/Puesta en Marcha	58
4.1.1 Proyecto Piloto	58
4.1.2 Despliegue	60
4.1.3 Despliegue de la SAP Dentro del Ciclo de Vida del Software	60
4.2 Evaluación del Riesgo y Estrategias de Mitigación	61
4.3 Mantenimiento de la Automatización de Pruebas	63
4.3.1 Tipos de Mantenimiento	63
4.3.2 Alcance y Enfoque	64
5. Información y Métricas de Automatización de Pruebas - 165 minutos	66
5.1 Selección de Métricas para la SAP	67
5.2 Implementación de la Medición	71
5.3 Registro de la SAP y del SSP	72
5.4 Información de la Automatización de Pruebas	74
6. Transición de Pruebas Manuales a un Entorno Automatizado - 120 minutos	75
6.1 Criterios para la Automatización	76
6.2 Identificar los Pasos Necesarios para Implementar la Automatización dentro de las Pruebas de Regresión	82
6.3 Factores que Hay que Tener en Cuenta al Implementar la Automatización dentro de Pruebas de Nuevas Prestaciones	85
6.4 Factores que Hay que Tener en Cuenta al Implementar la Automatización de las Pruebas de Confirmación	86
7. Verificación de la SAP - 120 minutos	87
7.1 Verificación de los Componentes del Entorno de Pruebas	88
7.2 Verificación del Juego de Pruebas Automatizadas	90

8. Mejora Continua - 150 minutos Referencias	93
8.1 Opciones para Mejorar la Automatización de Pruebas.....	94
8.2 Planificación de la Implementación de la Mejora de la Automatización de Pruebas	97
9. Referencias	99
9.1 Estándares.....	99
9.2 Documentos ISTQB.....	100
9.3 Marcas Registradas	100
9.4 Referencias bibliográficas.....	100
9.5 Referencias Web	101
10. Aviso a los Proveedores de Formación.....	102
10.1 Tiempos de Formación	102
10.2 Ejercicios Prácticos en el Área de Trabajo (Curso)	102
10.3 Normas para Aprendizaje Electrónico	102
11. Índice terminológico	103

Agradecimientos

Este documento fue elaborado por un equipo principal del Grupo de Trabajo de Nivel Avanzado del International Software Testing Qualifications Board.

El equipo principal agradece al equipo de revisión y a todos los Comités Nacionales y Regionales por sus sugerencias y aportaciones.

En el momento en que se completó el programa de estudios de nivel avanzado para este módulo, el Grupo de Trabajo de Nivel Avanzado - Automatización de Pruebas tenía la siguiente composición: 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 (Vicepresidenta del Grupo de Trabajo de Nivel Avanzado), Kateryna Nesmyelova, Mahantesh (Monty) Pattan, Andrew Pollner (Presidente de Automatización de Pruebas de Nivel Avanzado), Raluca Popescu, Ioana Prundaru, Riccardo Rosci, Ina Schieferdecker, Gil Shekel, Chris Van Bael.

El equipo principal autor de este plan de estudios está formado por las siguientes personas: 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 del ISTQB el 21 de octubre de 2016.

1.1 Versión Española

Las siguientes personas han participado en la traducción y revisión de esta versión del programa de estudio:

- Traducción:
 - Gustavo Márquez Sosa (SSTQB)
- Revisión:
 - Aurelio Gandarillas (MTP, España)

0. Introducción a este Programa de Estudio

0.1 Objetivo de este documento

Este programa de estudio constituye la base para la capacitación correspondiente a Probador Certificado de Nivel Avanzado en la especialidad de Ingeniero de Automatización de Pruebas. El ISTQB® ofrece el presente programa de estudio:

- a los Comités Nacionales, para que lo traduzcan a su idioma local y para acreditar a los proveedores de formación. Los Comités Nacionales podrán adaptar el programa de estudio a las necesidades específicas de su idioma y modificar las referencias para adaptarlas a sus publicaciones locales.
- a los Comités de Exámenes, para que puedan crear preguntas de examen en su idioma local que se adapten a los objetivos de aprendizaje de cada programa de estudio.
- a los proveedores de formación, para que elaboren los materiales didácticos y determinen los métodos de enseñanza apropiados.
- a los candidatos a la certificación, para preparar el examen (como parte de un curso de formación o de forma independiente).
- a la comunidad internacional de ingeniería de software y sistemas, para avanzar en la profesión de software y pruebas de sistemas, y como referencia para libros y artículos.

El ISTQB® podrá autorizar a otras entidades que utilicen este programa de estudio con otros fines, siempre y cuando soliciten y obtengan la correspondiente autorización previa por escrito

0.2 Alcance de este documento

0.2.1 Dentro de alcance

Este documento describe las tareas de un ingeniero de automatización de pruebas (IAP) en el diseño, desarrollo y mantenimiento de soluciones de automatización de pruebas. El contenido se concentra en los conceptos, métodos, herramientas y procesos para automatizar pruebas funcionales dinámicas y la relación de dichas pruebas con la gestión de pruebas, 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 del ciclo de vida del software (por ejemplo, ágiles, secuenciales, incrementales, iterativos), tipos de sistemas de software (por ejemplo, embebidos, distribuidos, móviles) y tipos de pruebas (pruebas funcionales y no funcionales).

0.2.2 Fuera de alcance

Los siguientes aspectos están fuera del alcance de este programa de estudio de Ingeniería de Automatización de Pruebas:

- Gestión de pruebas, creación automática de especificaciones de pruebas y generación automática de pruebas.
- Tareas del Gestor de Automatización de Pruebas (GAP) en la planificación, supervisión y ajuste del desarrollo y evolución de soluciones de automatización de pruebas.
- Aspectos específicos de la automatización de pruebas no funcionales (por ejemplo, rendimiento).
- Automatización del análisis estático (por ejemplo, análisis de vulnerabilidad) y herramientas de pruebas estáticas.
- Enseñanza de métodos de ingeniería de software y programación (por ejemplo, qué estándares usar y qué habilidades tener para realizar una solución de automatización de pruebas).
- Enseñanza de tecnologías de software (por ejemplo, qué técnicas de guion utilizar para implementar una solución de automatización de pruebas).

Selección de productos y servicios de pruebas de software (por ejemplo, qué productos y servicios utilizar para una solución de automatización de pruebas).

0.3 Probador Certificado de Nivel Avanzado - Ingeniero de Automatización de Pruebas

0.3.1 Expectativas

La cualificación de Nivel Avanzado está dirigida a personas que desean desarrollar los conocimientos y habilidades adquiridas en el Nivel Básico y desarrollar aún más su experiencia en una o más áreas específicas. Los módulos ofrecidos en el Especialista de Nivel Avanzado cubren una amplia gama de temas de la prueba.

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

El documento Advanced Level Modules Overview [ISTQB-AL-Modules] describe los resultados de negocio de este módulo.

0.3.2 Requisitos de Acceso y Renovación

Los criterios generales de acceso al Nivel Avanzado se describen en la sección Nivel Avanzado de la página web del ISTQB [ISTQB-Web].

Además de estos criterios generales de admisión, los candidatos deben poseer el certificado ISTQB Foundation Level [ISTQB-CTFL] para presentarse al examen de certificación de Ingeniero de Automatización de Pruebas de Nivel Avanzado.

0.3.3 Nivel de conocimiento

Los objetivos de aprendizaje para este plan de estudio se recogen al principio de cada capítulo para facilitar su identificación. Cada tema del programa de estudio se examinará de acuerdo con el objetivo de aprendizaje asignado.

Los niveles cognitivos asignados a los objetivos de aprendizaje (niveles-C o ("K-levels" en inglés) se describen en el sitio web del ISTQB [ISTQB-Web].

0.3.4 Examen

El examen para este Certificado de Nivel Avanzado se basará en este programa de estudio más el Programa 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 estos programas.

El formato del examen se describe en el sitio web del ISTQB [ISTQB-Web], sección Nivel Avanzado. En el sitio web del ISTQB también se puede encontrar información útil para quienes se presenten al examen.

0.3.5 Acreditación

Un Comité Miembro de ISTQB puede acreditar a los proveedores de capacitación cuyo material del curso siga este programa.

La sección Nivel Avanzado de la página web del ISTQB [ISTQB-Web] describe las normas específicas que se aplican a los proveedores de formación para la acreditación de cursos.

0.4 Partes Normativas frente a Partes Informativas

Las partes normativas del programa de estudios son objeto de examen. Estas partes son:

- Objetivos de aprendizaje.
- Palabras Clave.

El resto del programa de estudio es informativo y desarrolla los objetivos de aprendizaje.

0.5 Nivel de Detalle

El nivel de detalle de este plan de estudio permite una enseñanza y un examen coherentes a nivel internacional. Para lograr este objetivo, el plan de estudios consta de:

- Objetivos de aprendizaje para cada área de conocimiento, describiendo el resultado del aprendizaje cognitivo y el modo de pensar que debe lograrse (estos son normativos).
- Una lista de la información a enseñar, incluyendo una descripción de los conceptos clave a enseñar, fuentes tales como bibliografía o estándares aceptados, y referencias a fuentes adicionales si fuera necesario (estas son informativas).

El contenido del plan de estudio no es una descripción de toda el área de conocimiento de la ingeniería de automatización de pruebas; refleja el nivel de detalle que debe cubrirse en un curso de formación acreditado de Nivel Avanzado.

0.6 Organización del Programa de Estudio

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

3. La Arquitectura de Automatización de Pruebas Genérica - 270 minutos.

muestra que el Capítulo 3 debe tener un tiempo de 270 minutos para explicar el material del capítulo. Los objetivos de aprendizaje específicos se enumeran al principio de cada capítulo.

0.7 Términos, Definiciones y Acrónimos

Muchos de los términos utilizados en la literatura de software se utilizan de forma intercambiable. Las definiciones en este Programa de Estudio de Nivel Avanzado están disponibles en el Standard Glossary of Terms Used in Software Testing, publicado por el ISTQB [ISTQB-Glossary]¹.

Cada una de las palabras clave enumeradas al principio de cada capítulo en este programa de estudio de nivel avanzado se define en [ISTQB-Glossary].

En este documento se utilizan los siguientes acrónimos:

- AAP Arquitectura de Automatización de Pruebas (una instancia de una AAPg para definir la arquitectura de una SAP).
- AAPg Arquitectura de Automatización de Pruebas Genérica (aporta un modelo para las soluciones de automatización de pruebas).
- EPME Esfuerzo de Prueba Manual Equivalente.
- GAP Gestor de Automatización de Pruebas (responsable de la planificación y supervisión del desarrollo y evolución de un SAP).
- IAP Ingeniero de Automatización de Pruebas (la persona responsable del diseño de una AAP, incluyendo la implementación de la SAP resultante, su mantenimiento y evolución técnica).
- IGU Interfaz Gráfica de Usuario.
- ILC Interfaz de Línea de Comandos.
- IU Interfaz de Usuario
- MTAP Marco de Trabajo de Automatización de Pruebas (el entorno necesario para la automatización de pruebas, incluidos los arneses de pruebas y los artefactos, como las librerías de prueba).
- SAP Solución de Automatización de la Prueba (la realización/implementación de un AAP, incluyendo arneses de prueba y artefactos tales como librerías de prueba).
- SSP Sistema Sujeto a Prueba, véase también objeto de prueba.

¹ En la siguiente URL se puede consultar la versión española del glosario: <http://www.sstqb.es/recursos/descargas.html>

1. Introducción y Objetivos para la Automatización de Pruebas - 30 minutos

Palabras Clave

pruebas de Interfaz de Programación de Aplicaciones (antes pruebas de API), pruebas de Interfaz de Línea de Comandos (antes pruebas de ILC o pruebas de CLI), pruebas de Interfaz Gráfica de Usuario, Sistema Sujeto a Prueba, arquitectura de automatización de pruebas, marco de trabajo de automatización de pruebas, estrategia de automatización de pruebas, automatización de la prueba (antes automatización de pruebas), guion de prueba, producto de prueba (antes producto de soporte de prueba)

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

1.1 Objetivo de la Automatización de la Prueba

NA-IAP-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 Pruebas

NA-IAP-1.2.1 (K2) Identificar los factores técnicos de éxito 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 pruebas (que incluye la ejecución de pruebas automatizadas) es una o más de las siguientes tareas:

- Uso de herramientas software construidas de forma específica para controlar y establecer precondiciones de las pruebas.
- Ejecutar pruebas.
- Comparar los resultados reales con los resultados previstos.

Una buena práctica es separar el software utilizado para la prueba del propio sistema sujeto a prueba (SSP) para minimizar las interferencias. Hay excepciones, por ejemplo, sistemas embebidos en los que es necesario desplegar el software de prueba en el SSP.

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

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

Los productos de prueba son necesarios para las actividades de prueba que incluyen:

- Implementar casos de prueba automatizados.
- Monitorizar y controlar la ejecución de pruebas automatizadas.
- Interpretar, informar y registrar los resultados de pruebas automatizadas.

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

- Pruebas a través de las interfaces públicas a clases, módulos o librerías del SSP (pruebas de Interfaz de Programación de Aplicaciones).
- Pruebas a través de la interfaz de usuario del SSP (por ejemplo, pruebas de Interfaz Gráfica de Usuario o pruebas de Interfaz de Línea de Comandos).
- Pruebas a través de un servicio o protocolo.

Los objetivos de la automatización de pruebas incluyen:

- Mejorar la eficiencia de la prueba.
- Aportar una cobertura de funciones más amplia.
- Reducir el coste total de la prueba.
- Realizar pruebas que los probadores manuales no pueden.
- Acortar el período de ejecución de la prueba.
- Aumentar la frecuencia de la prueba/reducir el tiempo necesario para los ciclos de prueba.

Las ventajas de la automatización de pruebas incluyen:

- Se pueden realizar más pruebas por compilación ("build").
- La posibilidad de crear pruebas que no se pueden realizar manualmente (pruebas en tiempo real, remotas, en paralelo).
- Las pruebas pueden ser más complejas.
- Las pruebas se ejecutan más rápido.
- Las pruebas están menos sujetas a errores del operador.
- Uso más eficaz y eficiente de los recursos de pruebas.
- Información de retorno más rápida sobre la calidad del software.
- Mejora de la fiabilidad del sistema (por ejemplo, repetibilidad, consistencia).
- Mejora de la consistencia de la prueba.

Las desventajas de la automatización de pruebas incluyen:

- Existencia de costes adicionales
- Inversión inicial para el establecimiento de la SAP.
- Requiere tecnologías adicionales.
- El equipo necesita tener competencia² en desarrollo y automatización.
- Requiere un mantenimiento continuo de la SAP.
- Puede distraer la atención respecto a los objetivos de la prueba, por ejemplo, centrándose en la automatización de casos de prueba a expensas de la ejecución de las pruebas.
- Las pruebas pueden volverse más complejas.
- La automatización puede introducir errores adicionales.

Entre las limitaciones de la automatización de pruebas se incluyen:

- No todas las pruebas manuales se pueden automatizar.

² "competencia" es la traducción del término "skill".

- La automatización sólo puede comprobar resultados interpretables por la máquina.
- La automatización sólo puede comprobar los resultados reales que pueden ser verificados por un oráculo de prueba automatizado.
- No es un sustituto de las pruebas exploratorias.

1.2 Factores de Éxito en la Automatización de Pruebas

Los siguientes factores de éxito se aplican a los proyectos de automatización de pruebas que están en operación y, por lo tanto, el enfoque se centra en las influencias que impactan en el éxito a largo plazo de los proyectos. Aquí no se consideran los factores que influyen en el éxito de los proyectos de automatización de pruebas en la fase piloto.

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

- **Arquitectura de Automatización de Pruebas (AAP)**

La Arquitectura de Automatización de Pruebas (AAP) está muy alineada con la arquitectura de un producto software. Debe quedar claro qué requisitos funcionales y no funcionales deben ser soportados por la arquitectura. En general, estos serán los requisitos más importantes.

Con frecuencia, la AAP está diseñada para la mantenibilidad, el rendimiento y la capacidad de aprendizaje. (Ver ISO/IEC 25000:2014 para detalles de estas y otras características no funcionales.). Es útil involucrar a ingenieros de software que comprendan la arquitectura del SSP.

- **Capacidad de Ser Probado³ del SSP**

El SSP debe diseñarse para que cuente con capacidad de ser probado de forma que sea compatible con las pruebas automatizadas. En el caso de las pruebas de Interfaz Gráfica de Usuario, esto podría significar que el SSP debería desacoplar, en la medida de lo posible, la interacción de la Interfaz Gráfica de Usuario y los datos de la presentación de la interfaz gráfica. En el caso de las pruebas de Interfaz de Programación de Aplicaciones, esto podría significar que más clases, módulos o la interfaz de línea de comandos deban ser expuestos como públicos para que puedan ser probados.

En primer lugar, se deben seleccionar las partes con capacidad de ser probadas del SSP. En general, un factor de éxito clave en la automatización de pruebas se encuentra en la facilidad de implementar guiones de prueba automatizados. Con este objetivo en mente, y también para proporcionar una prueba de concepto exitosa, el Ingeniero de Automatización de Pruebas (IAP o TAE, por sus siglas en inglés) necesita identificar los módulos o componentes del SSP que se prueban fácilmente con la automatización y comenzar desde ese punto.

³ "capacidad de ser probado" es la traducción del término "testability". El término "testability" también se puede traducir como "testabilidad".

- **Estrategia de Automatización de Pruebas**

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

Puede que no sea posible aplicar la estrategia de automatización de pruebas de la misma manera a las partes antiguas y nuevas del SSP. En el momento de crear la estrategia de automatización, hay que tener en cuenta los costes, beneficios y riesgos de aplicarla a diferentes partes del código.

Se debe considerar la posibilidad de probar con casos de prueba automatizados tanto la interfaz de usuario como la Interfaz de Programación de Aplicaciones (IPA) (API por sus siglas en inglés) para comprobar la consistencia de los resultados.

- **Marco de Trabajo de Automatización de Pruebas (MTAP)**

Un marco de trabajo de automatización de pruebas (TAF, por sus siglas en inglés) que sea fácil de usar, bien documentado y mantenible, soporta un enfoque consistente para la automatización de pruebas.

Para poder establecer un marco de trabajo de automatización de pruebas (MTAP) (TAF, por sus siglas en inglés) fácil de usar y mantenible, se debe hacer lo siguiente:

- Implementar facilidades para la presentación de informes: Los informes de prueba deben proporcionar información (paso / fallo / error / no ejecutado / abortado, estadística, etc.) sobre la calidad del SSP. Los informes deben proporcionar la información necesaria para que los probadores, jefes de pruebas, desarrolladores, jefes de proyecto y otros implicados obtengan una visión general de la calidad.
- Facilitar la resolución de problemas: Además de la ejecución de la prueba y el registro, el MTAP tiene que proporcionar una forma fácil de solucionar problemas con las pruebas fallidas. La prueba puede fallar debido a:
 - fallos detectados en el SSP.
 - fallos detectados en la SAP.
 - problemas con las pruebas en sí mismas o con el entorno de prueba.
- Abordar el entorno de prueba de forma adecuada: Las herramientas de prueba dependen de la consistencia en el entorno de prueba. En pruebas automatizadas es necesario contar con un entorno de pruebas dedicado. Si no hay control del entorno de prueba y de los datos de prueba, la configuración de las pruebas puede no cumplir los requisitos para la ejecución de la prueba y es probable que produzcan resultados de ejecución falsos.
- Documentar los casos de prueba automatizados: Los objetivos de la automatización de pruebas deben ser claros, por ejemplo, qué partes de la aplicación deben probarse, en qué medida y qué atributos se deben probar (funcionales y no funcionales). Esto debe estar claramente descrito y documentado.
- Trazar la prueba automatizada: El MTAP soportará el trazado para que el ingeniero de automatización de pruebas pueda trazar pasos individuales a casos de prueba.

- Permitir un mantenimiento fácil: Preferiblemente, los casos de prueba automatizados deben ser fácilmente mantenidos para que el mantenimiento no consuma una parte significativa del esfuerzo de automatización de la prueba. Adicionalmente, el esfuerzo de mantenimiento debe ser proporcional a la magnitud de los cambios realizados en el SSP. Para ello, los casos deben ser fácilmente analizables, modificables y ampliables. Además, la reutilización de productos de prueba automatizados debe ser alta para minimizar el número de elementos que requieran cambios.
- Se deben mantener actualizadas las pruebas automatizadas: cuando los requisitos nuevos o modificados provoquen un fallo en las pruebas o en juegos de prueba completos, no se deben desactivar las pruebas que hayan fallado - se deben corregir.
- Planificar el despliegue: Se debe comprobar que los guiones de prueba se puedan desplegar, modificar y volver a desplegar con facilidad.
- Retirar pruebas según sea necesario: Se debe asegurar que los guiones de prueba automatizados se pueden retirar fácilmente si ya no son útiles o necesarios.
- Monitorizar y restaurar el SSP: En la práctica, para ejecutar de forma continua un caso de prueba o un conjunto de casos de prueba, el SSP debe ser monitorizado de forma continua. Si el SSP encuentra un error fatal (como un fallo del sistema), el MTAP debe tener la capacidad de recuperarse, omitir el caso actual⁴ y reanudar las pruebas con el siguiente caso.

El mantenimiento del código de automatización de pruebas puede ser complejo. No es extraño tener tanto código para las pruebas como código para el SSP. Por esta razón es de suma importancia que el código de prueba sea mantenible. Esto se debe a que se utilizan diferentes herramientas de prueba, diferentes tipos de verificación y diferentes artefactos de productos de prueba que deben mantenerse (como datos de entrada de prueba, oráculos de prueba, informes de prueba).

Con estas consideraciones de mantenimiento en mente, además de los puntos importantes que se deben realizar, hay algunos que no se deben realizar, como se indica a continuación:

- No cree código que sea sensible a la interfaz (es decir, que se vería afectado por cambios en la interfaz gráfica o en partes no esenciales de la Interfaz de Programación de Aplicaciones (o API por sus siglas en inglés).
- No cree una automatización de pruebas que sea sensible a los cambios de datos o que tenga una alta dependencia de valores de datos particulares (por ejemplo, la entrada de pruebas depende de otras salidas de pruebas).
- No cree un entorno de automatización que sea sensible al contexto (por ejemplo, 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.

Cuantos más factores de éxito se cumplan, mayor será la probabilidad de éxito del proyecto de automatización de pruebas. No todos los factores son necesarios, y en la práctica rara vez se cumplen todos los factores. Antes de iniciar el proyecto de automatización de pruebas, es importante analizar las posibilidades de éxito del proyecto considerando los factores existentes 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 en su sitio, es importante investigar qué artículos faltan o aún necesitan trabajo.

⁴ “actual” es la traducción del término “current”. El término “current” también podría traducir como “en curso”.

2. Preparación para la Automatización de la Prueba - 165 minutos

Palabras Clave

capacidad de ser probado, controlador, nivel de intrusión, stub, herramienta de ejecución de pruebas, anzuelo de prueba, jefe de automatización de la prueba

Objetivos de aprendizaje de Preparación para la Automatización de la Prueba

2.1 Factores del SSP que Influyen en la Automatización de Pruebas

NA-IAP-2.1.1 (K4) Analizar un sistema sujeto a prueba para determinar la solución de automatización adecuada.

2.2 Evaluación y Selección de Herramientas

NA-IAP-2.2.1 (K4) Analizar herramientas de automatización de pruebas para un proyecto determinado e informar sobre los hallazgos y recomendaciones técnicas.

2.3 Diseño para Capacidad de ser Probado y Automatización

NA-IAP-2.3.1 (K2) Comprender los métodos de "diseño para la capacidad de ser probado" y "diseño para la automatización de pruebas" aplicables al SSP.

2.1 Factores del SSP que Influyen en la Automatización de Pruebas

Al evaluar el contexto del SSP y su entorno, es necesario identificar los factores que influyen en la automatización de las pruebas para determinar una solución adecuada. Estos pueden incluir los siguientes aspectos:

- **Interfaces del SSP**

Los casos de prueba automatizados invocan acciones en el SSP. Para ello, el SSP debe proporcionar interfaces a través de las cuales se pueda controlar el mismo SSP. Esto se puede hacer a través de los controles de la interfaz de usuario, pero también a través de interfaces de software de nivel inferior. Además, algunos casos de prueba pueden ser capaces de interconectarse a nivel de comunicación (por ejemplo, utilizando interfaces TCP/IP, USB o de mensajería propietaria).

La descomposición del SSP permite que la automatización de la prueba interactúe con el SSP en diferentes niveles de prueba. Es posible automatizar las pruebas a un nivel específico (por ejemplo, a nivel de componente y de sistema), pero sólo cuando el SSP lo soporte adecuadamente. Por ejemplo, a nivel de componente, puede que no exista una interfaz de usuario que pueda utilizarse para las pruebas, por lo que es necesario disponer de interfaces software diferentes, posiblemente personalizadas (también llamadas anzuelos de prueba).

- **Software de terceros**

A menudo, el SSP no sólo 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 pruebas diferente, como el uso de una Interfaz de Programación de Aplicaciones (o API por sus siglas en inglés).

- **Niveles de intrusión**

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

El problema con los niveles de intrusión más altos es el riesgo de falsas alarmas. La SAP puede presentar fallos que pueden deberse al nivel de intrusión impuesto por las pruebas, pero es poco probable que ocurran cuando el sistema de software se utiliza en un entorno real. La prueba con un alto nivel de intrusión suele ser una solución más sencilla para el enfoque de automatización de pruebas.

- **Diferentes arquitecturas del SSP**

Diferentes arquitecturas del SSP pueden requerir diferentes soluciones de automatización de pruebas. Se necesita un enfoque diferente para un SSP programado en C++ utilizando tecnología COM que para un SSP programado en Python. Puede ser posible que estas diferentes arquitecturas sean tratadas por la misma estrategia de automatización de pruebas, pero eso requiere una estrategia híbrida con la capacidad de soportarlas.

- **Tamaño y complejidad del SSP**

Considerar el tamaño y la complejidad del SSP actual y los planes para un desarrollo futuro. Para un SSP pequeño y simple, un enfoque de automatización de pruebas complejo y ultra flexible puede no estar justificado. 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 conveniente comenzar con algo reducido y sencillo, incluso para un SSP complejo, pero éste debería ser un enfoque temporal (véase el capítulo 3 para más detalles).

Muchos de los factores descritos aquí son conocidos (por ejemplo, tamaño y complejidad, interfaces de software disponibles) cuando el SSP se encuentra disponible, pero la mayoría de las veces el desarrollo de la automatización de pruebas debe comenzar antes de que el SSP esté disponible. Cuando esto sucede varias cosas necesitan ser estimadas o el IAP (Ingeniero de Automatización de Pruebas) puede especificar las interfaces software que se necesitan. (Ver Sección 2.3 para más detalles).

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

- Cuando se conocen los requisitos (funcionales o no funcionales), se pueden seleccionar candidatos para la automatización de entre esos requisitos junto con la identificación de los medios para probarlos. La planificación para la automatización puede comenzar para esos candidatos, incluyendo la identificación de los requisitos para la automatización y la determinación de la estrategia de automatización de pruebas.
- Durante el desarrollo de la arquitectura y el diseño técnico, se puede emprender el diseño de interfaces software para soportar las pruebas.

2.2 Evaluación y Selección de Herramientas

La responsabilidad principal del proceso de selección y evaluación de herramientas recae en el Gestor de Automatización de Pruebas (GAP). Sin embargo, el IAP participará en el suministro de información al GAP y en la realización de muchas de las actividades de evaluación y selección. El concepto del proceso de evaluación y selección de herramientas se introdujo en el Nivel Básico y se describen más detalles de este proceso en el programa de estudios de Nivel Avanzado - Jefe de Pruebas [o por sus siglas en inglés, ISTQB-CTAL-TM].

El IAP estará involucrado durante todo el proceso de evaluación y selección de una herramienta, pero tendrá contribuciones particulares que hacer en las siguientes actividades:

- Evaluar la madurez de la organización e identificar oportunidades para el soporte de herramientas de prueba.
- Evaluar los objetivos apropiados para el soporte de la herramienta de prueba.
- Identificar y recopilar información sobre herramientas potencialmente adecuadas.
- Analizar la información de la herramienta con respecto a los objetivos y las restricciones del proyecto.
- Estimar la relación coste-beneficio en función de un caso de negocio robusto.

- Hacer una recomendación sobre la herramienta apropiada.
- Identificar la compatibilidad de la herramienta con los componentes del SSP.

Con frecuencia, las herramientas de automatización de pruebas funcionales no pueden satisfacer todas las expectativas o las situaciones a las que se enfrenta un proyecto de automatización. El siguiente es un conjunto de ejemplos de este tipo de asuntos (pero definitivamente no es una lista completa):

Hallazgo	Ejemplos	Posibles Soluciones
La interfaz de la herramienta no funciona con otras herramientas ya en funcionamiento.	<ul style="list-style-type: none"> • La herramienta de gestión de pruebas se ha actualizado y la interfaz de conexión se ha modificado. • La información del soporte de preventa era incorrecta y no todos los datos se pueden transferir a la herramienta de informes. 	<ul style="list-style-type: none"> • Preste atención a las notas de la versión antes de cualquier actualización, y para grandes migraciones pruebe antes de migrar a producción. • Intente obtener una demostración in situ de la herramienta que utiliza el SSP real. • Busque asistencia del proveedor y/o foros de la comunidad de usuarios.
Algunas dependencias del SSP se cambian por otras no soportadas por la herramienta de prueba.	<ul style="list-style-type: none"> • El departamento de desarrollo ha actualizado a la última versión de Java. 	<ul style="list-style-type: none"> • Sincronizar las actualizaciones para el entorno de desarrollo/prueba y la herramienta de automatización de pruebas.
Un objeto en la Interfaz Gráfica de Usuario no pudo ser capturado.	<ul style="list-style-type: none"> • El objeto es visible pero la herramienta de automatización de pruebas no puede interactuar con él. 	<ul style="list-style-type: none"> • Intentar utilizar sólo tecnologías bien conocidas u objetos en desarrollo. • Realizar un proyecto piloto antes de comprar una herramienta de automatización de pruebas. • Hacer que los desarrolladores definan estándares para los objetos.
La herramienta parece muy complicada.	<ul style="list-style-type: none"> • La herramienta tiene un enorme conjunto de prestaciones⁵, pero sólo se utilizará una parte de ellas. 	<ul style="list-style-type: none"> • Intentar encontrar una manera de limitar el conjunto de prestaciones eliminando las prestaciones no deseadas de la barra de herramientas. • Seleccionar una licencia que satisfaga sus necesidades. • Tratar de encontrar herramientas alternativas que se centren más en la funcionalidad requerida.

⁵ "prestación" es la traducción del término "feature".

Hallazgo	Ejemplos	Posibles Soluciones
Conflicto con otros sistemas.	<ul style="list-style-type: none"> Después de la instalación de otro software, la herramienta de automatización de pruebas deja de funcionar o viceversa. 	<ul style="list-style-type: none"> Leer las notas de la versión o los requisitos técnicos antes de la instalación. Obtener confirmación del proveedor de que no habrá impacto sobre otras herramientas. Preguntar en foros de la comunidad de usuarios.
Impacto en el SSP.	<ul style="list-style-type: none"> Durante/después del uso de la herramienta de automatización de pruebas, el SSP reacciona de forma diferente (por ejemplo, un tiempo de respuesta más largo). 	<ul style="list-style-type: none"> Utilizar una herramienta que no necesite modificar el SSP (por ejemplo, instalación de bibliotecas, etc.).
Acceso al código.	<ul style="list-style-type: none"> La herramienta de automatización de pruebas modificará partes del código fuente. 	<ul style="list-style-type: none"> Utilizar una herramienta que no necesite modificar el código fuente (por ejemplo, instalación de bibliotecas, etc.).
Recursos limitados (principalmente en entornos embebidos).	<ul style="list-style-type: none"> El entorno de prueba tiene recursos libres limitados o se queda sin recursos (por ejemplo, memoria). 	<ul style="list-style-type: none"> Leer las notas de la entrega y discutir el entorno con el proveedor de la herramienta para obtener la confirmación de que esto no dará lugar a problemas. Preguntar en los foros de la comunidad de usuarios.
Actualizaciones.	<ul style="list-style-type: none"> La actualización no migrará todos los datos o corromperá los guiones, datos o configuraciones de las pruebas automatizadas existentes. La actualización necesita un entorno diferente (mejor). 	<ul style="list-style-type: none"> Probar la actualización en el entorno de prueba y obtener la confirmación del proveedor de que la migración funcionará. Leer los prerequisites de la actualización y decidir si la actualización merece el esfuerzo. Busque apoyo en los foros de la comunidad de usuarios.
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 conceda acceso.

Hallazgo	Ejemplos	Posibles Soluciones
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 por consiguiente minimizando el coste de usar múltiples herramientas.

2.3 Diseño para Capacidad de ser Probado y Automatización

La capacidad de ser probado del SSP (disponibilidad de interfaces software que soporten pruebas, por ejemplo, para permitir 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 puede ser realizado por el arquitecto de software (ya que la capacidad de ser probado es sólo uno de los requisitos no funcionales del sistema), pero a menudo es realizado por, o con la participación de, un IAP.

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

- **Observabilidad:** El SSP necesita proporcionar interfaces que proporcionen información sobre el sistema. Los casos de prueba pueden utilizar estas interfaces para verificar, por ejemplo, si el comportamiento esperado es igual al comportamiento real.
- **Control(abilidad):** El SSP necesita proporcionar interfaces que se puedan utilizar para realizar acciones sobre el SSP. Estos pueden ser elementos de interfaz de usuario, llamadas de función, elementos de comunicación (por ejemplo, protocolo TCP/IP o USB), señales electrónicas (para conmutadores físicos), etc.
- **Arquitectura claramente definida:** La tercera parte importante del diseño para la capacidad de ser probado es una arquitectura que proporciona interfaces claras y comprensibles que proporcionan control y visibilidad sobre todos los niveles de prueba.

El IAP evalúa las formas en que se puede probar el SSP, incluyendo las pruebas automatizadas, de una forma efectiva (probando las áreas correctas y encontrando defectos críticos) y eficiente (sin hacer demasiado esfuerzo). Siempre que se necesiten interfaces software específicas, éstas deben ser especificadas por el IAP e implementadas por el desarrollador. Es importante definir la capacidad de ser probado y, si fuera necesario, interfaces software adicionales al principio del proyecto, para que el trabajo de desarrollo pueda ser planificado y presupuestado.

Algunos ejemplos de interfaces software que soportan pruebas incluyen:

- Las potentes funciones que ofrecen las hojas de cálculo modernas en lo que respecta a guiones.
- La aplicación de stubs o sustitutos ("mocks") para simular software y/o hardware (por ejemplo, transacciones financieras electrónicas, servicio de software, servidor dedicado, placa electrónica, parte mecánica) que aún no están disponibles o son demasiado caros para comprar, permite probar el software en ausencia de esa interfaz específica.
- Se pueden utilizar interfaces software (o stubs y controladores) para probar condiciones de error. Considere un dispositivo con una unidad de disco duro interna (HDD). El software que controla este disco duro (llamado controlador) debe probarse para detectar fallos o desgaste del disco duro. Hacer esto esperando a que un disco duro falle no es muy eficiente (o fiable). La

implementación de interfaces software que simulan discos duros defectuosos o lentos puede verificar que el software del controlador funciona correctamente (por ejemplo, proporciona un mensaje de error, intentos repetidos).

- Cuando aún no se dispone de una interfaz de usuario, se pueden utilizar interfaces software alternativas para probar un SSP (lo que a menudo se considera un mejor enfoque). El software embebido en los sistemas técnicos, a menudo, necesita monitorizar la temperatura del dispositivo y activar una función de enfriamiento para que se inicie cuando la temperatura supere un cierto nivel. Esto podría ser probado sin el hardware utilizando una interfaz software para especificar la temperatura.
- Las pruebas de transición de estado se utilizan para evaluar el comportamiento de estado del SSP. Una forma de comprobar si el SSP está en el estado correcto es consultarlo a través de una interfaz software personalizada diseñada con este fin (aunque esto también incluye un riesgo, véase el nivel de intrusión en la sección 2.1).

El diseño para la automatización debería tener en cuenta que:

- Se debe establecer la compatibilidad con las herramientas de prueba existentes desde el principio.
- El problema de la compatibilidad de las herramientas de prueba es crítico en el sentido de que puede afectar la capacidad de automatizar pruebas de funcionalidad importante (por ejemplo, la incompatibilidad con un control de rejilla⁶ impide que todas las pruebas utilicen ese control).
- Las soluciones pueden requerir el desarrollo de código de programa y llamadas a Interfaces de Programación de Aplicaciones.

El diseño para la capacidad de ser probado es de suma importancia para un buen enfoque de automatización de pruebas, y también puede beneficiar la ejecución de pruebas manuales.

⁶ “control de rejilla” es la traducción del término “grid control”.

3. La Arquitectura de Automatización de Pruebas Genérica - 270 minutos

Palabras Clave

arquitectura de automatización de pruebas genérica, arquitectura de automatización de pruebas, capa de adaptación de la prueba, capa de definición de la prueba, capa de ejecución de la prueba, capa de generación de pruebas, captura/reproducción, guion estructurado, guion guiado por proceso, guiones lineales, marco de trabajo de automatización de pruebas, pruebas basadas en modelos, pruebas guiadas por datos, pruebas guiadas por palabra clave, solución de automatización de la prueba

Objetivos de Aprendizaje para la Arquitectura de Automatización de Pruebas Genérica

3.1 Introducción a la AAPg

NA-IAP-3.1.1 (K2) Explicar la estructura de la AAPg.

3.2 Diseño de la AAP

NA-IAP-3.2.1 (K4) Diseñar la AAP apropiada para un proyecto dado.

NA-IAP-3.2.2 (K2) Explicar el papel que desempeñan las capas dentro de una AAP.

NA-IAP-3.2.3 (K2) Comprender las consideraciones de diseño para una AAP.

NA-IAP-3.2.4 (K4) Analizar los factores de implementación, uso y requisitos de mantenimiento para una SAP dada.

3.3 Desarrollo de una SAP

NA-IAP-3.3.1 (K3) Aplicar componentes de la AAP genérica (AAPg) para construir una AAP específica.

NA-IAP-3.3.2 (K2) Explicar los factores que se deben tener en cuenta al identificar la capacidad de reutilización de los componentes.

3.1 Introducción a la AAPg

Un ingeniero de automatización de pruebas (IAP) tiene el rol de diseñar, desarrollar, implementar y mantener soluciones de automatización de pruebas (SAP's). A medida que se desarrolla cada solución, es necesario realizar tareas similares, responder a preguntas similares y abordar y priorizar temas similares. Estos conceptos, pasos y enfoques recurrentes en la automatización de pruebas se convierten en la base de la arquitectura genérica de automatización de pruebas, llamada AAPg por su acrónimo.

La AAPg presenta las capas, los componentes y las interfaces de una AAPg, que entonces se redefinen en la AAP concreta para un SAP en particular. Esto permite un enfoque estructurado y modular para la construcción de una solución de automatización de pruebas al:

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

Una SAP consiste tanto en el entorno de prueba (y sus artefactos) como en 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 pruebas (MTAP) para realizar un SAP. Proporciona soporte para la realización del entorno de pruebas y proporciona herramientas, arneses de pruebas o librerías de soporte.

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

- Responsabilidad única: Cada componente de la SAP debe tener una sola responsabilidad, y esa responsabilidad debe estar totalmente encapsulada en el componente. En otras palabras, cada componente de una SAP debe estar a cargo de exactamente una cosa, por ejemplo, la generación de palabras clave o datos, la creación de escenarios de prueba, la ejecución de casos de prueba, el registro de resultados, la generación de informes de ejecución.
- Extensión (véase, por ejemplo, el principio abierto/cerrado de Bertrand Meyer): Cada componente de la SAP debe estar abierta para su extensión, pero cerrada para su modificación. Este principio significa que debería ser posible modificar o enriquecer el comportamiento de los componentes sin romper su compatibilidad con funcionalidad de versiones anteriores.
- Sustitución (véase, por ejemplo, el principio de sustitución por 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 que lo sustituyan, pero el comportamiento exhibido debe ser el mismo.

- Segregación de componentes (véase, por ejemplo, el principio de segregación de interfaces de R.C. Martin): Es mejor tener componentes más específicos que un componente general y polivalente. Esto facilita la sustitución y el mantenimiento al eliminar dependencias innecesarias.
- Inversión de dependencia: Los componentes de una SAP deben depender de abstracciones más que de detalles de bajo nivel. En otras palabras, los componentes no deben depender de escenarios de prueba automatizados específicos.

En general, una SAP basado en la AAPg será implementada por un conjunto de herramientas, sus extensiones y/o componentes. Es importante señalar que la AAPg es neutral con respecto al proveedor: no predefine ningún método, tecnología o herramienta concreta para la realización de una SAP. La AAPg puede ser implementada por cualquier enfoque de ingeniería de software, por ejemplo, estructurado, orientado a objetos, orientado a servicios, orientado a modelos, así como por cualquier tecnología y herramienta software. De hecho, una SAP, a menudo, se implementa utilizando herramientas disponibles en el mercado, pero normalmente necesitará adiciones y/o adaptaciones adicionales específicas del SSP.

Otras directrices y modelos de referencia relacionados con las SAP's son los estándares de ingeniería de software para el Modelo de Ciclo de Vida de Desarrollo de Software (SDLC por sus siglas en inglés) seleccionado, tecnologías de programación, estándares de formato⁷, etc. No está dentro del alcance de este programa de estudio enseñar ingeniería de software en general, sin embargo, se espera que un IAP tenga competencias, experiencia y pericia en ingeniería de software.

Además, un IAP necesita estar al tanto de los estándares de codificación y documentación de la industria y de las mejores prácticas para hacer uso de ellos mientras desarrolla una SAP. Estas prácticas pueden aumentar la mantenibilidad, fiabilidad y seguridad de la SAP. Tales estándares son, en general, específicos de un dominio. Entre los estándares más conocidos se encuentran:

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

3.1.1 Descripción general de la AAPg

La AAPg está estructurada en capas horizontales para lo siguiente:

- Generación de pruebas.
- Definición de pruebas.
- Ejecución de pruebas.
- Adaptación de pruebas.

⁷ "estándar de formato" es la traducción del término "formatting standard".

La AAPg (ver Figura 1: La Arquitectura de Automatización de Pruebas Genérica) comprende lo siguiente:

- La Capa de Generación de Pruebas es la capa que soporta el diseño manual o automatizado de casos de prueba. Proporciona los medios para diseñar casos de prueba.
- La Capa de Definición de Pruebas es la capa que soporta la definición e implementación de juegos de prueba y/o casos de prueba. Separa la definición de la prueba de la SSP y/o de las tecnologías y herramientas del sistema de prueba. Contiene medios para definir pruebas de alto y bajo nivel, que son tratadas en los datos de prueba, casos de prueba, procedimientos de prueba y componentes de librerías de prueba o combinaciones de los mismos.
- La Capa de Ejecución de Pruebas es la capa que soporta la ejecución de casos de prueba y el registro de pruebas. Proporciona una herramienta de ejecución de pruebas para ejecutar las pruebas seleccionadas de forma automática y un componente de registro y generación de informes.
- La Capa de Adaptación de Pruebas es la capa 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 Interfaces de Programación de Aplicaciones (API por sus siglas en inglés), protocolos, servicios y otros.
- También dispone de interfaces para la gestión de proyectos, la gestión de configuración y la gestión de pruebas en relación con la automatización de pruebas. Por ejemplo, la interfaz entre la gestión de pruebas y la capa de adaptación de pruebas se encarga de la selección y configuración de los adaptadores adecuados en relación con la configuración de pruebas elegida.

Las interfaces entre las capas de la AAPg y sus componentes son, en general, específicas y, por lo tanto, no se elaboran más aquí.

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

- Si se va a automatizar la ejecución de la prueba, es necesario utilizar las capas de ejecución de pruebas y de adaptación de pruebas. No es necesario separarlas y se podrían realizar juntas, por ejemplo, en marcos de pruebas unitarias.
- La capa de definición de pruebas es necesaria si se va a automatizar la definición de pruebas.
- La capa de generación de pruebas es necesaria si se va a automatizar la generación de pruebas.

La mayoría de las veces, se comenzaría con la implementación de un SAP de abajo hacia arriba, pero otros enfoques como la generación automatizada de pruebas para pruebas manuales también pueden ser útiles. En general, se aconseja implementar la SAP en pasos incrementales (por ejemplo, en sprints) con el fin de utilizar el SAP tan pronto como sea posible y demostrar el valor añadido del SAP. Además, se recomienda realizar pruebas de concepto como parte del proyecto de automatización de pruebas.

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

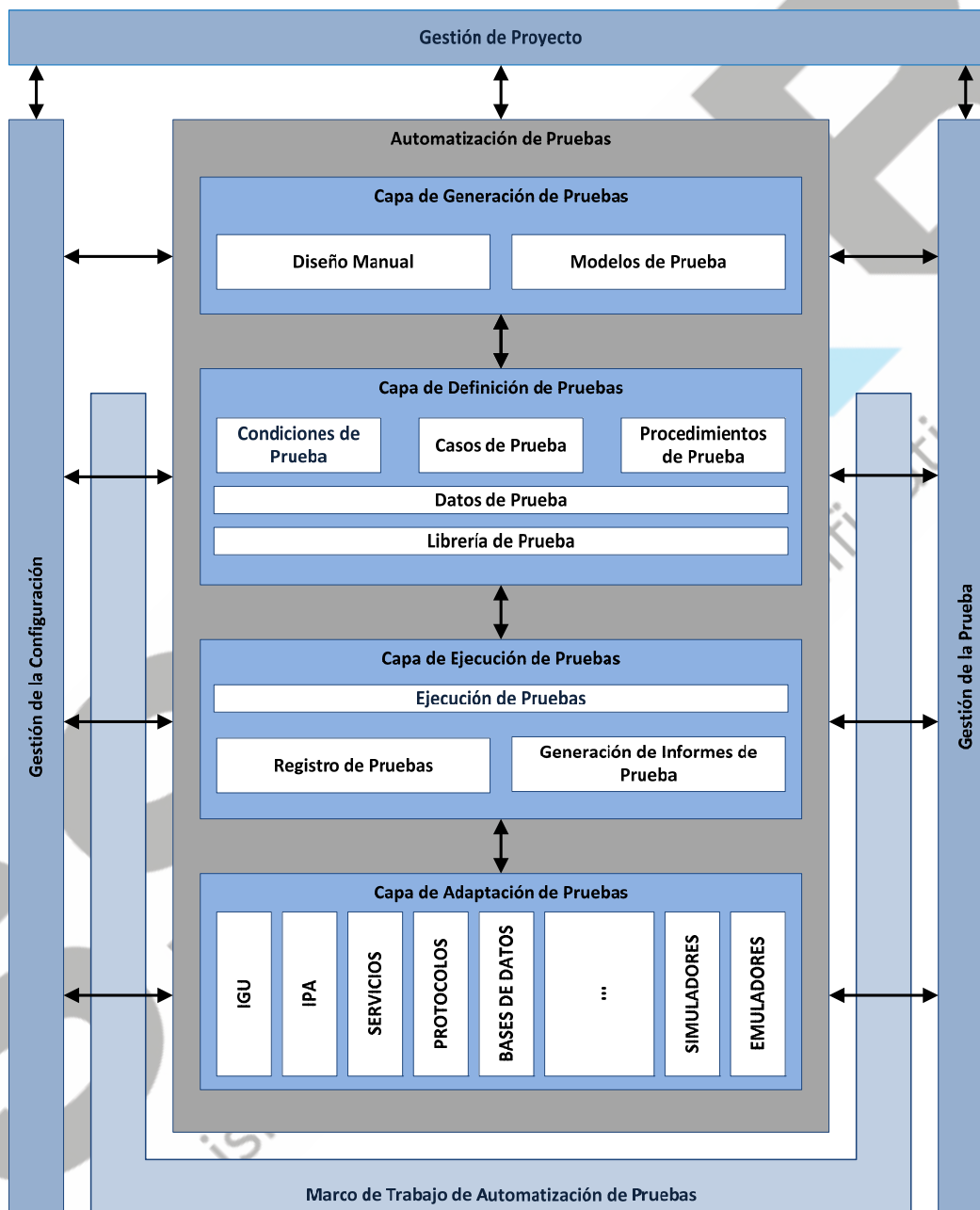


Figura 1: La Arquitectura de Automatización de Pruebas Genérica

3.1.2 Capa de Generación de Pruebas

La capa de generación de pruebas consiste en el soporte de herramientas para lo siguiente:

- Diseño manual de casos de prueba.

- Desarrollo, captura o derivación de 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 las estructuras del juego de pruebas.
- Relacionar los casos de prueba con los objetivos de la prueba o los requisitos del SSP.
- Documentar el diseño de la prueba.

Para la generación de pruebas automatizadas también se pueden incluir las siguientes capacidades:

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

3.1.3 Capa de Definición de Pruebas

La capa de definición de pruebas consiste en el soporte de herramientas para lo siguiente:

- Especificación de casos de prueba (a alto y/o bajo nivel).
- Definición de datos de prueba para casos de prueba de bajo nivel.
- Especificación de procedimientos de prueba para un caso de prueba o un conjunto de casos de prueba.
- Definición de guiones de prueba para la ejecución de los casos de prueba.
- Proporcionar acceso a las librerías de prueba según sea necesario (por ejemplo, en enfoques basados en palabras clave).

Los componentes de esta capa se utilizan para:

- Partición/restricción, parametrización o instanciación de datos de prueba.
- Especificar secuencias de prueba o comportamientos de prueba completos (incluyendo sentencias y expresiones de control), para parametrizarlos y/o agruparlos.
- Documentar los datos de prueba, casos de prueba y/o procedimientos de prueba.

3.1.4 Capa de Ejecución de Pruebas

La capa de ejecución de pruebas consiste en soporte de herramientas para lo siguiente:

- Ejecución automática de casos de prueba.
- Registro de las ejecuciones de casos de prueba.
- Generación de información de los resultados de la prueba.

La capa de ejecución de pruebas puede consistir en componentes que proporcionen las siguientes capacidades:

- Preparar y desmantelar⁸ el SSP para la ejecución de la prueba.
- Preparar y desmantelar los juegos de prueba (es decir, el conjunto de casos de prueba, incluidos los datos de prueba).
- Configurar y parametrizar la configuración de la prueba.
- Interpretar tanto datos de prueba como casos de prueba y transformarlos en guiones ejecutables.
- Instrumentalizar el sistema de prueba y/o el SSP para el registro (filtrado) de la ejecución de la prueba y/o para la inyección de defectos.
- Analizar las respuestas del SSP durante la ejecución de la prueba para orientar la ejecución de pruebas posteriores.
- Validar las respuestas del SSP (comparación de los resultados esperados y reales) para los resultados de la ejecución de casos de prueba automatizados.
- Controlar la ejecución temporal de las pruebas automatizadas.

3.1.5 Capa de Adaptación de Pruebas

La capa de adaptación de pruebas consiste en el 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.

⁸ "desmantelar" es la traducción del término "tear down".

La capa de adaptación de pruebas proporciona la siguiente funcionalidad:

- Intermediación entre las definiciones de las pruebas tecnológicamente neutras y los requisitos tecnológicos específicos del SSP y los dispositivos de prueba.
- Aplicación de diferentes adaptadores específicos de la tecnología para interactuar con el SSP.
- Distribución de la ejecución de la prueba a través de múltiples dispositivos de prueba/interfaces de prueba o ejecución pruebas de forma local.

3.1.6 Gestión de la Configuración de una SAP

Normalmente, una SAP se desarrolla en varias iteraciones/versiones y necesita ser compatible con las iteraciones/versiones del SSP. La gestión de la configuración de una SAP puede necesitar que se incluya:

- Modelos de prueba.
- Definiciones/especificaciones de pruebas, incluyendo datos de prueba, casos de prueba y librerías.
- Guiones de prueba.
- Motores de ejecución de pruebas así como herramientas y componentes suplementarios.
- Adaptadores de prueba para el SSP.
- Simuladores y emuladores para el entorno del SSP.
- Resultados e informes de la prueba.

Estos elementos constituyen el producto de prueba y deben estar en la versión correcta para que correspondan con la versión del SSP. En algunas situaciones puede ser necesario volver a las versiones anteriores de la SAP, por ejemplo, en caso de que sea necesario reproducir problemas de campo 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 pruebas es un proyecto de software, requiere la misma gestión de proyectos que cualquier otro proyecto de software. Un IAP necesita realizar tareas en todas las fases del Modelo de Ciclo de Vida de Desarrollo de la metodología establecida (SDLC por sus siglas en inglés) al desarrollar la SAP. Además, un IAP necesita entender que el entorno de desarrollo de una SAP debe ser diseñado de tal manera que la información sobre el estado (métricas) pueda ser extraída fácil o automáticamente informada a la dirección del proyecto de la SAP.

3.1.8 Soporte de la SAP para la Gestión de Pruebas

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

3.2 Diseño de la AAP

3.2.1 Introducción al Diseño de la AAP

Hay un número de actividades fundamentales requeridas para diseñar una AAP, las cuales pueden ser ordenadas de acuerdo a las necesidades del proyecto u organización de la automatización de pruebas. Estas actividades se abordan en las siguientes secciones. Se necesitarán más o menos actividades dependiendo de la complejidad de la AAP.

Capturar los requisitos necesarios para definir una AAP adecuada.

Los requisitos para un enfoque de automatización de pruebas deben tener en cuenta los siguientes elementos:

- Qué actividad o fase del proceso de prueba debe automatizarse, por ejemplo, la gestión de la prueba, el diseño de la prueba, la generación de la prueba o la ejecución de la prueba. Tenga en cuenta que la automatización de pruebas refina el proceso fundamental de la prueba insertando la generación de pruebas entre el diseño de la prueba y la implementación de la prueba.
- Qué nivel de test se debe soportar, por ejemplo, nivel de componente, nivel de integración, nivel de sistema.
- Qué tipo de prueba se debe soportar, por ejemplo, pruebas funcionales, pruebas de conformidad, pruebas de interoperabilidad.
- Qué rol de prueba debe ser soportado, por ejemplo, ejecutor de pruebas, analista de pruebas, arquitecto de pruebas, jefe de pruebas.
- Qué producto software, línea de producto software, familia de producto software debería ser soportado, por ejemplo, para definir el ámbito y la duración de la SAP implementada.
- Qué tecnologías SSP deben ser soportadas, por ejemplo, para definir la AAP en vista de su compatibilidad con las tecnologías de la SSP.

Comparar y contrastar diferentes enfoques de diseño/arquitectura

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

Aspectos que se deben tener en cuenta para la capa de generación de pruebas:

- Selección de la generación de pruebas manual o automatizada.
- Selección, por ejemplo, de la generación de pruebas basada en requisitos, basada en datos, basada en escenarios o basada en el comportamiento.
- Selección de estrategias de generación de pruebas (por ejemplo, cobertura de modelos tales como árboles de clasificación para enfoques basados en datos, cobertura de casos de uso/excepción para enfoques basados en escenarios, cobertura de transición/estado/camino para enfoques basados en el comportamiento, etc.).
- Elección de la estrategia de selección de pruebas. En la práctica, la generación de pruebas combinatorias completas es inviable, dado que puede conducir a una explosión de casos de prueba. Por lo tanto, se deben utilizar criterios de cobertura prácticos, factores de ponderación, evaluaciones de riesgos, etc., para guiar la generación de pruebas y la posterior selección de pruebas.

Aspectos que se deben tener en cuenta para la capa de definición de pruebas:

- Selección de la definición de prueba guiada por datos, guiada por palabras clave, basada en patrones o guiada por modelos.
- Selección de la notación para la definición de la prueba (por ejemplo, tablas, notación basada en el estado, notación estocástica, notación del flujo de datos, notación del proceso de negocio, notación basada en el escenario, etc. mediante el uso de hojas de cálculo, lenguajes de prueba específicos del dominio, Testing and Test Control Notation (TTCN- 3 por sus siglas en inglés), el Perfil de Pruebas UML (UTP por sus siglas en inglés), etc.).
- Selección de guías de estilo y directrices para la definición de pruebas de alta calidad.
- Selección de repositorios para casos de prueba (hojas de cálculo, bases de datos, archivos, etc.).

Aspectos que se deben tener en cuenta para la capa de ejecución de pruebas:

- Selección de la herramienta de ejecución de pruebas.
- Selección del enfoque de interpretación (mediante el uso de una máquina virtual) o de compilación para implementar los procedimientos de prueba - generalmente esta elección depende de la herramienta de ejecución de pruebas elegida.
- Selección de la tecnología de implementación para implementar procedimientos de prueba (imperativo, como C; funcional, como Haskell o Erlang; orientada a objetos, como C++, C#, Java; lenguajes de guion, como Python o Ruby, o una tecnología específica de herramienta) - en general, esta elección depende de la herramienta de ejecución de pruebas elegida.
- Selección de librerías de ayuda para facilitar la ejecución de la prueba (por ejemplo, librerías de dispositivos de prueba, librerías de codificación/decodificación, etc.).

Aspectos que se deben tener en cuenta para la capa de adaptación de pruebas:

- 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 trazar la ejecución de la prueba (por ejemplo, incluyendo la cronología de la ejecución de la prueba).

Identificar áreas en las que la abstracción puede aportar beneficios

La abstracción en una AAP permite la independencia de la tecnología en el sentido de que el mismo juego de pruebas se puede utilizar en diferentes entornos de prueba y en diferentes tecnologías destino. Se incrementa la portabilidad de los artefactos de prueba. Además, se garantiza la neutralidad con respecto a proveedores, lo que evita los efectos de bloqueo para una SAP. La abstracción también mejora la mantenibilidad y adaptabilidad para las tecnologías del SSP nuevas o en evolución. Adicionalmente, la abstracción ayuda a hacer que una AAP (y sus instanciaciones por parte de las SAP's) sea más accesible para los no técnicos, ya que los juegos de prueba pueden ser documentados (incluyendo medios gráficos) y explicados a un nivel superior, lo que mejora la legibilidad y la capacidad de ser entendido.

El IAP necesita acordar con los implicados en desarrollo de software, aseguramiento de la calidad y pruebas qué nivel de abstracción utilizar en qué área de la SAP. Por ejemplo, ¿qué interfaces de la capa de adaptación y/o ejecución de pruebas deben externalizarse, definirse formalmente y mantenerse estables a lo largo de la vida útil de la AAP? También se necesita acordar si está en uso una definición de prueba abstracta o si la AAP utiliza una capa de ejecución de prueba sólo con guiones de prueba. Del mismo modo, se debe entender si la generación de pruebas se abstrae mediante el uso de modelos de prueba y enfoques de prueba basados en modelos. El IAP necesita ser consciente de que hay compromisos entre implementaciones sofisticadas y sencillas de una AAP con respecto a la totalidad de la funcionalidad, mantenibilidad y capacidad de ampliación. Una decisión sobre qué abstracción utilizar en una AAP debe tener en cuenta estos compromisos.

Cuanto más abstracción se utilice para una AAP, más flexible será con respecto a una evolución posterior o a una transición hacia nuevos enfoques o tecnologías. Esto se produce a costa de mayores inversiones iniciales (por ejemplo, una arquitectura y herramientas de automatización de pruebas más complejas, mayores requisitos para el conjunto de competencias, mayores curvas de aprendizaje), lo que retrasa el punto de equilibrio inicial pero puede ser rentable a largo plazo. También puede conducir a un menor rendimiento de la SAP.

Mientras que las consideraciones detalladas del Retorno de la Inversión (ROI por sus siglas en inglés) son responsabilidad del GAP, el IAP necesita aportar las entradas al análisis del Retorno de la Inversión proporcionando evaluaciones técnicas y comparaciones de diferentes arquitecturas y enfoques de automatización de pruebas con respecto a tiempos, costos, esfuerzos y beneficios.

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

El acceso a las interfaces de prueba del SSP es fundamental para cualquier ejecución de pruebas automatizadas. El acceso puede estar disponible en los siguientes niveles:

- A nivel de software, por ejemplo, el SSP y el software de prueba están vinculados entre sí.
- A nivel de Interfaz de Programación de Aplicaciones (API por sus siglas en inglés), por ejemplo, la SAP invoca las funciones/operaciones/métodos proporcionados por una interfaz de programación de aplicaciones (remota).
- A nivel de protocolo, por ejemplo, la SAP interactúa con el SSP a través de HTTP, TCP, etc.
- A nivel de servicio, por ejemplo, el 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 una Interfaz de Programación de Aplicaciones (API por sus siglas en inglés), protocolos o servicios. Estos paradigmas incluyen los siguientes elementos:

- Paradigma guiado por eventos, el cual impulsa la interacción a través de eventos que se intercambian en un bus de eventos.
- Paradigma cliente-servidor, que dirige la interacción a través de la invocación de servicios desde los solicitantes de servicios hasta el proveedor de servicios.
- El paradigma entre pares ("peer-to-peer"), que impulsa la interacción a través de la invocación de servicios de cualquiera de los pares.

A menudo, la elección del paradigma depende de la arquitectura SSP y puede tener implicaciones en la arquitectura del SSP. La interconexión entre el SSP y la AAP se debe analizar y diseñar cuidadosamente a fin de seleccionar una arquitectura segura para el futuro entre los dos sistemas.

Entender el entorno del SSP

Un SSP puede ser un software independiente o un software que funcione sólo en relación con otro software (por ejemplo, sistemas de sistemas), hardware (por ejemplo, sistemas integrados) o componentes del entorno (por ejemplo, sistemas ciberfísicos). Una SAP simula o emula el entorno del SSP como parte de una configuración de prueba automatizada.

Entre los ejemplos de entornos de prueba y ejemplos de uso se incluyen los siguientes:

- Un ordenador con el SSP y la SAP - útil para probar una aplicación software.
- Ordenadores individuales conectados en red para un SSP y una SAP respectivamente - útil 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, por ejemplo, en un decodificador.
- Dispositivos de prueba en red para emular el entorno operativo del SSP - útil para probar el software de un router de red.
- Simuladores para simular el entorno físico del SSP - útiles para probar el software de una unidad de control embebida.

Tiempo y complejidad para una implementación determinada de una arquitectura de producto de prueba

Mientras que la estimación del esfuerzo para un proyecto de una SAP es responsabilidad de un GAP, un IAP necesita apoyar a un GAP en esta tarea proporcionando buenas estimaciones de los tiempos y la complejidad del diseño de una AAP. Entre los métodos de estimación y ejemplos se incluyen los siguientes:

- Estimación basada en la analogía, como por ejemplo puntos función, estimación de tres puntos, delphi de banda ancha y estimación experta.
- Estimación mediante el uso de estructura de descomposición del trabajo, como el que se encuentra en software de gestión o plantillas de proyecto.
- Estimación paramétrica como el Modelo de Coste Constructivo (COCOMO).
- Estimaciones basadas en el tamaño, como el Análisis de Puntos Función, el Análisis de Puntos de Historia o el Análisis de Casos de Uso.
- Estimaciones grupales como la Planificación Poker.

Facilidad de uso para una determinada implementación de arquitectura del producto de prueba

Además de la funcionalidad de la SAP, su compatibilidad con el SSP, su estabilidad y evolución a largo plazo, sus requisitos de esfuerzo y las consideraciones sobre el retorno de la inversión (ROI por sus siglas en inglés), un IAP tiene la responsabilidad específica de abordar los problemas de usabilidad de una SAP. Esto incluye, pero no se limita a:

- Diseño orientado al probador.
- Facilidad de uso de la SAP.
- Soporte de la SAP para otros roles en el desarrollo de software, aseguramiento de la calidad y gestión de proyectos.
- Organización, navegación y búsqueda eficaces en/con la SAP.
- Documentación útil, manuales y texto de ayuda para la SAP.
- Informes prácticos de y sobre la SAP.
- Diseños iterativos para abordar la realimentación y las percepciones empíricas de la SAP.

3.2.2 Enfoques para la Automatización de Casos de Prueba

Los casos de prueba deben traducirse en secuencias de acciones que se ejecutan contra un SSP. Esa secuencia de acciones puede estar documentada en un procedimiento de prueba y/o puede ser implementada en un guion 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 verificar que el SSP ha logrado el resultado esperado. Se pueden utilizar diversos enfoques para crear la secuencia de acciones:

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

Hay que señalar que las opciones dependen en gran medida del contexto del proyecto. También puede ser eficaz iniciar la automatización de pruebas aplicando una de las opciones menos avanzadas, ya que por lo general resultan más fáciles de implementar. Esto puede proporcionar valor añadido a corto plazo, aunque dará lugar a una solución menos mantenible.

Los enfoques consolidados para automatizar los casos de prueba incluyen:

- Enfoque de captura/reproducción, que puede ser utilizado para la opción 1.
- Enfoque de guiones estructurados, enfoque basado en datos y enfoque basado en palabras clave, que pueden utilizarse para las opciones 2 ó 3.
- Pruebas basadas en modelos (incluido el enfoque basado en proceso), que pueden utilizarse para la opción 4.

Estos enfoques se explican posteriormente en términos de los conceptos básicos y de los pros y contras.

Enfoque de captura/reproducción

Concepto principal

En los enfoques de captura/reproducción, las herramientas se utilizan para capturar las interacciones con el SSP mientras se ejecuta la secuencia de acciones definida por un procedimiento de prueba. Se capturan las entradas; las salidas también se pueden registrar para realizar comprobaciones posteriores. Durante la repetición de eventos, hay diversas posibilidades para la comprobación de salidas manuales y automatizadas:

- Manual: el probador tiene que vigilar las salidas del SSP para detectar anomalías.
- Completo: todas las salidas del sistema registradas durante la captura deben ser reproducidas por el SSP.
- Exacto: todas las salidas del sistema que se registraron durante la captura deben ser reproducidas por el SSP al nivel de detalle de la grabación.
- Puntos de comprobación: sólo se comprueban las salidas del sistema seleccionadas en determinados puntos para los valores especificados.

Pros

El enfoque de captura/reproducción se puede utilizar para SSP's a nivel de interfaz gráfica de usuario (GUI por sus siglas en inglés) y/o interfaz de programación de aplicaciones (API por sus siglas en inglés). Inicialmente, es fácil de configurar y utilizar.

Contras

Los guiones de captura/reproducción son difíciles de mantener y evolucionar porque la ejecución capturada de SSP depende en gran medida de la versión del SSP de la que se ha tomado la captura. Por ejemplo, cuando se graba a nivel de la interfaz gráfica de usuario (GUI por sus siglas en inglés), los cambios en la disposición de la interfaz gráfica de usuario pueden afectar al guion de prueba, incluso si sólo se trata de un cambio en la posición de un elemento de la interfaz gráfica de usuario. Por lo tanto, los enfoques de captura/reproducción siguen siendo vulnerables a los cambios.

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

Guion lineal

Concepto principal

Como con todas las técnicas de guiones, los guiones lineales comienzan con algunos procedimientos de prueba manuales. Hay que tener en cuenta que estos pueden no ser documentos escritos - el conocimiento sobre qué pruebas realizar y cómo realizarlas puede ser "conocido" por uno o más Analistas de Pruebas.

Cada prueba se ejecuta manualmente mientras la herramienta de prueba registra la secuencia de acciones y, en algunos casos, captura la salida visible del SSP en la pantalla. Esto resulta, generalmente, en un guion (típicamente de gran tamaño) para cada procedimiento de prueba. Los guiones grabados se pueden editar para mejorar la legibilidad (por ejemplo, añadiendo comentarios para explicar lo que está ocurriendo en puntos clave) o añadir comprobaciones adicionales utilizando el lenguaje de guiones de la herramienta.

Los guiones pueden ser reproducidos por la herramienta, haciendo que la herramienta repita las mismas acciones tomadas por el probador cuando el guion fue grabado. Aunque esto se puede utilizar para automatizar las pruebas de interfaz gráfica de usuario (GUI por sus siglas en inglés), no es una buena técnica para utilizar cuando se va a automatizar un gran número de pruebas y son necesarias para numerosas versiones del software. Esto se debe al elevado coste de mantenimiento que suelen ocasionar los cambios en el SSP (cada cambio en el SSP puede requerir muchos cambios en los guiones grabados).

Pros

Las ventajas de los guiones lineales se centran en el hecho de que se necesita poco o ningún trabajo de preparación antes de comenzar la automatización. Una vez que haya aprendido a utilizar la herramienta es simplemente cuestión de grabar una prueba manual y volver a reproducirla (aunque la parte de grabación de este proceso puede requerir interacción adicional con la herramienta de prueba para solicitar que se realicen comparaciones de la salida real con la esperada para verificar que el software está funcionando correctamente). No se requieren competencias de programación, pero suelen ser útiles.

Contras

Las desventajas de los guiones lineales son numerosas. La cantidad de esfuerzo requerido para automatizar un procedimiento de prueba determinado dependerá en gran medida 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 de esfuerzo similar a la del procedimiento de prueba número 100. En otras palabras, no hay mucho margen para reducir el coste de la construcción de nuevas pruebas automatizadas.

Adicionalmente, si hubiera un segundo guion que realizara una prueba similar aunque con diferentes valores de entrada, ese guion contendría la misma secuencia de instrucciones que el primer guion; solo la información incluida con las instrucciones (conocida como argumentos o parámetros de la instrucción) diferiría. Si hubiera varias pruebas (y por lo tanto guiones) todas ellas tendrían la misma secuencia de instrucciones, las cuales tendrían que ser mantenidas 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 propietario (exclusivos de la herramienta), por lo que lleva tiempo aprender el idioma y dominarlo.

Los guiones grabados sólo contienen afirmaciones generales en los comentarios, si es que contienen alguna. Los guiones largos, en particular, se documentan mejor con comentarios para explicar lo que está sucediendo en cada paso de la prueba. Esto facilita el mantenimiento. Los guiones pronto pueden llegar a ser muy extensos (conteniendo muchas instrucciones) cuando la prueba comprenda muchos pasos.

Los guiones no son modulares y son difíciles de mantener. Los guiones lineales no siguen los paradigmas comunes de reutilización y modularidad del software y están estrechamente acoplados a la herramienta utilizada.

Guion estructurado

Concepto principal

La principal diferencia entre la técnica de guion estructurado y la técnica de guion lineal es la introducción de una librería de guiones. Esta contiene guiones reutilizables que realizan secuencias de instrucciones que generalmente se necesitan en diferentes pruebas. Buenos ejemplos de estos guiones son aquellos que interactúan, por ejemplo, con las operaciones de las interfaces SSP.

Pros

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

Las ventajas de los guiones estructurados se obtienen, en gran medida, a través de la reutilización de guiones. Se pueden automatizar más pruebas sin tener que crear el volumen de guiones que requeriría un enfoque de guiones lineales. Esto tiene un impacto directo en los costes de construcción y mantenimiento. La segunda y subsiguientes pruebas no requerirán tanto esfuerzo de automatización porque algunos de los guiones creados para implementar la primera prueba pueden ser reutilizados nuevamente.

Contras

El esfuerzo inicial para crear los guiones compartidos puede ser visto como una desventaja, pero esta inversión inicial debería ser muy rentable si se aborda adecuadamente. Se requerirán competencias de programación para crear todos los guiones, ya que una simple grabación no será suficiente. La librería de guiones debe estar bien gestionada, es decir, los guiones deben estar documentados y debe ser fácil, para los Analistas de Pruebas Técnicas, encontrar los guiones necesarios (por lo que una convención de nomenclatura adecuada será de gran ayuda en este caso).

Pruebas Guiadas por Datos

Concepto principal

La técnica de guiones guiada por datos se desarrolla a partir de la técnica de guiones estructurados. La diferencia más significativa se debe a la forma en que se tratan las entradas de prueba. Las entradas son extraídas de los guiones y almacenadas en uno o más archivos separados (típicamente llamados archivos de datos).

Esto significa que el guion de prueba principal se puede reutilizar para implementar varias pruebas (en lugar de una sola prueba). Generalmente, el guion principal de prueba "reutilizable" se denomina guion de "control". El guion de control contiene la secuencia de instrucciones necesarias para realizar las pruebas pero lee los datos de entrada de un archivo de datos. Una prueba de control se puede utilizar para muchas pruebas, pero generalmente es insuficiente para automatizar un rango amplio de pruebas. Por lo tanto, serán necesarios varios guiones de control, pero esto es sólo una fracción del número de pruebas que están automatizadas.

Pros

El coste de añadir nuevas pruebas automatizadas puede reducirse significativamente mediante esta técnica de guion. Esta técnica se utiliza para automatizar muchas variaciones de una prueba útil, dando pruebas más profundas en un área específica y puede incrementar 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 rellenando uno o más archivos de datos. Esto da a los Analistas de Pruebas más libertad para especificar pruebas automatizadas sin depender tanto de los Analistas de Pruebas Técnicas (que pueden ser un recurso escaso).

Contras

La necesidad de gestionar los archivos de datos y asegurarse de que puedan ser leídos por la SAP es una desventaja, pero se puede abordar de forma adecuada.

También, se pueden pasar por alto casos de prueba negativos importantes. Las pruebas negativas son una combinación de procedimientos de prueba y datos de prueba. En un enfoque centrado principalmente en datos de prueba, se pueden omitir " procedimientos de prueba negativos".

Pruebas guiadas por palabras clave

Concepto principal

La técnica de guion guiado por palabras clave se basa en la técnica de guion guiado por datos. Hay dos diferencias importantes: (1) los archivos de datos se denominan ahora archivos de "definición de pruebas" o algo similar (por ejemplo, archivos de palabras de acción); y (2) sólo hay un guion 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 manera que tengan sentido para el Analista de Pruebas, las pruebas que se describen y la aplicación que se está probando. Estos se utilizan principalmente (pero no de forma exclusiva) para representar interacciones de negocio de alto nivel con un sistema (por ejemplo, "realizar un pedido"). Cada palabra clave representa un número de interacciones detalladas con el sistema sujeto a prueba. Se utilizan secuencias de palabras clave (incluidos los datos de prueba relevantes) para especificar los casos de prueba. Se pueden utilizar 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 Pruebas incluye la creación y el mantenimiento de los archivos de palabras clave. Esto significa que una vez implementados los guiones de soporte, los Analistas de Pruebas pueden añadir pruebas "automatizadas" simplemente especificándolas en un archivo de palabras clave (como en el caso de los guiones guiados por datos).

Pros

Una vez que el guion de control y los guiones de soporte para las palabras clave han sido escritos, el coste de añadir nuevas pruebas automatizadas será muy reducido por esta técnica de guion.

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 da a los Analistas de Pruebas más libertad para especificar pruebas automatizadas sin depender tanto de los Analistas de Pruebas Técnicas (que pueden ser un recurso escaso). El beneficio del enfoque basado en palabras clave sobre el enfoque basado en datos a este respecto 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 una cuenta", "realizar un pedido", "comprobar el estado de un pedido" son todas acciones posibles para una aplicación de compra en línea que incluyen una serie de pasos detallados. Cuando un Analista de Pruebas describe una prueba de 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 basado en palabras clave es aplicar estas acciones de alto nivel y permitir que las pruebas se definan en términos de acciones de alto nivel sin referencia a los pasos detallados.

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

Contras

La implementación de las palabras clave sigue siendo una gran tarea para los ingenieros de automatización de pruebas, especialmente si se utiliza una herramienta que no ofrece soporte para esta técnica de guion. En el caso de los sistemas pequeños, la implementación puede ser demasiado costosa y los costes serían mayores que los beneficios.

Se debe tener cuidado para asegurar que se implementen las palabras clave correctas. Unas palabras clave buenas se utilizarán a menudo en muchas pruebas diferentes, mientras que las palabras clave de baja calidad sólo se utilizarán una o pocas veces.

Guion guiado por proceso

Concepto principal

El enfoque guiado por procesos se basa en la técnica de guion guiado por palabras clave con la diferencia de que los escenarios, que representan casos de uso del SSP y sus variantes, constituyen los guiones que se parametrizan con datos de prueba o se combinan en definiciones de prueba de mayor nivel.

Estas definiciones de prueba son más fáciles de afrontar, ya que se puede determinar la relación lógica entre las acciones, por ejemplo, "comprobar el estado del pedido" después de "realizar un pedido" en una prueba de prestaciones o "comprobar el estado del pedido" sin un "realizar un pedido" previo en una prueba de robustez.

Pros

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

Contras

Los procesos de un SSP pueden no ser fáciles de comprender por un Analista de Pruebas Técnicas, al igual que la implementación de los guiones orientados al proceso, especialmente si la herramienta no soporta ninguna lógica de procesos de negocio.

Es preciso también asegurarse de que se implementen los procesos correctos, mediante el uso de palabras clave correctas. Los procesos de buena calidad serán referenciados por otros procesos y resultarán en muchas pruebas relevantes, mientras que los procesos de mala calidad no serán rentables 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 (véase también el Programa de Estudio Pruebas Basadas en Modelos del ISTQB) - en contraposición a la ejecución automatizada de casos de prueba - mediante el uso de captura/reproducción, guion lineal, guion estructurado, guion guiado por datos o guion guiado por procesos. Las pruebas basadas en modelos utilizan modelos (semi) formales que se abstraen de las tecnologías de guion de la AAP. Se pueden utilizar diferentes métodos de generación de pruebas para obtener pruebas para cualquiera de los marcos de trabajo de guion discutidos previamente.

Pros

Las pruebas basadas en modelos permiten, mediante abstracción, concentrarse en la esencia de las pruebas (en términos de lógica de negocio, datos, escenarios, configuraciones, etc. a probar). También permite generar pruebas para diferentes sistemas destino y tecnologías objetivo, de modo que los modelos utilizados para la generación de pruebas constituyen una representación segura para el futuro del software de prueba que puede ser reutilizado y mantenido a medida que la tecnología evoluciona.

En caso de cambios en los requisitos, el modelo de prueba sólo tiene que adaptarse; se genera automáticamente un conjunto completo de casos de prueba. Las técnicas de diseño de casos de prueba están incorporadas en los generadores de casos de prueba.

Contras

Se requiere experiencia en modelado para ejecutar un enfoque de pruebas 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 pruebas basadas en modelos todavía no son una corriente predominante, 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 ser asegurados en términos de calidad y mantenidos.

3.2.3 Consideraciones técnicas con respecto al SSP

Además, al diseñar una AAP deben tenerse en cuenta los aspectos técnicos de un SSP. Algunos de ellos se discuten a continuación, aunque no se trata de una lista completa, sino que deben servir como ejemplo de los aspectos importantes.

Interfaces del SSP

Un SSP tiene interfaces internas (dentro del sistema) e interfaces externas (al entorno del sistema y a sus usuarios o por componentes expuestos). La AAP debe ser capaz de controlar y/u observar todas las interfaces del SSP que puedan verse potencialmente afectadas por los procedimientos de prueba (es decir, las interfaces deben contar con capacidad de ser probadas). Además, también puede ser necesario registrar las interacciones entre el SSP y la SAP con diferentes niveles de detalle, que suelen incluir marcas de tiempo.

Durante la definición de la arquitectura es necesario concentrarse en la prueba (por ejemplo, una prueba) al principio del proyecto (o de forma continua en entornos ágiles) para verificar la disponibilidad de las interfaces de prueba necesarias o de las instalaciones de prueba necesarias para que el SSP cuente con capacidad de ser probado (diseño para la capacidad de ser probado).

Datos del SSP

Un SSP utiliza datos de configuración para controlar su instanciación, configuración, gestión, etc. Además, utiliza datos de usuario que procesa. Un SSP también puede utilizar 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 instanciados por la AAP. La forma específica de hacer frente a los datos del SSP se decide en el diseño de la AAP. Dependiendo del enfoque, los datos pueden ser tratados como parámetros, hojas de datos de prueba, bases de datos de prueba, datos reales, etc.

Configuraciones del SSP

Un SSP puede desplegarse en diferentes configuraciones, por ejemplo, en diferentes sistemas operativos, en diferentes dispositivos 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 requerir diferentes configuraciones de prueba (en un laboratorio) o configuraciones de prueba virtuales (en la nube) de la AAP en combinación con una configuración dada del SSP. También puede requerir que se introduzcan simuladores y/o emuladores de componentes seleccionados del SSP para aspectos seleccionados del SSP.

Estándares y marcos legales del SSP

Además de los aspectos técnicos relativos al SSP, puede ser que el diseño de la AAP deba respetar requisitos legales y/o estándares de manera que se diseñe la AAP de forma compatible. Como ejemplos cabe citar los requisitos de privacidad de los datos de prueba o los requisitos de confidencialidad que afectan a las capacidades de registro y elaboración de informes de la AAP.

Herramientas y entornos de herramientas utilizados para desarrollar el SSP

Junto al 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 conjunto de herramientas del SSP para permitir la compatibilidad de las herramientas, la trazabilidad y/o la reutilización de artefactos.

Interfaces de prueba en el producto software

Se recomienda especialmente no quitar todas las interfaces de prueba con anterioridad a la entrega del producto. En la mayoría de los casos, estas interfaces se pueden dejar en el SSP sin causar problemas en el producto final. Cuando se dejan en el sitio, las interfaces pueden ser utilizadas por los ingenieros de servicio y soporte para el diagnóstico de problemas, así como para probar entregas de mantenimiento. Es importante verificar que las interfaces no presenten riesgos de seguridad. Si es necesario, los desarrolladores normalmente pueden deshabilitar estas interfaces de prueba para que no se puedan utilizar fuera del área de desarrollo.

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

Se deben tener en cuenta los aspectos de los procesos de desarrollo y aseguramiento de la calidad de un SSP a la hora de diseñar una AAP. Algunos de ellos se discuten a continuación, aunque no se trata de una lista completa, sino que deben servir como ejemplo de aspectos importantes.

Requisitos de control de la ejecución de la prueba

Dependiendo del nivel de automatización requerido por la AAP, la ejecución interactiva de pruebas, la ejecución de pruebas en modo por lotes o la ejecución de pruebas totalmente automatizadas pueden requerir ser soportadas por la AAP.

Requisitos relativos a informes

Dependiendo de los requisitos de informes, incluidos los tipos de informes y sus estructuras, la AAP debe ser capaz de soportar informes de prueba fijos, parametrizados o definidos en diferentes formatos y diseños.

Roles y derechos de acceso

Dependiendo de los requisitos de seguridad, la AAP puede estar obligada a proporcionar un sistema de roles y derechos de acceso.

Conjunto de herramientas establecido

La gestión de proyecto, la gestión de pruebas, el repositorio de código y pruebas, el seguimiento de defectos, la gestión de incidencias, el análisis de riesgos, etc. del SSP pueden estar soportados por las herramientas que componen el conjunto de herramientas establecido. La AAP también cuenta con el soporte de una herramienta o conjunto de herramientas que debe integrarse a la perfección con las demás herramientas del conjunto. Además, los guiones de prueba se deben almacenar y versionar 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 la SAP

El desarrollo de una SAP es comparable a otros proyectos de desarrollo de software. Puede seguir los mismos procedimientos y procesos, incluyendo revisiones entre pares por desarrolladores y probadores. Lo específico de una SAP es su compatibilidad y sincronización con el SSP. Éstos deben tenerse en cuenta en el diseño de la AAP (véase la sección 3.2) y en el desarrollo de la SAP. Además, el SSP se ve afectado por la estrategia de prueba, por ejemplo, tener que poner las interfaces de prueba a disposición de la SAP.

Esta sección utiliza el ciclo de vida de desarrollo de software (SDLC por sus siglas en inglés) 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 en marcha para el desarrollo del SSP y/o de la SAP, por lo que se deberán adaptar en consecuencia.

En la Figura 2 se presenta el Modelo de Ciclo de Vida de Desarrollo Software básico para la SAP.

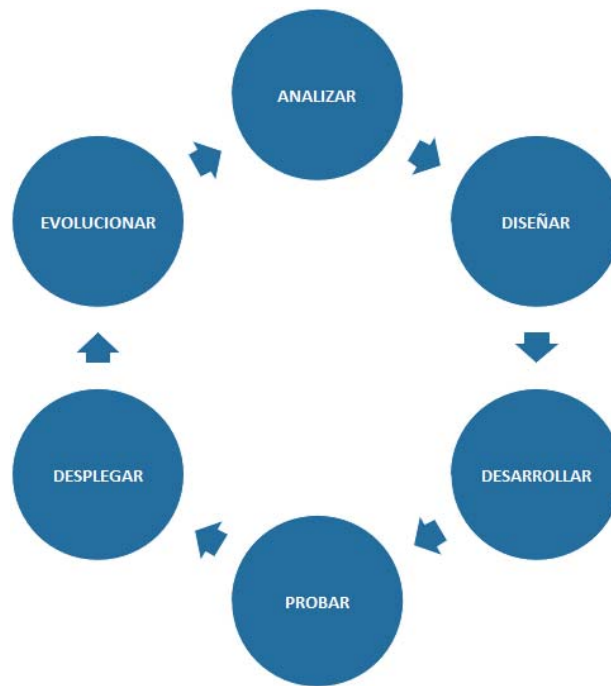


Figura 2: Modelo de Ciclo de Vida de Desarrollo básico para la SAP.

El conjunto de requisitos para un SAT debe ser analizado y recopilado (ver Figura 2). Los requisitos dirigen el diseño de la SAP tal como se define en su AAP (véase la sección 3.2). El diseño se convierte en software mediante enfoques de ingeniería de software. Se debe tener en cuenta que una SAP también puede utilizar hardware dedicado del dispositivo de prueba, lo cual está fuera de consideración para este programa de estudio. Como cualquier otro software, una SAP necesita ser probada. Esto se realiza normalmente mediante pruebas de capacidad básica para la SAP, seguidas de una interacción entre la SAP y el SSP. Después del despliegue y uso de una SAP, a menudo se necesita una evolución de la SAP para añadir más capacidad de prueba, cambiar pruebas o actualizar la SAP para que coincida con el cambio en el SSP. La evolución de la SAP requiere una nueva ronda de desarrollo de la SAP según el Modelo de Ciclo de Vida de Desarrollo de Software (SDLC por sus siglas en inglés).

Se debe tener en cuenta también que el Modelo de Ciclo de Desarrollo de Software no muestra la copia de seguridad, el archivo y el desmantelamiento de una SAP. Al igual que con el desarrollo de la SAP, estos procedimientos deben seguir los métodos establecidos por la organización.

3.3.2 Compatibilidad entre la SAP y el SSP

Compatibilidad de proceso

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 conveniente coordinar los procesos para el desarrollo del SSP, el desarrollo de la SAP y las pruebas. Se puede obtener un gran beneficio cuando el desarrollo del SSP y la SAP son compatibles en términos de estructura de procesos, gestión de procesos y soporte de herramientas.

Compatibilidad de equipo

La compatibilidad del equipo es otro aspecto de la compatibilidad entre el desarrollo de la SAP y el SSP. Si se adopta una mentalidad compatible para abordar y gestionar el desarrollo de la SAP y el SSP, ambos equipos se beneficiarán de la revisión de los requisitos, diseños y/o artefactos de desarrollo de cada uno, de la discusión de los problemas y de la búsqueda de soluciones compatibles. La compatibilidad de los equipos también ayuda en la comunicación e interacción entre ellos.

Compatibilidad tecnológica

Además, se debe considerar la compatibilidad tecnológica entre la SAP y el SSP. Es beneficioso diseñar e implementar una interacción sin fisuras entre la SAP y el SSP desde el principio. Incluso si esto no es posible (por ejemplo, porque no se dispone de soluciones técnicas ni para la SAP ni para el SSP), puede ser posible una interacción sin fisuras mediante el uso de adaptadores, envolturas u otras formas de intermediarios.

Compatibilidad de herramientas

Es necesario considerar la compatibilidad entre las herramientas de gestión, desarrollo y aseguramiento de la calidad de la SAP y el SSP. 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 el SSP serán más fáciles.

3.3.3 Sincronización entre la SAP y el SSP

Sincronización de requisitos

Después de la educación⁹ de los requisitos, se desarrollarán los requisitos de ambos SSP y SAP. Los requisitos de la SAP se pueden agrupar en dos grupos principales de requisitos: (1) requisitos que abordan el desarrollo de la SAP como un sistema basado en software, tales como los requisitos de las características de la SAP para el diseño de la prueba, la especificación de la prueba, el análisis de los resultados de la prueba, etc. y (2) requisitos que abordan la prueba del SSP por medio de la SAP. Estos requisitos de prueba corresponden a los requisitos del SSP y reflejan todas las características y prestaciones del SSP que deben ser probadas por la SAP. Cada vez que se actualizan los requisitos el SSP o de la SAP, es importante verificar la consistencia entre los dos y comprobar que todos los requisitos del SSP que deben ser probados por la SAP tienen requisitos de prueba definidos.

Sincronización de las fases de desarrollo

⁹ "educación" es la traducción del término "elicitation".

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

Sincronización del seguimiento de defectos

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

Sincronización de la evolución de SSP y SAP

Tanto el SSP como la SAP pueden evolucionar para dar cabida a nuevas características o desactivar características, 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 a la otra, por lo que la gestión de estos cambios debe abordar tanto el SSP como la SAP.

La Figura 3 y la Figura 4 muestran dos enfoques de sincronización entre los procesos de desarrollo del SSP y la SAP.

La Figura 3 muestra un enfoque en el que los dos procesos del Modelo de Ciclo de Vida de Desarrollo para el SSP y la SAP se sincronizan, principalmente, en dos fases: (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.

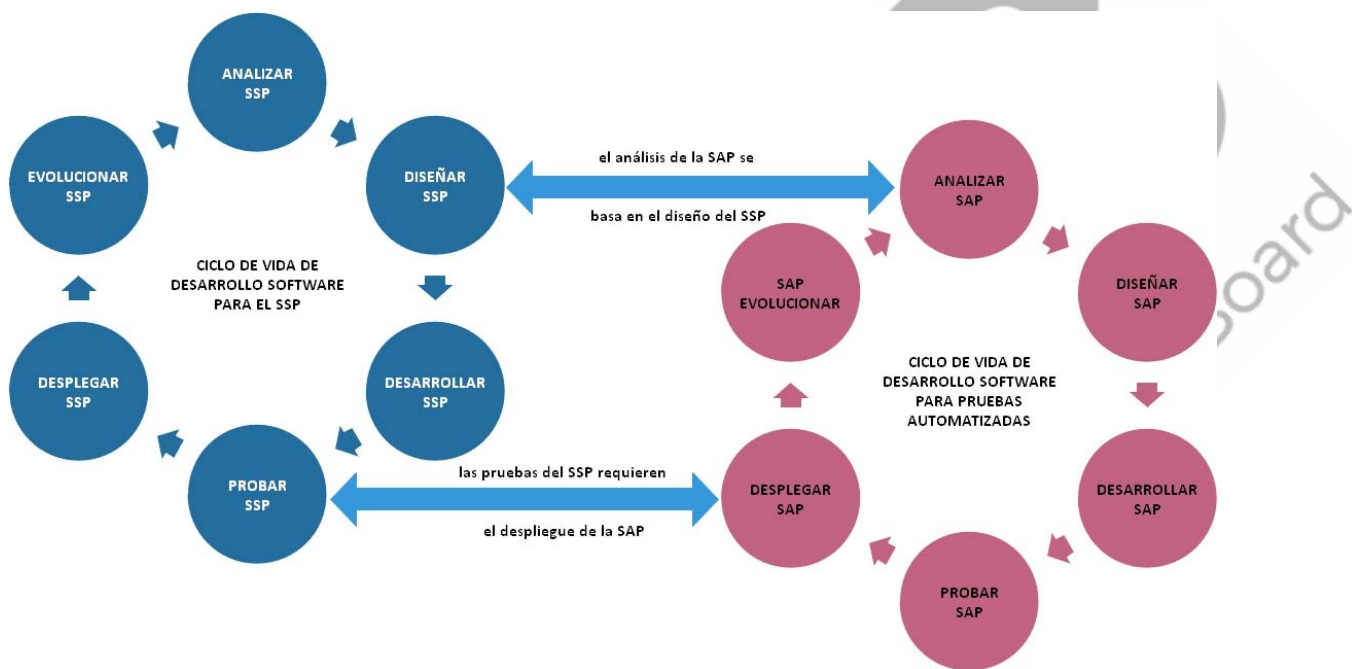


Figura 3: Ejemplo de sincronización 1 de los procesos de desarrollo de la SAP y el SSP.

La Figura 4 muestra un enfoque híbrido con pruebas manuales y automatizadas. Cuando se utilicen pruebas manuales antes de automatizar las pruebas o cuando se utilicen conjuntamente pruebas manuales y automáticas, el análisis de la SAP deberá basarse tanto en el diseño del SSP como en las pruebas manuales. De esta manera, la SAP se sincroniza con ambas. El segundo punto de sincronización importante para este enfoque es el mismo que antes: la prueba del SSP requiere pruebas desplegadas, que en el caso de las pruebas manuales podrían ser sólo los procedimientos de prueba manual que se deben seguir.

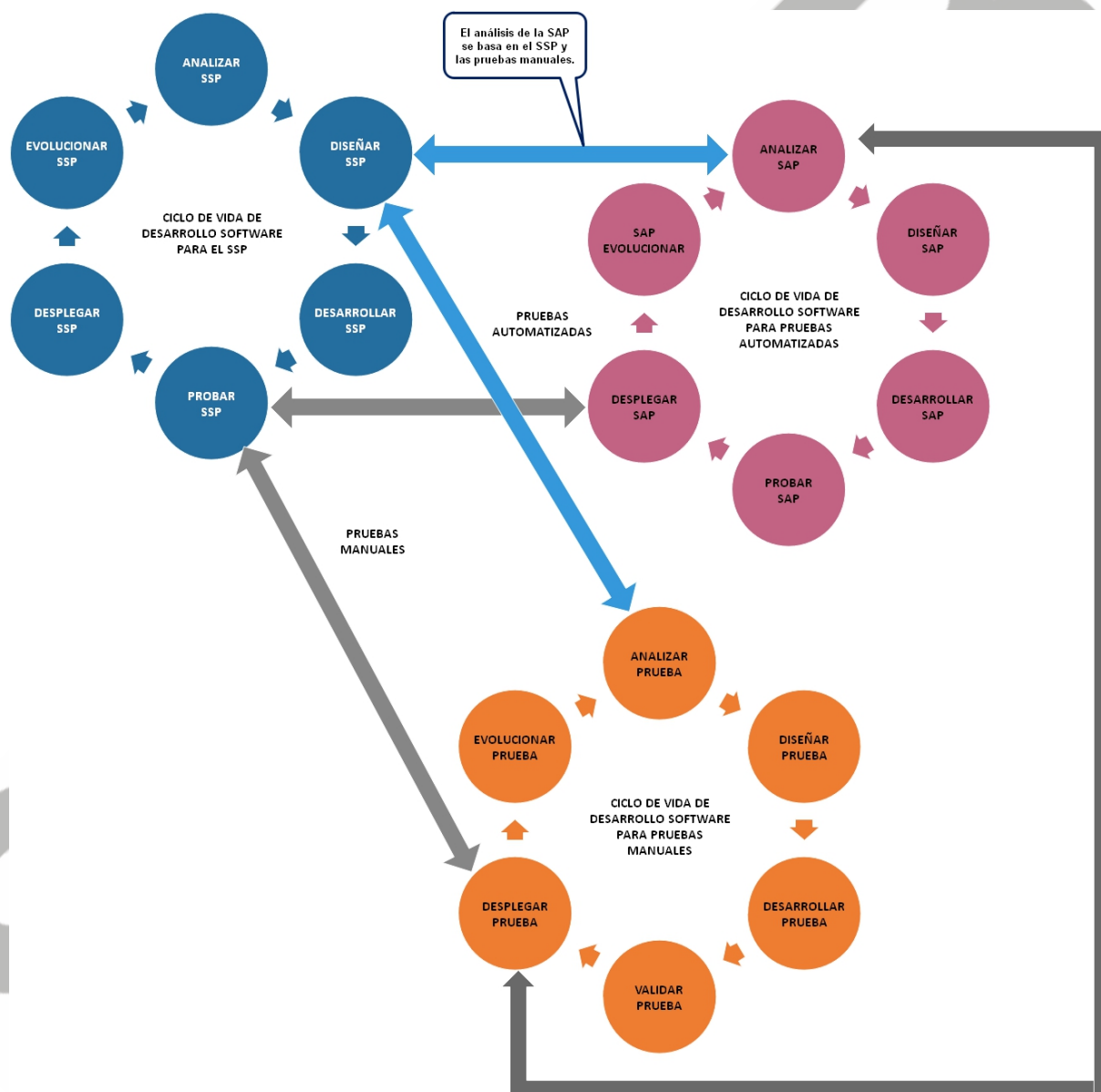


Figura 4: Ejemplo de sincronización 2 de los procesos de desarrollo de la SAP y el SSP

3.3.4 Construcción de la reutilización en la SAP

La reutilización de un SAP se refiere a la reutilización de artefactos de la SAP (de cualquier nivel de su arquitectura) a través de líneas de productos, marcos 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. Entre los artefactos reutilizables de la SAP se pueden encontrar:

- (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 las propias librerías de prueba.
- El motor de prueba y/o el marco de trabajo de informes de prueba.
- Los adaptadores para los componentes y/o interfaces del SSP.

Mientras que los aspectos de reutilización ya están resueltos cuando se define la AAP, la SAP puede ayudar a aumentar la capacidad de reutilización mediante:

- Seguir la AAP o revisarla y actualizarla cuando sea necesario.
- Documentar los artefactos de la SAP para que sean fáciles de comprender y puedan ser incorporados en nuevos contextos.
- Asegurar la corrección de cualquier artefacto de la SAP para que su uso en nuevos contextos se apoye en su alta calidad.

Es importante señalar que, si bien el diseño para la reutilización es principalmente una cuestión de la AAP, el mantenimiento y las mejoras para la reutilización son una preocupación a lo largo de todo el ciclo de vida de la SAP. Requiere una consideración y un esfuerzo continuos para hacer que se produzca la reutilización, para medir y demostrar el valor añadido de la reutilización, y para evangelizar a otros para que reutilicen las SAP's existentes.

3.3.5 Soporte para una Variedad de Sistemas Objetivo

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

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

Mientras que los primeros cuatro aspectos impactan a la SAP en cualquier nivel de prueba, el último se aplica principalmente a las pruebas a nivel de componentes y a nivel de integración.

La capacidad de una SAP para probar diferentes configuraciones de productos software se determina cuando se define la AAP. Sin embargo, la SAP tiene que implementar la capacidad de tratar la varianza

Probador Certificado

Programa de Estudio - Nivel Avanzado

Ingeniero de Automatización de Pruebas

técnica y tiene que permitir la gestión de las prestaciones y componentes de la SAP necesarios para las diferentes configuraciones de un producto software.

El tratamiento de la variedad de la SAP en relación con la variedad del producto software se puede abordar de diferentes formas:

- Se puede utilizar la gestión de versiones/configuración para la SAP y el SSP para suministrar las versiones y configuraciones respectivas de la SAP y el SSP que se correspondan mutuamente.
- Se puede utilizar la parametrización de la SAP para ajustar una SAP a una configuración del SSP.

Es importante señalar que, si bien el diseño de la variabilidad de la SAP es principalmente un asunto de la AAP, el mantenimiento y las mejoras de la variabilidad son una preocupación a lo largo del ciclo de vida de la SAP. Se requiere una consideración y esfuerzos continuos para revisar, añadir e incluso eliminar opciones y formas de variabilidad.

4. Riesgos y Contingencias del Despliegue - 150 minutos

Palabras Clave

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

Objetivos de Aprendizaje para Riesgos y Contingencias del Despliegue

4.1 Selección del Enfoque de Automatización de Pruebas y Planificación del Despliegue/Puesta en Marcha

NA-IAP-4.1.1 (K3) Aplicar directrices que apoyen las actividades del piloto y de despliegue de herramientas de prueba efectivas.

4.2 Evaluación del Riesgo y Estrategias de Mitigación

NA-IAP-4.2.1 (K4) Analizar los riesgos de despliegue e identificar los problemas técnicos que podrían conducir al fracaso del proyecto de automatización de pruebas, y planificar estrategias de mitigación.

4.3 Mantenimiento de la Automatización de Pruebas

NA-IAP-4.3.1 (K2) Comprender qué factores soportan y afectan la mantenibilidad de la SAP.

4.1 Selección del Enfoque de Automatización de Pruebas y Planificación del Despliegue/Puesta en Marcha

Hay dos actividades principales involucradas en la implementación y puesta en marcha de una SAP: piloto y despliegue. Los pasos que comprenden estas dos actividades variarán dependiendo del tipo de SAP y de la situación específica.

Para el piloto, al menos, se deben considerar los siguientes pasos:

- Identificar un proyecto adecuado.
- Planificar el piloto.
- Llevar a cabo el piloto.
- Evaluar el piloto.

Para el despliegue, al menos, se deben considerar los siguientes pasos:

- Identificar el(los) proyecto(s) objetivo inicial(es).
- Desplegar la SAP en los proyectos seleccionados.
- Monitorizar y evaluar la SAP en proyectos después de un período predefinido.
- Puesta en marcha en el resto de la organización/proyectos.

4.1.1 Proyecto Piloto

La implementación de una herramienta comienza, normalmente, con un proyecto piloto. El objetivo del proyecto piloto es asegurar que la SAP se puede utilizar para lograr los beneficios planificados. Entre los objetivos del proyecto piloto se encuentran los siguientes:

- Aprender más detalles sobre la SAP.
- Ver cómo se adecua la SAP a los procesos, procedimientos y herramientas existentes; identificar cómo necesitarían cambiar. (Generalmente se prefiere modificar la SAP para que se adecúe a los procesos/procedimientos existentes. Si estos deben ser ajustados para "soportar la SAP", esto debería ser al menos una mejora de los mismos procesos).
- Diseñar la interfaz de automatización para que coincida con las necesidades de los probadores.
- Decidir las formas estándar de utilizar, gestionar, almacenar y mantener la SAP y los activos de prueba, incluida la integración con la gestión de la configuración y la gestión de cambios (por ejemplo, decidir sobre las convenciones de asignación de nombres para archivos y pruebas, crear librerías y definir la modularidad de los juegos de prueba).
- Identificar métricas y métodos de medición para monitorizar la automatización de pruebas en uso, incluyendo usabilidad, mantenibilidad y capacidad de ampliación.
- Evaluar si los beneficios se pueden lograr a un coste razonable. Esta será una oportunidad para restablecer las expectativas una vez que se hayan utilizado las SAP's.
- Determinar qué competencias se requieren, cuáles de ellas están disponibles y cuáles faltan.

Identificar un proyecto adecuado

El proyecto piloto se debe seleccionar cuidadosamente utilizando las siguientes directrices:

- No seleccionar un proyecto crítico. Cuando el despliegue de la SAP cause retrasos, esto no debería tener un impacto importante en proyectos críticos. El despliegue de la SAP, al principio, costará tiempo. El equipo del proyecto debe ser consciente de ello.
- No seleccionar un proyecto trivial. Un proyecto trivial no es un buen candidato ya que el éxito del despliegue no implica el éxito en proyectos no triviales, y por lo tanto añade menos información a la necesaria para el despliegue.
- Involucrar a los implicados necesarios (incluida la dirección) en el proceso de selección.
- El SSP del proyecto piloto debe ser una buena referencia para los otros proyectos de la organización, por ejemplo, el SSP debe contener componentes representativos de la interfaz gráfica de usuario que deben ser automatizados.

Planificar el piloto

El proyecto piloto debe ser tratado como un proyecto de desarrollo normal: hacer un plan, reservar presupuesto y recursos, informar sobre el progreso, 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 defensor¹⁰) pueden invertir suficiente esfuerzo en el despliegue, incluso cuando otros proyectos exigen los recursos para sus actividades. Es importante tener un compromiso de la dirección, en particular con respecto a los recursos compartidos. Es probable que estas personas no puedan trabajar a tiempo completo en el despliegue.

Cuando la SAP no haya sido suministrada por un proveedor, sino que se haya desarrollado internamente, los desarrolladores correspondientes deberán participar en las actividades de implementación.

Llevar a cabo 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, es necesario abordarlo lo antes posible. Cuando la SAP se desarrolla internamente, los desarrolladores correspondientes deben contribuir a la implementación proporcionando cualquier funcionalidad que falte.
- ¿La SAP y el proceso existente se soportan mutuamente? Si no es así, es necesario alinearlas.

¹⁰ “defensor” es la traducción del término “champion”.

Evaluar el piloto

Utilizar a todos los implicados para la evaluación.

4.1.2 Despliegue

Una vez que el piloto ha sido evaluado, la SAP sólo debe ser desplegada al resto del departamento/organización si el piloto ha sido considerado un éxito. La puesta en marcha debe llevarse a cabo de forma gradual y bien gestionada. Los factores de éxito para el despliegue incluyen:

- Un despliegue incremental: Realice la puesta en marcha al resto de la organización en pasos, en incrementos. De esta manera, el soporte a los nuevos usuarios llega en "oleadas" más que en una sola vez. Esto permite que el uso de la SAP aumente en incrementos. Los posibles cuellos de botella pueden ser identificados y resueltos antes de que se conviertan en problemas reales. Se pueden añadir licencias cuando sea necesario.
- Adaptar y mejorar los procesos para adaptarlos al uso de la SAP: Cuando diferentes usuarios utilizan la SAP, diferentes procesos entran en contacto con la SAP, y necesitan adaptarse a la SAP, o la SAP puede necesitar adaptaciones (pequeñas) a los procesos.
- Proporcionar formación y orientación/asesoramiento a los nuevos usuarios: Los nuevos usuarios necesitan formación y orientación en el uso de la nueva SAP. Asegurarse de que esto esté en su lugar. Se debe proporcionar formación/talleres a los usuarios antes de que utilicen realmente la SAP.
- Definición de directrices de uso: Es posible escribir directrices, listas de comprobación y preguntas frecuentes para el uso de la SAP. Esto puede evitar extensas preguntas de soporte.
- Implementar una forma de recopilar información sobre el uso real: Debería haber una forma automatizada de recopilar información sobre el uso real de la SAP. Idealmente no sólo el uso en sí, sino también qué partes de la SAP (ciertas funcionalidades) están siendo utilizadas. De esta forma, el uso de la SAP se puede monitorizar fácilmente.
- Monitorización del uso, beneficios y costes de la SAP: La monitorización del uso de la SAP durante un cierto período de tiempo indica si la SAP se utiliza realmente. Esta información también se puede utilizar para volver a calcular el caso de negocio (por ejemplo, cuánto tiempo se ha ahorrado, cuántos problemas se han evitado).
- Proporcionar apoyo a los equipos de prueba y desarrollo de una SAP determinada.
- 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 sus aportaciones son necesarias y deseadas para mejorar el uso de la SAP.
- Identificación e implementación de mejoras: A partir de la realimentación del equipo y la monitorización de la SAP, identificar e implementar los pasos para la mejora. Asimismo, comunicar esto claramente a los implicados.

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 software que será probado por la SAP.

Normalmente, una nueva SAP o una nueva versión de la misma, se despliega ya sea al principio del proyecto o cuando se alcanza un hito, como la congelación del código o el final de una iteración (sprint en inglés). Esto se debe a que las actividades de despliegue, con todas las pruebas y modificaciones

implicadas, requieren tiempo y esfuerzo. También es una buena forma de mitigar el riesgo de que la SAP no funcione y cause interrupciones en el proceso de automatización de pruebas. Sin embargo, si hay problemas críticos que necesitan ser corregidos para la SAP o si un componente del entorno en el que se ejecuta necesita ser reemplazado, entonces el despliegue se hará independientemente de la fase de desarrollo del SSP.

4.2 Evaluación del Riesgo y Estrategias de Mitigación

Los problemas técnicos pueden conducir a riesgos de producto o a riesgos de proyecto. Los problemas técnicos más comunes incluyen:

- Demasiada abstracción puede llevar a dificultades en la comprensión de lo que realmente sucede (por ejemplo, con palabras clave).
- Guiado por datos: las tablas de datos pueden volverse demasiado grandes, complejas o complicadas.
- Dependencia de la SAP para utilizar determinadas librerías del sistema operativo u otros componentes que pueden no estar disponibles en todos los entornos objetivo del SSP.

Los riesgos de proyecto típicos del despliegue incluyen:

- Problemas de dotación de personal: puede ser difícil conseguir a las personas adecuadas para mantener el código base.
- Nuevos entregables 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.

Puntos potenciales de fallo del proyecto de la SAP incluyen:

- Migración a un entorno diferente.
- Despliegue en el entorno objetivo.
- Nueva entrega desde desarrollo.

Hay una serie de estrategias de mitigación de riesgos que pueden emplearse para abordar estas áreas de riesgo. Éstas se exponen a continuación.

La SAP tiene su propio ciclo de vida de software, tanto si se trata de una solución desarrollada internamente como si se trata de una solución adquirida. Hay que recordar que la SAP, como cualquier otro software, necesita estar bajo control de versiones y sus prestaciones documentadas. De lo contrario, se hace muy difícil desplegar diferentes partes de la misma y hacerlas trabajar juntas, o trabajar en ciertos entornos.

Además, tiene que haber un procedimiento de despliegue documentado, claro y fácil de seguir. Este procedimiento depende de la versión; por lo tanto, también debe estar bajo control de versiones.

Hay dos casos diferenciados cuando se despliega una SAP:

1. Despliegue inicial.
2. Despliegue de mantenimiento - la SAP ya existe y necesita ser objeto de mantenimiento.

Antes de comenzar con el primer despliegue de una SAP, es importante asegurarse de que puede ejecutarse en su propio entorno, de que está aislada de los cambios aleatorios y de que los casos de prueba se pueden actualizar y gestionar. Tanto la SAP como su infraestructura deben ser objeto de mantenimiento.

En el caso de un 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 paquete de pruebas que ejecutará la SAP.

Los riesgos relacionados con un primer despliegue incluyen:

- El tiempo total de ejecución del juego de pruebas puede ser superior al tiempo de ejecución planificado para el ciclo de prueba. En este caso, es importante asegurarse de que el juego de pruebas disponga del tiempo suficiente para ejecutarse por completo antes de que comience el siguiente ciclo de pruebas programado.
- Se han producido problemas de instalación y configuración en el entorno de pruebas (por ejemplo, configuración de la base de datos y carga inicial, inicio/parada de los servicios). En general, la SAP necesita una manera eficaz de establecer las precondiciones 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, tiene que evolucionar, y las actualizaciones para ello tienen que ser desplegadas en producción. Antes de desplegar una versión actualizada de la SAP en producción, es necesario que sea probada como cualquier otro software. Por lo tanto, es necesario comprobar la nueva funcionalidad, verificar que el juego de pruebas se puede ejecutar en la SAP actualizada, que se pueden emitir informes y que no hay problemas de rendimiento u otras regresiones funcionales. En algunos casos, puede ser necesario cambiar todo el juego de pruebas para adaptarlo a la nueva versión de la SAP.

Cuando se realiza un despliegue de mantenimiento, se necesitan los siguientes pasos:

- Realizar una evaluación de los cambios en la nueva versión de la SAP comparando con la versión antigua.
- Probar la SAP para las nuevas funcionalidades y las regresiones.
- Comprobar si el juego de pruebas necesita ser adaptado a la nueva versión de la SAP.

Una actualización también conlleva los siguientes riesgos y las correspondientes acciones de mitigación:

- El juego de pruebas necesita cambios para poder ejecutarse en la SAP actualizada: realizar los cambios necesarios en el juego de pruebas y probarlos antes de desplegarlos en la SAP.
- Stubs, controladores e interfaces utilizados en las pruebas deben ser objeto de cambio para que concuerden con la versión actualizada de la SAP: realizar los cambios necesarios en el arnés de prueba y probarlo antes de su despliegue en la SAP.

- La infraestructura necesita cambiar para adecuarse a la SAP actualizada: hacer una evaluación de los componentes de la infraestructura que necesitan ser modificados, realizar los cambios y probarlos con la SAP actualizada.
- La SAP actualizada tiene defectos adicionales o problemas de rendimiento: realizar un análisis de riesgos vs. beneficios. Si los problemas descubiertos hacen imposible actualizar la SAP, puede ser mejor no proceder con la actualización o esperar a una próxima versión de la SAP. Si los problemas son poco significativos en comparación con los beneficios, la SAP aún puede ser actualizada. Se debe crear una nota de entrega con los problemas conocidos para notificar a los ingenieros de automatización de pruebas y a otros implicados e intentar obtener una estimación de cuándo se van a solucionar los problemas.

4.3 Mantenimiento de la Automatización de Pruebas

El desarrollo de soluciones de automatización de pruebas no es trivial. Deben ser modulares, escalables, comprensibles, fiables y presentar capacidad de ser probadas. Para añadir aún más complejidad, las soluciones de automatización de pruebas, como cualquier otro software, tienen que 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. El mantenimiento de la SAP mediante su adaptación a los nuevos tipos de sistemas que se van a probar, la compatibilidad con nuevos entornos de software o el cumplimiento de las nuevas leyes y reglamentos contribuye a garantizar un funcionamiento fiable 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 sobre una SAP operativa existente y se activa por modificaciones, migración o retirada del sistema. Este proceso se puede estructurar en las siguientes categorías:

- **Mantenimiento preventivo:** se realizan cambios para que la SAP admita más tipos de pruebas, pruebe en múltiples interfaces, pruebe múltiples versiones del SSP o soporte 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 operaciones, por lo tanto reduciendo el riesgo en su uso, es a través de la ejecución de pruebas de mantenimiento regulares.
- **Mantenimiento perfectivo:** la SAP está optimizada y se solucionan los problemas no funcionales. Pueden abordar el rendimiento de la SAP, su usabilidad, robustez o fiabilidad.
- **Mantenimiento adaptativo:** A medida que se lanzan al mercado nuevos sistemas software (sistemas operativos, gestores de bases de datos, navegadores web, etc.), puede ser necesario que la SAP los soporte. Además, puede ser el caso de que la SAP necesite cumplir con nuevas leyes, regulaciones o requisitos específicos de la industria. En este caso, se realizan cambios en la SAP para que se adapte en consecuencia. Nota: por lo general, la conformidad con las leyes y reglamentos crea un mantenimiento obligatorio con normas, requisitos específicos y, a veces, requisitos de auditoría. Además, a medida que se actualizan las herramientas de integración y se crean nuevas versiones, los puntos terminales de integración de herramientas deben mantenerse y conservarse en funcionamiento.

4.3.2 Alcance y Enfoque

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

- El tamaño y la complejidad de la SAP.
- La magnitud del cambio.
- El riesgo del cambio.

Dado que el mantenimiento se refiere a una SAP en operaciones, es necesario un análisis de impacto para determinar cómo puede verse afectado el sistema a consecuencia de los cambios. Dependiendo del impacto, los cambios deben introducirse de forma incremental y deben llevarse a cabo pruebas después de cada paso para garantizar el funcionamiento continuo de la SAP. Nota: El mantenimiento de la SAP puede ser difícil si sus especificaciones y documentación no están actualizadas.

Dado que la eficiencia con respecto al tiempo es el principal factor que contribuye al éxito de la automatización de pruebas, es fundamental contar con buenas prácticas para mantener la SAP, incluyendo:

- Los procedimientos de despliegue y el uso de la SAP deben ser claros y estar documentados.
- Se deben documentar las dependencias de terceras partes, junto con los inconvenientes y los problemas conocidos.
- La SAP debe ser modular, de modo que sus partes puedan ser reemplazadas con facilidad.
- La SAP debe ejecutarse en un entorno reemplazable o con componentes reemplazables.
- La SAP debe separar los guiones de prueba del propio MTAP.
- La SAP se debe ejecutar aislada con respecto al entorno de desarrollo, de modo que los cambios en la SAP no afecten negativamente al entorno de prueba.
- La SAP, junto con el entorno, el juego de pruebas y los artefactos de los productos de prueba deben estar bajo gestión de configuración.

También hay consideraciones para el mantenimiento de los componentes de terceros y otras librerías como se indica a continuación:

- Muy a menudo es el caso de que la SAP utilice componentes de terceros para ejecutar las pruebas. También puede ocurrir que la SAP dependa de librerías de terceros (por ejemplo, las librerías de automatización de la interfaz de usuario). Todos los componentes de terceros de la SAP deben estar documentados y bajo gestión de configuración.
- Es necesario tener un plan en caso de que estos componentes externos necesiten ser modificados o reparados. La persona responsable del mantenimiento de la SAP necesita saber a quién contactar o dónde remitir un problema.
- Debe haber documentación sobre la licencia bajo la cual se utilizan los componentes de terceros, para que haya información sobre si pueden ser modificados, 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 librerías de terceros es una acción preventiva que compensa la inversión en el largo plazo.

Entre las consideraciones para las normas de nomenclatura y otras convenciones se incluyen:

- La idea de los estándares de nomenclatura y otras convenciones tiene una razón simple: el juego de pruebas y la propia SAP tiene que ser fácil de leer, entender, modificar y mantener. Esto

ahorra tiempo en el proceso de mantenimiento y también minimiza el riesgo de introducir regresiones o correcciones erróneas que, de otro modo, podrían evitarse fácilmente.

- Es más fácil introducir nuevas personas en el proyecto de automatización de pruebas cuando se utilizan convenciones de nomenclatura estándar.
- Los estándares de nomenclatura pueden referirse a variables y archivos, escenarios de prueba, palabras clave y parámetros de palabras clave. Otras convenciones se refieren a prerrequisitos y post-acciones para la ejecución de la prueba, el contenido de los datos de prueba, el entorno de prueba, el estado de ejecución de la prueba y los registros e informes de ejecución.
- Todos los estándares y convenciones deben ser acordados y documentados al iniciar un proyecto de automatización de pruebas.

Las consideraciones sobre la documentación incluyen:

- La necesidad de una documentación buena y actualizada tanto para los escenarios de prueba como para la SAP es bastante clara, pero hay dos cuestiones relacionadas con esto: alguien tiene que escribirla y alguien tiene que mantenerla.
- Mientras que el código de la herramienta de prueba puede ser autodocumentado o documentado de forma semiautomática, todos los diseños, componentes, integraciones con terceros, dependencias y procedimientos de despliegue deben ser documentados por alguien.
- Es una buena práctica introducir la elaboración de documentación como parte del proceso de desarrollo. Una tarea no debería considerarse realizada a menos que esté documentada o que la documentación esté actualizada.

Las consideraciones sobre el material de formación incluyen:

- Si la documentación de la SAP está bien redactada, puede utilizarse como base para el material de formación de la SAP.
- El material de formación es una combinación de especificaciones funcionales de la SAP, diseño y arquitectura de la SAP, despliegue y mantenimiento de la SAP, uso de la SAP (manual del usuario), ejemplos prácticos y ejercicios, y consejos y trucos.
- El mantenimiento del material de formación consiste en una redacción inicial y su revisión periódica. En la práctica, lo hacen los miembros del equipo designados como formadores en la SAP y lo más probable es que ocurra hacia el final de una iteración del ciclo de vida del SSP (al final de las iteraciones, sprints en inglés, por ejemplo).

5. Información y Métricas de Automatización de Pruebas - 165 minutos

Palabras Clave

cobertura, densidad de defectos del código de automatización, esfuerzo de prueba manual equivalente, gestión de información de la prueba, matriz de trazabilidad, métrica, registro en bitácora de la prueba

Objetivos de Aprendizaje para Información y Métricas de Automatización de Pruebas

5.1 Selección de Métricas para la SAP

NA-IAP-5.1.1 (K2) Clasificar las métricas que se pueden utilizar para monitorizar la estrategia y efectividad de la automatización de pruebas.

5.2 Implementación de la Medición

NA-IAP-5.2.1 (K3) Implementar métodos de recopilación de métricas para soportar requisitos técnicos y de gestión. Explicar cómo se puede implementar la medición de la automatización de pruebas.

5.3 Registro de la SAP y del SSP

NA-IAP-5.3.1 (K4) Analizar los datos del registro de la prueba tanto de la SAP como del SSP.

5.4 Información de la Automatización de Pruebas

NA-IAP-5.4.1 (K2) Explicar cómo se construye y publica el informe de ejecución de una prueba.

5.1 Selección de Métricas para la SAP

Esta sección se centra en las métricas que se pueden utilizar para monitorizar la estrategia de automatización de pruebas así como la eficacia y eficiencia de la SAP. Éstas 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. El Jefe de Pruebas general del proyecto se encarga de seleccionarlas. Las métricas de automatización de pruebas permiten al GAP y al IAP realizar un seguimiento del progreso hacia los objetivos de automatización de pruebas y supervisar 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 las que se utilizan para medir la efectividad y eficiencia de la SAP en la consecución de sus objetivos.

Las métricas de la SAP medidas, en general, incluyen las siguientes:

- Métricas externas de la SAP:
 - Beneficios de la automatización.
 - Esfuerzo para construir pruebas automatizadas.
 - Esfuerzo para analizar incidencias de pruebas automatizadas.
 - Esfuerzo para mantener las pruebas automatizadas.
 - Proporción de fallos con respecto a los defectos.
 - Tiempo para ejecutar pruebas automatizadas.
 - Número de casos de prueba automatizados.
 - Número de resultados paso y fallo.
 - Número de resultados de falsos fallo y falsos paso.
 - Cobertura de código.
- Métricas internas de la SAP:
 - Métricas de herramientas de guion.
 - Densidad de defectos del código de automatización.
 - Velocidad y eficiencia de los componentes de la SAP.

A continuación se describen cada una de ellas.

Beneficios de la automatización

Es particularmente importante medir e informar sobre las ventajas de una SAP. Esto se debe a que los costes (en términos del número de personas involucradas en un período de tiempo determinado) son fáciles de ver. Las personas que trabajan fuera de las pruebas podrán hacerse una idea del coste global, pero es posible que no vean los beneficios obtenidos.

Cualquier medición del 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 frecuencia de ejecución), o alguna otra ventaja como una mayor repetibilidad, mayor uso de recursos, o menos errores manuales. Las posibles mediciones incluyen:

- Número de horas de esfuerzo de prueba manual ahorradas.
- Reducción del tiempo para ejecutar 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 con respecto al conjunto completo de casos de prueba (aunque los casos de prueba automatizados no pueden compararse fácilmente con los 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 medio de los defectos encontrados anteriormente, éste puede "calcularse" sumando los costes evitados).
- Número de defectos detectados a causa de la SAP que no se habrían detectado mediante pruebas manuales (por ejemplo, defectos de fiabilidad).

Hay que tener en cuenta que, por lo general, la automatización de las pruebas ahorra esfuerzo manual de prueba. Este esfuerzo se puede dedicar a otros tipos de pruebas (manuales) (por ejemplo, pruebas exploratorias). Los defectos detectados por estas pruebas adicionales también pueden verse como beneficios indirectos de la SAP, ya que la automatización de las pruebas permitió ejecutar estas pruebas manuales. Sin la SAP estas pruebas no se habrían realizado y, posteriormente, no se habrían encontrado defectos adicionales.

Esfuerzo para construir pruebas automatizadas

El esfuerzo para automatizar las pruebas es uno de los costes clave asociados con la automatización de pruebas. Esto es, a menudo, más que el coste de realizar la misma prueba manualmente y, por lo tanto, puede ir en detrimento de la expansión del uso de la automatización de pruebas. Mientras que el coste de implementar una prueba automatizada específica dependerá en gran medida de la prueba misma, otros factores como el enfoque de guion 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 las pruebas más extensas o más complejas suelen tardar más tiempo en ser automatizadas que las pruebas cortas o simples, el cálculo del coste de construcción¹¹ para la automatización de pruebas puede basarse en un tiempo medio de construcción. Esto puede refinarse aún más si se considera el coste medio de un conjunto específico de pruebas, como las que se refieren a la misma función o a un nivel de prueba determinado. Otro enfoque es expresar el costo de construcción como un factor del esfuerzo requerido para ejecutar la prueba manualmente (esfuerzo de prueba manual equivalente, EPME). Por ejemplo, puede ser que se necesite el doble del esfuerzo de una prueba manual para automatizar un caso de prueba, o dos veces el EPME.

Esfuerzo para analizar los fallos del SSP

El análisis de los fallos en el SSP detectados a través de la ejecución de pruebas automatizadas puede ser significativamente más complejo que en el caso de una prueba ejecutada manualmente, ya que los eventos que conducen al fallo de una prueba manual suelen ser conocidos por el probador que realiza la prueba. Esto puede mitigarse como se describe en el nivel de diseño en el Capítulo 3.1.4 y en el nivel de información en los Capítulos 5.3 y 5.4. Esta medición puede expresarse como una media por caso

¹¹ "construcción" es la traducción del término "build".

de prueba fallido 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.

Los registros disponibles del SSP y de la SAP desempeñan un papel crucial en el análisis de los fallos. Los registros deben proporcionar suficiente información para realizar este análisis de manera eficiente. Entre las características importantes de registro se incluyen:

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

El SSP, por otra parte, debe registrar todas las acciones que se realizan (independientemente de si la acción es el resultado de una prueba manual o automática). Cualquier error interno debe ser registrado y deben estar disponibles todos los volcados de memoria y trazas de pila.

Esfuerzo para mantener las pruebas automatizadas

El esfuerzo de mantenimiento necesario para mantener las pruebas automatizadas sincronizadas con el SSP puede ser muy significativo y, en última instancia, puede superar los beneficios logrados por la SAP. Esta ha sido la causa del fracaso de muchos esfuerzos de automatización. Por lo tanto, la monitorización del esfuerzo de mantenimiento es importante para hacer evidente cuándo hay que 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 versión del SSP. También pueden expresarse como una media por cada prueba automatizada actualizada o como un factor de EPME.

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

Cuando se conoce (o se puede calcular) el esfuerzo de mantenimiento para las pruebas automatizadas, esta información puede desempeñar un papel crucial a la hora de decidir si implementar o no una determinada funcionalidad o corregir un determinado defecto. El esfuerzo necesario para mantener el caso de prueba debido a cambios en el software debe considerarse con el cambio del SSP.

Proporción de fallos con respecto a los defectos.

Un problema común con las pruebas automatizadas es que muchas de ellas pueden fallar por la misma razón: un solo defecto en el software. Mientras que el propósito de las pruebas es hacer evidentes los defectos en el software, tener más de una prueba poniendo en evidencia el mismo defecto es un desperdicio. Este es, en particular, el caso de las pruebas automatizadas ya que el esfuerzo requerido para analizar cada prueba fallida puede ser significativo. La medición del número de pruebas automatizadas que fallan para un defecto dado puede ayudar a identificar dónde puede haber un problema. La solución subyace en el diseño de las pruebas automatizadas y su selección para la ejecución.

Tiempo para ejecutar pruebas automatizadas

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

Número de casos de prueba automatizados

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

Número de resultados paso y fallo

Esta es una métrica común y hace un seguimiento de cuántas pruebas automatizadas pasaron y cuántas fallaron en lograr el resultado esperado. Los fallos se deben analizar para determinar si el fallo se debió a un defecto en el SSP o se debió a factores externos como un problema con el entorno o con la propia SAP.

Número de resultados de falsos fallo y falsos paso

Como se ha visto 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 mantenga bajo. 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 paso falso, se produce un fallo en el SSP, pero no se identifica por la automatización de la prueba, por lo que se informa de 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 de forma adecuada, porque se utilizó un oráculo de prueba no válido o porque el caso de prueba esperaba un resultado incorrecto.

Se debe tener en cuenta que las falsas alarmas pueden ser causadas por defectos en el código de prueba (véase la métrica "Densidad de defectos del código de automatización"), pero también pueden ser causadas por un SSP inestable que se está comportando de forma impredecible (por ejemplo, tiempo de espera). Los anzuelos de prueba también pueden causar falsas alarmas debido al nivel de intrusión que están causando.

Cobertura de 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, por ejemplo, la cobertura de código del conjunto de pruebas de regresión. No existe un porcentaje absoluto que indique una cobertura adecuada, y el 100% de la cobertura de código es inalcanzable en cualquier otra cosa que no sea la

más simple de las aplicaciones de software. Sin embargo, en general se admite que una mayor cobertura es mejor, ya que reduce el riesgo general en el despliegue de software. Esta métrica también puede indicar actividad en el SSP. Por ejemplo, si la cobertura de código disminuye, lo más probable es que se haya añadido funcionalidad al SSP, pero que no se haya añadido el caso de prueba correspondiente al juego de pruebas automatizado.

Métricas de herramientas de guion

Hay muchas métricas que se pueden utilizar para monitorizar el desarrollo de guiones de automatización. La mayoría de ellas son similares a las métricas de código fuente para el SSP. Las líneas de código (LOC por sus siglas en inglés) y la complejidad ciclomática pueden utilizarse para resaltar guiones demasiado grandes o complejos (lo que sugiere que es necesario volver a diseñarlos).

La proporción entre comentarios y sentencias ejecutables se puede utilizar para dar una posible indicación del alcance de la documentación del guion y los comentarios. El número de no conformidades con los estándares de codificación de guiones puede dar una indicación del grado de cumplimiento de dichos estándares.

Densidad de defectos del código de automatización

El código de automatización no difiere del código del SSP en que es 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 éstos debe ser monitorizado por métricas tales como la densidad de defectos del código. Estos serán más fáciles de recopilar con el apoyo de un sistema de gestión de la configuración.

Velocidad y eficiencia de los componentes de la SAP

Las diferencias en el tiempo que se requiere para realizar los mismos pasos de prueba en el mismo entorno pueden indicar un problema en el SSP. Si el SSP no realiza la misma funcionalidad en el mismo tiempo transcurrido, es necesario realizar una investigación. Esto puede indicar una variabilidad en el sistema que no es aceptable y que podría empeorar con un aumento de la carga. La SAP necesita un rendimiento lo suficientemente bueno para no obstaculizar el rendimiento del SSP. Si el rendimiento es un requisito crítico para el SSP, entonces la SAP debe diseñarse de forma tal que lo tenga en cuenta.

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 de informar que el valor de una medición en un momento específico. Por ejemplo, saber que el coste medio de mantenimiento por cada prueba automatizada que requiere mantenimiento es superior al de las dos versiones anteriores del SSP puede dar lugar a la adopción de medidas para determinar la causa del aumento y tomar medidas para invertir la tendencia.

El coste de la medición debe ser lo más bajo posible y esto puede lograrse a menudo automatizando la recogida y la presentación de informes.

5.2 Implementación de la Medición

Dado que una estrategia de automatización de pruebas tiene productos de prueba automatizados en su núcleo, los productos de prueba automatizados pueden ser mejorados para registrar información sobre su uso. Cuando la abstracción se combina con el producto de prueba estructurado, cualquier

mejora hecha al producto de prueba subyacente puede ser utilizada por todos los guiones de prueba automatizados de nivel superior. Por ejemplo, la mejora de un producto de prueba subyacente para registrar las horas de inicio y fin de la ejecución de una prueba puede aplicarse a todas las pruebas.

Prestaciones de la automatización que soportan la generación de mediciones e informes

Los lenguajes de guion de muchas herramientas de prueba soportan la medición y la generación de informes a través de facilidades que pueden utilizarse para grabar y registrar información antes, durante y después de la ejecución de pruebas individuales, conjuntos de pruebas y un juego completo de pruebas.

La información sobre cada una de una serie de ejecuciones de pruebas necesita disponer de una prestación de análisis que tenga en cuenta los resultados de las ejecuciones de prueba anteriores para poder resaltar las tendencias (como los cambios en el porcentaje de éxito de la prueba).

La automatización de las pruebas, normalmente, requiere la automatización tanto de la ejecución de la prueba como de la verificación de la prueba, esto último se consigue comparando elementos específicos del resultado de la prueba con un resultado esperado predefinido. Por lo general, la mejor manera de realizar esta comparación es mediante una herramienta de prueba. Se debe tener en cuenta el nivel de información que se transmite como resultado de esta comparación. Es importante que el estado de la prueba se determine de forma correcta (por ejemplo, paso, fallo). En el caso de un estado de fallo, se requerirá más información sobre la causa del fallo (por ejemplo, capturas de pantalla).

Distinguir entre las diferencias esperadas en el resultado real y esperado de una prueba no siempre es trivial, aunque el soporte de herramientas puede ayudar en gran medida a definir comparaciones que ignoren las diferencias esperadas (como fechas y horas) al tiempo que resaltan cualquier diferencia inesperada.

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

Cuando la información de la ejecución de casos de prueba automatizados se utiliza en otras herramientas (para el seguimiento y la presentación de informes, por ejemplo, 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 a través de la funcionalidad de la herramienta de prueba existente (formatos de exportación de informes) o mediante la creación de informes personalizados que se generan en un formato coherente con otros programas (".xls" para Excel, ".doc" para Word, ".html" para Web, etc.).

Visualización de resultados (cuadros de mando, diagramas, gráficos, etc.)

Los resultados de las pruebas deben ser visibles en gráficos. Se debe considerar el uso de colores para indicar problemas en la ejecución de la prueba, tales como semáforos para indicar el progreso de la ejecución / automatización de la prueba, de manera que las decisiones se puedan tomar en base a información comunicada. La dirección 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, se puede profundizar en los detalles.

5.3 Registro de la SAP y del SSP

El registro es muy importante en la SAP, incluyendo el registro tanto para la propia automatización de pruebas como para el SSP. Los registros de pruebas constituyen una fuente que se utiliza con

frecuencia para analizar problemas potenciales. En la siguiente sección hay ejemplos de registro de pruebas, clasificados por SAP o SSP.

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

- Qué caso de prueba se está ejecutando actualmente, incluyendo la hora de inicio y la hora final.
- El estado de la ejecución del caso de prueba porque, si bien los fallos se pueden identificar fácilmente en los archivos de registro, el propio marco de trabajo también debe tener esta información y debe informar a través de un panel de control. El estado de ejecución del caso de prueba puede ser de paso, fallo o error de la SAP. El resultado de error de la SAP se utiliza en situaciones en las que el problema no está en el SSP.
- Detalles del registro de la prueba a alto nivel (registro de pasos significativos) incluyendo información relativa a los tiempos.
- Información dinámica sobre el SSP (por ejemplo, fugas de memoria) que el caso de prueba pudo identificar con la ayuda de herramientas de terceros. Los resultados y fallos reales de estas mediciones dinámicas se deben registrar con el caso de prueba que se estaba ejecutando cuando se detectó el incidente.
- En el caso de pruebas de fiabilidad / pruebas de estrés (donde se realizan numerosos ciclos) se debe registrar un contador, para poder determinar fácilmente cuántas veces se han ejecutado casos de prueba.
- Cuando los casos de prueba tienen partes aleatorias (por ejemplo, parámetros aleatorios o pasos aleatorios en pruebas de máquinas de estado), se debe registrar el número/las opciones aleatorias.
- Todas las acciones que realiza un caso de prueba deben registrarse de tal manera que el archivo de registro (o partes del mismo) pueda reproducirse para volver a ejecutar la prueba con exactamente los mismos pasos y el mismo tiempo. Esto es útil para comprobar la reproducibilidad de un fallo identificado y para capturar información adicional. La información de la acción del caso de prueba también se puede registrar en el propio SSP para utilizarla al reproducir problemas identificados por el cliente (el cliente ejecuta el escenario, la información del registro se captura y el equipo de desarrollo puede volver a reproducirla durante el diagnóstico del problema).
- Capturas de pantalla y otras capturas visuales pueden ser guardadas durante la ejecución de la prueba para su uso posterior durante el análisis del fallo.
- Cuando un caso de prueba detecta un fallo, la SAP debe asegurarse de que toda la información necesaria para analizar el problema está disponible/almacenada, así como cualquier información relativa a la continuación de la prueba, si procede. La SAP debe almacenar en un lugar seguro cualquier volcado de memoria y trazas de pila asociados. También cualquier archivo de registro que pueda sobrescribirse (memorias intermedias cíclicas se utilizan a menudo para archivo de registro en el SSP) debe copiarse a esta ubicación, donde estarán disponibles para su análisis posterior.
- El uso de colores puede ayudar a distinguir diferentes tipos de información registrada (por ejemplo, errores en rojo, información de progreso en verde).

Registro del SSP:

- Cuando el SSP identifica un problema, se debe registrar toda la información necesaria para analizar el problema, incluidos fecha y hora de registro, la ubicación del origen del problema, mensajes de error, etc.

- El SSP puede registrar todas las interacciones 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 pueden ser analizados adecuadamente, y el equipo de desarrollo puede tratar de reproducir el problema.
- Al iniciar el sistema, la información de la configuración se debe registrar en un archivo, que comprende las diferentes versiones del software/firmware, la configuración del SSP, la configuración del sistema operativo, etc.

Toda la información de los diferentes registros debe poder ser fácilmente consultable. Un problema identificado en el archivo de registro por la SAP 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 Información de la Automatización de Pruebas

Los registros de prueba proporcionan información detallada sobre los pasos de ejecución, las acciones y las respuestas de un caso de prueba y/o juego de pruebas. Sin embargo, los registros por sí solos no pueden proporcionar un buen resumen del resultado global de la ejecución. Para ello, es necesario disponer de una funcionalidad de informes. Después de cada ejecución del juego de pruebas, se debe generar y publicar un informe conciso. Para ello se podría utilizar un componente generador de informes reutilizable.

Contenido de los informes

El informe de ejecución de pruebas debe contener un resumen que ofrezca una visión general de los resultados de la ejecución, del sistema que se está probando y del entorno en el que se realizaron las pruebas, que sea adecuado para cada uno de los implicados.

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

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

Publicación de los informes

El informe debe ser publicado para todos los interesados en los resultados de la ejecución. Puede ser subido a un sitio web, enviado a una lista de correo o subido a otra herramienta como una herramienta de gestión de pruebas. Desde un punto de vista práctico, lo más probable es que los interesados en el resultado de la ejecución lo miren y analicen si tienen la posibilidad de suscribirse y recibir el informe por correo electrónico.

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

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

Palabras Clave

pruebas de confirmación, pruebas de regresión

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

6.1 Criterios para la Automatización

NA-IAP-6.1.1 (K3) Aplicar criterios para determinar la idoneidad de las pruebas para la automatización.

NA-IAP-6.1.2 (K2) Comprender los factores de la transición de pruebas manuales a la automatización de pruebas.

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

NA-IAP-6.2.1 (K2) Explicar los factores que hay que considerar al implementar pruebas de regresión automatizadas.

6.3 Factores que Hay que Tener en Cuenta al Implementar la Automatización dentro de Pruebas de Nuevas Prestaciones

NA-IAP-6.3.1 (K2) Factores que hay que tener en cuenta al implementar la automatización dentro de pruebas de nuevas prestaciones.

6.4 Factores que Hay que Tener en Cuenta al Implementar la Automatización de las Pruebas de Confirmación

NA-IAP-6.4.1 (K2) Explicar los factores que hay que tener en cuenta al implementar la automatización de las pruebas de confirmación.

6.1 Criterios para la Automatización

Tradicionalmente, las organizaciones han desarrollado casos de prueba manuales. Al decidir migrar hacia un entorno de pruebas automatizadas, se debe evaluar el estado actual de las pruebas manuales y determinar el enfoque más eficaz para automatizar estos activos de pruebas. La estructura existente de una prueba manual puede o no ser adecuada para la automatización, en cuyo caso puede ser necesaria una nueva redacción completa de la prueba para soportar la automatización. Alternativamente, los componentes relevantes de las pruebas manuales existentes (por ejemplo, valores de entrada, resultados esperados, camino de navegación) se pueden extraer de las pruebas manuales existentes y reutilizar para la automatización. Una estrategia de pruebas manuales que tenga en cuenta la automatización permitirá realizar pruebas cuya estructura facilite la migración a la automatización.

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

Tenga en cuenta también que ciertos tipos de pruebas sólo pueden ejecutarse (eficazmente) de forma automatizada, por ejemplo, pruebas de fiabilidad, pruebas de esfuerzo 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. Aunque este tipo de casos de prueba también pueden ejecutarse manualmente (utilizando comandos introducidos manualmente para activar las interfaces), esto podría 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 pruebas automatizadas, uno necesita considerar la aplicabilidad y viabilidad de crear pruebas automatizadas vs. manuales. Los criterios de adecuación pueden incluir, pero no se limitan a:

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

Cada uno de estos puntos se explica con más detalle a continuación.

Frecuencia de uso

La frecuencia con la que se debe realizar una prueba es una consideración en cuanto a la viabilidad de automatizar o no. Las pruebas que se ejecutan más regularmente, como parte de un ciclo de entrega mayor o menor, son mejores candidatos para la automatización, ya que se utilizarán con frecuencia. Por 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, pueden utilizarse en versiones 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 por sus siglas en inglés) y mitigación de riesgos para la base de código existente.

Si se ejecuta un guion de prueba una vez al año, y el SST cambia dentro del año, puede no ser factible o eficiente crear una prueba automatizada. El tiempo que podría requerir la adaptación anual de la prueba para ajustarse al SST podría hacerse mejor de forma manual.

Complejidad para automatizar

En los casos en los que un sistema complejo necesita ser probado, puede haber un gran beneficio a partir de la automatización para ahorrarle al probador manual la difícil tarea de tener que repetir pasos complejos que son tediosos, consumen mucho tiempo y son propensos a cometer errores.

Sin embargo, ciertos guiones de prueba pueden ser difíciles o no rentables de automatizar. Hay un rango de factores que pueden afectar esto, incluyendo: un SST que no es compatible con las soluciones de pruebas automatizadas disponibles; el requisito de producir una cantidad sustancial de código y desarrollar llamadas a la Interfaz de Programación de Aplicaciones (API por sus siglas en inglés) para automatizar; la multiplicidad de sistemas que necesitan ser tratados como parte de la ejecución de una prueba; la interacción con interfaces externas y/o sistemas propietarios; algunos aspectos de las pruebas de usabilidad; la cantidad de tiempo que se necesita para probar los guiones de automatización, etc.

Compatibilidad y soporte de herramientas

Existe un amplio rango de plataformas de desarrollo utilizadas para crear aplicaciones. El desafío para el probador es saber qué herramientas de prueba hay disponibles (si las hubiera) para soportar cualquier plataforma dada, y hasta qué punto la plataforma está soportada. Las organizaciones utilizan una variedad de herramientas de prueba, incluyendo las de proveedores comerciales, de código abierto y desarrolladas internamente. Cada organización tendrá necesidades y recursos diferentes para dar soporte a las herramientas de prueba. Los proveedores comerciales suelen proporcionar apoyo mediante pago y, en el caso de los líderes del mercado, por lo general cuentan con un ecosistema de expertos que pueden ayudar en la implementación de la herramienta de prueba. Las herramientas de código abierto pueden ofrecer soporte, como foros en línea, en los que los usuarios pueden obtener información y formular preguntas. Las herramientas de prueba desarrolladas internamente dependen del personal existente para proporcionar soporte.

La cuestión de la compatibilidad de las herramientas de prueba no debe subestimarse. Embarcarse en un proyecto de automatización de pruebas sin entender 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 hacerlo.

Madurez del proceso de prueba

Para implementar eficazmente la automatización dentro de un proceso de prueba, ese proceso debe ser estructurado, disciplinado y repetible. La automatización aporta un proceso de desarrollo completo al proceso de pruebas existente que requiere la gestión del código de automatización y los componentes relacionados.

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

Un SSP tiene un ciclo de vida de producto que puede extenderse de años a décadas. A medida que comienza el desarrollo de un sistema, el sistema cambia y se expande para tratar los defectos y añadir 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 pruebas automatizadas. A medida que se optimizan y mejoran los diseños de pantalla y los controles, la creación de automatización en un entorno que cambia dinámicamente puede requerir un trabajo de repaso continuo, que no es eficiente ni efectivo. Esto sería similar a tratar de cambiar una llanta en un vehículo en movimiento; es mejor esperar a que el vehículo se detenga. Para sistemas grandes en un entorno de desarrollo secuencial, cuando un sistema se ha estabilizado e incluye funcionalidad principal, este es el mejor momento para comenzar la implementación de pruebas automatizadas.

Con el tiempo, los sistemas llegan al final de su ciclo de vida de producto y se retiran o se rediseñan para utilizar tecnologías más nuevas y eficientes. La automatización no se recomienda para un sistema que se acerca al final de su ciclo de vida, ya que no tendrá mucho valor emprender una iniciativa tan efímera. Sin embargo, para los sistemas que se están rediseñando utilizando una arquitectura diferente pero preservando la funcionalidad existente, un entorno de pruebas automatizado que defina los elementos de datos será igualmente útil en los sistemas antiguos y nuevos. En este caso, sería posible la reutilización de los datos de prueba y sería necesaria la recodificación del entorno automatizado para que fuera compatible con la nueva arquitectura.

Sostenibilidad del entorno

Un entorno de pruebas 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 y corregir rápidamente problemas con la automatización, la facilidad con la que se pueden mantener los componentes de automatización y la facilidad con la que se pueden añadir nuevas prestaciones y soporte en el entorno automatizado. Estos atributos son una parte integral del diseño general y la implementación de la AAPg.

Controlabilidad del SSP (precondiciones, configuración y estabilidad)

El IAP debe identificar características de control y visibilidad en el SSP que contribuyan a la creación de pruebas automatizadas eficaces. De lo contrario, la automatización de pruebas depende únicamente de las interacciones de la interfaz de usuario, lo que resulta en una solución de automatización de pruebas menos mantenible. Para obtener más información, consulte la Sección 2.3 sobre Diseño para Capacidad de ser Probado y Automatización.

Planificación técnica para soporte del análisis del Retorno de la Inversión¹²

La automatización de pruebas puede proporcionar diversos grados de beneficio a un equipo de pruebas. Sin embargo, un nivel significativo de esfuerzo y coste está asociado con la implementación de una solución de pruebas automatizada efectiva. Antes de incurrir en una inversión de tiempo y esfuerzo para desarrollar pruebas automatizadas, se debe realizar una evaluación para valorar cuál podría ser el beneficio y el resultado global previsto y potencial de la implementación de la automatización de pruebas. Una vez determinado lo anterior, se deben definir las actividades necesarias para llevar a cabo dicho plan y se deben determinar los costes asociados para calcular el Retorno sobre la Inversión.

Para prepararse adecuadamente para la transición a un entorno automatizado, es necesario abordar las siguientes áreas:

- Disponibilidad de herramientas en el entorno de pruebas para la automatización de pruebas.
- Corrección de los datos de prueba y casos de prueba.
- Alcance del esfuerzo de automatización de pruebas.
- Educación del equipo de prueba para el cambio de paradigma.
- Roles y responsabilidades.
- Cooperación entre desarrolladores e ingenieros de automatización de pruebas.
- Esfuerzo paralelo.
- Informes de automatización de pruebas.

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

Es necesario que las herramientas de prueba seleccionadas se instalen y se confirme que funcionan en el entorno del laboratorio de pruebas. Esto puede implicar la descarga de paquetes de servicios o actualizaciones de versiones, la selección de la configuración de instalación adecuada (incluidos los complementos) necesaria para soportar el SSP y asegurar que la SAP funciona correctamente en el entorno de laboratorio de pruebas frente al entorno de desarrollo de automatización.

¹² "retorno de la inversión" es la traducción del término "return of investment" o "ROI" por sus siglas en inglés.

Corrección de los datos de prueba y casos de prueba

Es necesario que los datos de las pruebas manuales y los casos de prueba sean correctos y completos para asegurar que el uso con la automatización proporcione resultados predecibles. Las pruebas ejecutadas bajo automatización necesitan datos explícitos para la entrada, navegación, sincronización y validación.

Alcance del esfuerzo de automatización de pruebas

Con el fin de demostrar un éxito temprano en la automatización y obtener información sobre cuestiones técnicas que pueden afectar al avance, comenzar con un alcance limitado facilitará las tareas de automatización futuras. Un proyecto piloto puede concentrarse en un área de la funcionalidad de un sistema que sea representativa de la interoperabilidad global del sistema. Las lecciones aprendidas del piloto contribuirán a ajustar las estimaciones y los calendarios futuros y a determinar las áreas que requieren recursos técnicos especializados. Un proyecto piloto proporciona una forma rápida de demostrar el éxito temprano de la automatización, lo que refuerza aún más el apoyo de la gestión.

Para ayudar en este aspecto, los casos de prueba que deben automatizarse se deben seleccionar con sensatez. Seleccionar aquellos casos que requieran poco esfuerzo de automatización, pero que aporten un alto valor añadido. Se pueden implementar regresiones automáticas o pruebas de humo y añadir un valor considerable, ya que estas pruebas normalmente se ejecutan con bastante frecuencia, incluso diariamente. Otro buen candidato para empezar es la prueba de fiabilidad. Estas pruebas, a menudo, se componen de pasos y se ejecutan una y otra vez, revelando problemas que son difíciles de detectar de forma manual. Estas pruebas de fiabilidad requieren poco esfuerzo de implementación, pero pueden mostrar un valor añadido muy pronto.

Estos proyectos piloto ponen la automatización en el centro de atención (ahorro en el esfuerzo de prueba manual, o identificación de problemas relevantes) y allanan 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 el mayor valor inicialmente. Sin embargo, en este contexto, es importante que, como parte de un esfuerzo piloto, se eviten las pruebas técnicamente más difíciles de automatizar. De lo contrario, se dedicará demasiado esfuerzo a desarrollar la automatización con muy pocos resultados que mostrar. Como regla general, la identificación de pruebas que compartan características con una gran parte de la aplicación proporcionará el impulso necesario para mantener vivo el esfuerzo de automatización.

Formación del equipo de prueba para el cambio de paradigma

Los probadores se presentan en muchos sabores: algunos son expertos en la materia y provienen de la comunidad de usuarios finales o están involucrados como analistas de negocio, mientras que otros tienen fuertes competencias técnicas que les permite entender mejor la arquitectura subyacente del sistema. Para que las pruebas sean efectivas, es preferible una amplia combinación de antecedentes. A medida que el equipo de pruebas avance hacia la automatización, los roles se especializarán más. Cambiar la composición del equipo de pruebas es esencial para que la automatización tenga éxito, y educar al equipo desde el principio sobre el cambio deseado ayudará a reducir la ansiedad sobre los roles o la posible idea de ser prescindible. Cuando se aborda correctamente, el cambio hacia la automatización debería entusiasmar a todos los miembros del equipo de pruebas y prepararlos para participar en el cambio organizativo y técnico.

Roles y responsabilidades

La automatización de pruebas debe ser una actividad en la que todos puedan participar. Sin embargo, esto no significa que todos deban desempeñar el mismo rol. Diseñar, implementar y mantener un entorno de pruebas automatizadas es de naturaleza técnica y, como tal, debe estar reservado a

personas con sólidos conocimientos de programación y trayectoria técnica. Los resultados de un esfuerzo de desarrollo de pruebas automatizadas deben ser un entorno que pueda ser utilizado por personas tanto técnicas como no técnicas por igual. Con el fin de maximizar el valor de un entorno de pruebas automatizadas, es necesario contar con personas con conocimientos especializados en el dominio de pruebas y competencias en esta materia, ya que será necesario desarrollar guiones de prueba adecuados (incluidos los datos de prueba correspondientes). Éstos se utilizarán para controlar el entorno automatizado y proporcionar una cobertura de pruebas específica. Los expertos en el dominio revisan los informes para confirmar la funcionalidad de la aplicación, mientras que los expertos técnicos se aseguran de que el entorno automatizado funcione correcta y eficientemente. Estos expertos técnicos también pueden ser desarrolladores con interés en las pruebas. La experiencia en el desarrollo de software es esencial para diseñar software que sea mantenible, y esto es de suma importancia en la automatización de pruebas. Los desarrolladores pueden concentrarse en el marco de trabajo de automatización de pruebas o en las librerías de prueba. La implementación de los casos de prueba debe permanecer en manos de probadores.

Cooperación entre desarrolladores e ingenieros de automatización de pruebas

El éxito de la automatización de pruebas también requiere la participación del equipo de desarrollo de software así como de los probadores. Los desarrolladores y los probadores tendrán que trabajar mucho más estrechamente para automatizar las pruebas, de modo que los desarrolladores puedan proporcionar personal de soporte e información técnica sobre sus métodos y herramientas de desarrollo. Los ingenieros de automatización de pruebas pueden plantear dudas sobre la capacidad de ser probado del diseño del sistema y el código del desarrollador. Este será el caso, en particular, si no se siguen los estándares o si los desarrolladores utilizan librerías/objetos extraños, propios o incluso muy nuevos. Por ejemplo, los desarrolladores pueden elegir un control de Interfaz Gráfica de Usuario (GUI por sus siglas en inglés) de terceros que puede no ser compatible con la herramienta de automatización seleccionada. Por último, la dirección de proyectos de una organización debe tener una noción clara de los tipos de roles y responsabilidades que se requieren para un esfuerzo de automatización de éxito.

Esfuerzos paralelos

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

En muchos casos, se realizará una evaluación de los guiones manuales antes de la conversión a la automatización. Como resultado de esa evaluación, podría determinarse que es necesario reestructurar los guiones de prueba manuales existentes para adoptar un enfoque más eficiente y eficaz en el marco de la automatización.

Informes de automatización

Hay varios informes que pueden ser generados de forma automática por una SAP. Éstos incluyen el estado de paso/fallo de guiones individuales o pasos dentro de un guion, las estadísticas generales de ejecución de pruebas y el desempeño general de la SAP. Es igualmente importante tener visibilidad en la operación correcta de la SAP para que cualquier resultado específico de la aplicación que sea informado pueda ser considerado exacto y completo (Ver 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 ofrecen una gran oportunidad para utilizar la automatización. Una cama de pruebas de regresión crece a medida que las pruebas funcionales de hoy se convierten en las pruebas de regresión de mañana. Es sólo 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 pruebas manuales tradicionales.

En el desarrollo de los pasos para prepararse para automatizar las pruebas de regresión, hay que plantearse una serie de preguntas:

- ¿Con qué frecuencia se deben ejecutar las pruebas?
- ¿Cuál es el tiempo de ejecución para cada prueba, para el juego de regresión?
- ¿Hay solapamiento funcional entre las pruebas?
- ¿Las pruebas comparten datos?
- ¿Las pruebas dependen unas de otras?
- ¿Qué precondiciones se necesitan antes de la ejecución de la prueba?
- ¿Qué porcentaje de cobertura del SSP representan las pruebas?
- ¿Actualmente se ejecutan las pruebas sin fallos?
- ¿Qué debe suceder cuando la ejecución de las pruebas de regresión requiere un tiempo excesivo?

Cada uno de estos puntos 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 candidatos para la automatización. Estas pruebas ya se han desarrollado, practican una funcionalidad conocida del SSP y su tiempo de ejecución se reducirá de forma muy considerable gracias al uso de la automatización.

Tiempo de ejecución de la prueba

El tiempo que se tarda en ejecutar una prueba determinada o un juego de pruebas completo 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 con pruebas que consumen mucho tiempo. Esto permitirá que cada prueba se ejecute de forma más rápida y eficiente, a la vez que se añaden ciclos adicionales de ejecución de pruebas de regresión automatizadas. El beneficio es la realimentación adicional y más frecuente sobre la calidad del SSP y la reducción del riesgo de despliegue.

Solapamiento funcional

Al automatizar las pruebas de regresión existentes, es una buena práctica identificar cualquier solapamiento funcional que pudiera haber entre los casos de prueba y, en la medida de lo posible, reducir ese solapamiento en la prueba automatizada equivalente. Esto proporcionará mayores eficiencias en el tiempo de ejecución de las pruebas automatizadas, que serán significativas a medida que se ejecuten más y más casos de pruebas automatizados. A menudo, las pruebas desarrolladas utilizando la automatización adoptarán una nueva estructura ya que dependen de componentes reutilizables y repositorios de datos compartidos. No es raro descomponer las pruebas manuales existentes en varias pruebas automatizadas más pequeñas. Asimismo, la consolidación de varias pruebas manuales en una prueba automatizada más extensa puede ser la solución adecuada. Las pruebas manuales necesitan ser evaluadas individualmente, y como grupo, para que se pueda desarrollar una estrategia de conversión efectiva.

Compartir datos

A menudo, las pruebas comparten datos. Esto puede ocurrir cuando las pruebas utilizan el mismo registro de datos para ejecutar diferentes funcionalidades 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" podría verificar a qué cursos asistió el empleado como parte de sus objetivos de desarrollo profesional. Cada caso de prueba utiliza el mismo empleado, pero verifica parámetros diferentes. En un entorno de pruebas manuales, los datos de los empleados se suelen duplicar muchas veces a través de cada caso de prueba manual que verifica los datos de los empleados utilizando este empleado. Sin embargo, en una prueba automatizada, los datos que se comparten se deben, siempre que sea posible y factible, almacenar y acceder a ellos desde una sola fuente para evitar la duplicación o la introducción de errores.

Interdependencia de las pruebas

Cuando se ejecutan escenarios de prueba de regresión complejos, una prueba puede tener una dependencia de una o más pruebas. Esta situación puede ser bastante común y puede ocurrir, por ejemplo, como resultado de un nuevo "ID de Pedido" que se crea como resultado de un paso de prueba. Las pruebas subsiguientes pueden querer verificar que: a) la nueva orden se muestra correctamente en el sistema, b) son posibles cambios en la orden, o c) el borrado de la orden es correcto. En cada caso, el valor "ID de Pedido" que se crea de forma dinámica en la primera prueba debe ser capturado para su reutilización en pruebas posteriores. Esta situación se puede abordar en función del diseño de la SAP.

Precondiciones de prueba

A menudo no se puede ejecutar una prueba antes de establecer las condiciones iniciales. Estas condiciones pueden incluir la selección de la base de datos correcta o el conjunto de datos de prueba desde el cual realizar la prueba, o el establecimiento de valores o parámetros iniciales. Muchos de

estos pasos de inicialización son necesarios para determinar que una precondition de prueba puede ser automatizada. Esto permite una solución más fiable y sólida cuando estos pasos no se pueden omitir antes de la ejecución de las pruebas. A medida que las pruebas de regresión se convierten a la automatización, estas condiciones deben ser parte del proceso de automatización.

Cobertura del SSP

Cada vez que se ejecutan las pruebas, se ejercita o practica parte de la funcionalidad de un SSP. Con el objetivo de determinar la calidad general del SSP, es necesario diseñar pruebas que tengan la cobertura más amplia y profunda. Además, las herramientas de cobertura de código pueden ser utilizadas 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 pruebas adicionales proporcionen una cobertura adicional. Medir esto proporciona un medio efectivo para cuantificar el valor de las propias pruebas.

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 asegurar una conversión exitosa a una prueba de regresión automatizada. Si la prueba manual no se ejecuta de forma correcta, ya sea porque está mal redactada, utiliza datos no válidos, no está actualizada o no está sincronizada con el SSP actual, o como resultado de un defecto del SSP, convertirla a la automatización antes de comprender y/o resolver la causa raíz del fallo creará una prueba automatizada que no funciona, lo cual es un desperdicio e improductivo.

Grandes juegos de pruebas de regresión

El conjunto de pruebas de regresión para un SST puede llegar a ser bastante grande, tan grande que el conjunto de pruebas no se puede ejecutar completamente en una noche o durante un fin de semana. En ese caso, la ejecución concurrente de casos de prueba es una posibilidad si se dispone de varios SST's (para aplicaciones de PC esto probablemente no plantea un problema, pero cuando el SST consiste en un avión o un cohete espacial esta es una historia diferente). Los SST's 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 sólo partes de la prueba de regresión. Con el tiempo (semanas) se ejecutará el juego completo, eventualmente. La elección de qué parte del conjunto de pruebas de regresión se ejecutará también puede basarse en un análisis de riesgos (¿qué partes del SST se han modificado últimamente?).

6.3 Factores que Hay que Tener en Cuenta al Implementar la Automatización dentro de Pruebas de Nuevas Prestaciones

En general, es más fácil automatizar casos de prueba para nuevas funcionalidades, ya que la implementación aún no ha finalizado (o mejor aún: no se ha iniciado). El ingeniero de pruebas puede utilizar sus conocimientos para explicar a los desarrolladores y arquitectos qué es exactamente lo que se necesita en la nueva funcionalidad, de modo que la solución de automatización de pruebas pueda probarla de forma eficaz y eficiente.

A medida que se introducen nuevas prestaciones en un SST, se requiere que los probadores desarrollen nuevas pruebas contra estas nuevas prestaciones y los requisitos correspondientes. El IAP debe solicitar realimentación por parte de los diseñadores de pruebas especializados en el dominio y determinar si la SAP actual satisfará 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 en relación con los componentes de los productos de prueba automatizados existentes, de modo que los cambios o ampliaciones estén totalmente documentados y no afecten al comportamiento (o rendimiento) de la funcionalidad de la SAP existente.

Si se implementa una nueva prestación con, por ejemplo, una clase diferente de objeto, puede ser necesario realizar actualizaciones o ampliaciones de los componentes de los productos de prueba. Además, se deberá evaluar la compatibilidad con las herramientas de prueba existentes y, si fuera necesario, identificar soluciones alternativas. Por ejemplo, si se utiliza un enfoque basado en palabras clave, puede ser necesario desarrollar palabras clave adicionales o modificar o ampliar las palabras clave existentes para adaptarse a la nueva funcionalidad.

Puede haber un requisito para evaluar herramientas de prueba adicionales para dar soporte al nuevo entorno bajo el cual opera la nueva funcionalidad. Por ejemplo, puede ser necesaria una nueva herramienta de pruebas si la herramienta de pruebas existente sólo soporta HTML.

Los nuevos requisitos de prueba pueden afectar a las pruebas automatizadas y a los componentes de los productos de prueba existentes. Por lo tanto, antes de hacer cualquier cambio, las pruebas automatizadas existentes se deben ejecutar contra 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 las interdependencias con otras pruebas. Cualquier nuevo cambio en la tecnología requerirá la evaluación de los componentes actuales de los productos de prueba (incluidas las herramientas de prueba, las librerías de funciones, las Interfaces de Programación de Aplicaciones, etc.) y la compatibilidad con la SAP existente.

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

Finalmente, es necesario determinar si la SAP existente continuará satisfaciendo las necesidades actuales del SSP. ¿Siguen siendo válidas las técnicas de implementación, o se requiere una nueva arquitectura, y esto se puede hacer ampliando la capacidad actual?

Cuando se introducen nuevas funcionalidades, los ingenieros de pruebas tienen la oportunidad de asegurarse de que las nuevas funcionalidades definidas presenten capacidad de ser probadas. Durante la fase de diseño, las pruebas se deben tener en cuenta al planificar el suministro de interfaces de prueba que se puedan utilizar mediante lenguajes de guion o la herramienta de automatización de pruebas para verificar la nueva funcionalidad. Para obtener más información, consulte la Sección 2.3, Diseño para Capacidad de ser Probado y Automatización.

6.4 Factores que Hay que Tener en Cuenta al Implementar la Automatización de las Pruebas de Confirmación

Las pruebas de confirmación se realizan después de una corrección de código que resuelve un defecto notificado. Normalmente, un probador sigue los pasos necesarios para replicar el defecto y verificar que el defecto ya no está presente.

Los defectos tienen una forma de volver a introducirse en versiones posteriores (esto puede indicar un problema de gestión de la configuración) y, por lo tanto, las pruebas de confirmación son los principales candidatos para la automatización. El uso de la automatización ayudará a reducir el tiempo de ejecución de las pruebas de confirmación. La prueba de confirmación se puede añadir y complementar la cama de prueba de regresión automatizada existente.

La prueba de confirmación automatizada tiene, por lo general, un alcance limitado en términos de funcionalidad. La implementación puede ocurrir en cualquier momento una vez que un defecto es notificado y se comprenden los pasos necesarios para reproducirlo. Las pruebas de confirmación automatizadas pueden incorporarse a un juego de regresión automatizado estándar o, cuando sea conveniente, incluirse en las pruebas automatizadas existentes. Con cualquiera de los dos enfoques, el valor de automatizar las pruebas de confirmación de defectos sigue siendo válido.

El seguimiento automatizado de las pruebas de confirmación permite la presentación de informes adicionales sobre el tiempo y el número de ciclos invertidos en la resolución de defectos.

Además de las pruebas de confirmación, es necesario realizar pruebas de regresión para garantizar que no se han introducido nuevos defectos como efecto secundario de la corrección del defecto. Puede ser necesario realizar un análisis de impacto para determinar el alcance adecuado de las pruebas de regresión.

7. Verificación de la SAP - 120 minutos

Palabras Clave

verificación

Objetivos de Aprendizaje para Verificación de la SAP

7.1 Verificación de los Componentes del Entorno de Pruebas Automatizadas

NA-IAP-7.1.1 (K3) Verificar la corrección de un entorno de prueba automatizado, incluida la configuración de la herramienta de prueba.

7.2 Verificación del Juego de Pruebas Automatizadas

NA-IAP-7.2.1 (K3) Verificar el comportamiento correcto de un determinado guion de prueba automatizado y/o juego de pruebas.

7.1 Verificación de los Componentes del Entorno de Pruebas

El equipo de automatización de pruebas necesita verificar que el entorno de pruebas automatizadas funciona como se espera. Estas comprobaciones se realizan, por ejemplo, antes de iniciar las pruebas automatizadas.

Hay una serie de pasos que se pueden tomar para verificar los componentes del entorno de pruebas automatizadas. A continuación se explica cada uno de ellos con más detalle:

Instalación, preparación¹³, configuración y personalización de herramientas de prueba

La SAP se compone de numerosos componentes. Se debe tener en cuenta cada uno de estos aspectos para garantizar un rendimiento fiable y repetible. En el núcleo de una SAP se encuentran los componentes ejecutables, las librerías funcionales correspondientes y los archivos de datos y configuración de soporte. El proceso de configuración de una SAP puede comprender desde el uso de guiones de instalación automatizados hasta la colocación manual de archivos en las carpetas correspondientes. Las herramientas de prueba, al igual que los sistemas operativos y otras aplicaciones, suelen tener paquetes de servicio o pueden tener complementos opcionales o necesarios para garantizar la compatibilidad con cualquier entorno dado del SSP.

La instalación automatizada (o copia) desde un repositorio central tiene ventajas. Se puede garantizar que se han realizado pruebas en diferentes SSP's con la misma versión de la SAP, y la misma configuración de la SAP, cuando sea conveniente. Las actualizaciones de la SAP se pueden realizar a través del repositorio. El uso del repositorio y el proceso de actualización a una nueva versión de la SAP debería ser el mismo que para las herramientas de desarrollo estándar.

Guiones de prueba con resultados de paso y fallo conocidos

Cuando fallan casos de prueba que se sabe que han pasado con anterioridad, es inmediatamente evidente que algo está básicamente mal y debe corregirse lo antes posible. Por el contrario, cuando los casos de prueba presentan un resultado pasado a pesar de que deberían haber fallado, es necesario identificar el componente que no funcionó de forma correcta. Es importante verificar la correcta generación de los archivos de registro y las métricas de rendimiento, así como la preparación y el desmantelamiento automatizados del caso/guion de prueba. También es útil ejecutar algunas pruebas de los diferentes tipos y niveles de pruebas (pruebas funcionales, pruebas de rendimiento, pruebas de componentes, etc.). Esto también debería hacerse a nivel del marco de trabajo.

Repetibilidad en la configuración/desmantelamiento del entorno de prueba

Se implementará una SAP sobre una variedad de sistemas y servidores. Para garantizar que la SAP funcione correctamente en cada entorno, es necesario tener un enfoque sistemático de la carga y descarga de la SAP para cualquier entorno determinado. Esto se logra con éxito cuando la construcción y reconstrucción de la SAP no proporciona una diferencia discernible en la forma en que opera dentro y a través de múltiples entornos. La gestión de la configuración de los componentes de la SAP asegura que se puede crear una configuración determinada de forma fiable.

Configuración del entorno de prueba y de los componentes

¹³ "preparación es la traducción del término "setup".

Comprender y documentar los diversos componentes que conforman la SAP proporciona el conocimiento necesario sobre qué aspectos de la SAP pueden verse afectados o requerir cambios cuando cambia el entorno del SSP.

Conectividad con sistemas/interfaces internos y externos

Una vez instalado una SAP en el entorno de un SSP determinado, y antes de su uso real frente a un SSP, se debe aplicar un conjunto de comprobaciones o precondiciones para asegurar que la conectividad con sistemas internos y externos, interfaces, etc., está disponible. Establecer las precondiciones para la automatización es esencial para asegurar que la SAP se ha instalado y configurado correctamente.

Nivel de intrusión de las herramientas de prueba automatizadas

A menudo, la SAP estará estrechamente acoplada al SSP. Esto es así por diseño para que haya un alto nivel de compatibilidad, especialmente en lo que se refiere a las interacciones de nivel Interfaz Gráfica de Usuario. 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 que cuando se utiliza manualmente; el rendimiento del SSP se ve afectado por la SAP en el entorno o cuando se ejecuta la SAP contra el SSP.

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

- Cuando se interactúa 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 a efectos de la prueba. El enfoque de la prueba no influye en el comportamiento ni en los tiempos del SSP. Interactuar con el SSP de esta manera puede ser muy complejo. Es posible que sea necesario hardware dedicado, idiomas de descripción de hardware para interactuar con el SSP, etc. En el caso de sistemas sólo software, no se trata de un enfoque típico, pero en el caso de productos con software embebido, este enfoque es más común.
- Cuando se interactúa con el SSP en el nivel de Interfaz Gráfica de Usuario (GUI), el entorno del SSP se adapta para inyectar comandos de Interfaz de Usuario (UI) y extraer la información necesaria para los casos de prueba. El comportamiento del SSP no se modifica directamente, pero los tiempos se ven afectados, lo que puede tener un impacto en el comportamiento. El nivel de intrusión es mayor que en el punto anterior, pero la interacción con el SSP de esta manera es menos compleja. A menudo se pueden utilizar herramientas comerciales de distribución masiva (COTS por sus siglas en inglés) para este tipo de automatización.
- La interacción con el SSP se puede realizar a través de interfaces de prueba en el software o utilizando interfaces existentes ya proporcionadas por el software. La disponibilidad de estas interfaces (APIs por sus siglas en inglés) es una parte importante del diseño para la capacidad de ser probado. El nivel de intrusión puede ser bastante alto en este caso. Las pruebas automatizadas utilizan interfaces que pueden no ser utilizadas en absoluto por los usuarios finales del sistema (interfaces de prueba) o las interfaces pueden utilizarse en un contexto diferente al del mundo real. Por otro lado, es muy fácil y económico realizar pruebas automatizadas a través de interfaces de programación de aplicación (API por sus siglas en inglés). La prueba del SSP a través de interfaces de prueba puede ser un enfoque sólido, siempre y cuando se comprenda el riesgo potencial.

Un alto nivel de intrusión puede presentar fallos durante las pruebas que no se observan en condiciones reales de uso. Si esto causa fallos en las pruebas automatizadas, la confianza en la solución de automatización de pruebas puede verse reducida de forma drástica. Los desarrolladores pueden

requerir que los fallos identificados por las pruebas automatizadas se reproduzcan primero manualmente, si es posible, para ayudar en su análisis.

Pruebas de Componentes del Marco de Trabajo

Al igual que cualquier proyecto de desarrollo de software, los componentes del marco de trabajo automatizado necesitan ser probados y verificados individualmente. Esto puede incluir pruebas funcionales y no funcionales (rendimiento, utilización de recursos, usabilidad, etc.).

Por ejemplo, los componentes que proporcionan verificación de objetos en sistemas de interfaz gráfica de usuario deben probarse para una amplia gama de clases de objeto con el fin de establecer que la verificación de objetos funciona correctamente. Del mismo modo, los registros e informes de errores deben producir información precisa sobre el estado de la automatización y el comportamiento del SSP.

Entre los ejemplos de pruebas no funcionales se pueden incluir la comprensión de la degradación del rendimiento del marco de trabajo y la utilización de recursos del sistema que pueden indicar problemas tales como fugas de memoria. Interoperabilidad de componentes dentro y/o fuera del marco de trabajo.

7.2 Verificación del Juego de Pruebas Automatizadas

Los juegos de pruebas automatizadas deben probarse para comprobar su completitud, consistencia y comportamiento correcto. Se pueden aplicar diferentes tipos de comprobaciones de verificación para asegurarse de que el juego de pruebas automatizadas se encuentre en funcionamiento en cualquier momento, o para determinar si es apto para su uso.

Hay una serie de pasos que se pueden seguir para verificar el juego de pruebas automatizadas. Estos incluyen:

- Ejecución de guiones de prueba con resultados de paso y fallo conocidos.
- Comprobación del juego de pruebas.
- Verificación de nuevas pruebas que se centran en las nuevas prestaciones del marco de trabajo.
- Analizar la repetibilidad de las pruebas.
- Comprobar que hay suficientes puntos de verificación en el juego de pruebas automatizadas.

A continuación se explica cada uno de estos puntos con más detalle.

Ejecución de guiones de prueba con resultados de paso y fallo conocidos

Cuando fallan casos de prueba que se sabe que han tenido un resultado de paso, es muy evidente que algo está básicamente mal y debe corregirse lo antes posible. Por el contrario, cuando un juego de pruebas pasa aunque debería haber fallado, es necesario identificar el caso de prueba que no funcionó correctamente. Es importante verificar la correcta generación de los archivos de registro, los datos de rendimiento, la configuración y el desarmado del caso/guion de prueba. También es útil ejecutar algunas pruebas de los diferentes tipos y niveles de pruebas (pruebas funcionales, pruebas de rendimiento, pruebas de componentes, etc.).

Comprobación del juego de pruebas

Comprobar la integridad del juego de pruebas (todos los casos de prueba tienen resultados esperados, datos de prueba disponibles), y la versión correcta con respecto al marco de trabajo y al SSP.

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

La primera vez que se utiliza una nueva prestación de la SAP en casos de prueba, se debería verificar y monitorizar de cerca para asegurar que la prestación funciona correctamente.

Analizar la repetibilidad de las pruebas

Cuando se repiten pruebas, el resultado/veredicto de la prueba debe ser siempre el mismo. Tener casos de prueba en el conjunto de pruebas que no proporcionen un resultado fiable (por ejemplo, condiciones de carrera¹⁴) se podrían separar del juego de pruebas automatizadas activo y analizarse por separado para encontrar la causa raíz. De lo contrario, se invertirá tiempo de forma reiterada en estas pruebas para analizar el problema.

Los fallos intermitentes necesitan ser analizados. El problema puede estar en el propio caso de prueba o en el marco de trabajo (o incluso puede ser un problema en el SST). El análisis del archivo de registro (del caso de prueba, el marco de trabajo y el SST) puede identificar la causa raíz del problema. También puede ser necesario depurar. Puede ser necesario el soporte del analista de pruebas, el desarrollador de software y el experto en el dominio para encontrar la causa raíz.

¹⁴ “condiciones de carrera” es la traducción del término “race conditions”.

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

Debe ser posible verificar que el juego de pruebas automatizadas se ha ejecutado y ha alcanzado los resultados esperados. Se deberán aportar evidencias para asegurar que el juego de pruebas y/o los casos de prueba se han 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 poscondiciones, etc.



8. Mejora Continua - 150 minutos Referencias

Palabras Clave

mantenimiento

Objetivos de Aprendizaje para Mejora Continua

Opciones para Mejorar la Automatización de Pruebas

NA-IAP-8.1.1 (K4) Analizar los aspectos técnicos de una solución de automatización de pruebas implementada y ofrecer recomendaciones para mejorarla.

Adaptación de la Automatización de Pruebas al Entorno y a Cambios del SSP

NA-IAP-8.2.1 (K4) Analizar el software de pruebas automatizadas, incluidos los componentes del entorno de pruebas, las herramientas y las librerías de funcionalidades de soporte, con el fin de comprender dónde se deben realizar la consolidación y las actualizaciones de acuerdo con un conjunto determinado de entornos de prueba o cambios en el 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, normalmente hay muchas oportunidades para mejorar la SAP. Las mejoras de la SAP pueden llevarse a cabo para lograr una serie de beneficios, entre los que se incluyen una mayor eficiencia (reduciendo aún más la intervención manual), una mayor facilidad de uso, capacidades adicionales y un mejor apoyo a las actividades de prueba. La decisión de cómo mejorar la SAP se verá condicionada por los beneficios que aportarán el mayor valor añadido a un proyecto.

Las áreas específicas de una SAP que se pueden considerar para mejorar incluyen la creación de guiones, la verificación, la arquitectura, el preprocesamiento y el posprocesamiento, la documentación y el soporte de herramientas. Estas áreas se describen con más detalle a continuación.

Guiones

Los enfoques basados en lenguajes de guion varían desde el enfoque estructurado simple de los enfoques basados en datos hasta los enfoques basados en palabras clave más sofisticados, como se describe en la Sección 3.2.2. Puede ser conveniente actualizar el actual enfoque de guion de la SAP para todas las nuevas pruebas automatizadas. El enfoque podrá adaptarse a todas las pruebas automatizadas existentes o, al menos, a aquellas que supongan un mayor esfuerzo de mantenimiento.

En lugar de cambiar por completo el enfoque basado en guion, las mejoras de la SAP se pueden concentrar en la implementación de guiones. Por ejemplo:

- Evaluar el solapamiento de casos/pasos/procedimientos de prueba en un esfuerzo por consolidar las pruebas automatizadas.

Los casos de prueba que contienen secuencias de acciones similares no deberían implementar estos pasos varias veces. Estos pasos deben convertirse en una función y añadirse a una librería, para que puedan ser reutilizados. Estas funciones de librería¹⁵ pueden ser utilizadas por diferentes casos de prueba. Esto aumenta la mantenibilidad del producto de prueba. Cuando los pasos de la prueba no son idénticos sino 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 casos de prueba, la SAP debería 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 ser capaz de realizar las acciones de recuperación necesarias en el SSP (por ejemplo, un reinicio del SSP completo).

- Evaluar los mecanismos de espera para asegurar de que se está utilizando el mejor tipo. Hay tres mecanismos de espera habituales:
 1. Las esperas incrustadas en el código¹⁶ (esperar un cierto número de milisegundos) pueden ser la causa raíz de muchos problemas de automatización de pruebas.
 2. La espera dinámica por sondeo, por ejemplo, comprobar si se ha producido un determinado cambio de estado o una determinada acción, es mucho más flexible y eficaz:
 - Espera sólo el tiempo necesario y no se pierde tiempo de prueba.
 - Cuando por alguna razón el proceso toma más tiempo, el sondeo simplemente esperará hasta que la condición sea verdadera. Se debe recordar incluir un mecanismo

¹⁵ “funciones de librería” es la traducción de “library functions”.

¹⁶ “espera incrustada en el código” es la traducción del término “hard-coded wait”.

de tiempo de espera, de lo contrario la prueba puede esperar para siempre en caso de un problema.

3. Una forma aún mejor es suscribirse al mecanismo de eventos del SSP. Esto es mucho más fiable que las otras dos opciones, pero el lenguaje de guiones de prueba debe admitir la suscripción a eventos y el SSP debe ofrecer estos eventos a la aplicación de prueba. Se debe recordar incluir un mecanismo de tiempo de espera, de lo contrario la prueba puede esperar para siempre en caso de un problema.

- Tratar el producto de prueba como software.

El desarrollo y mantenimiento de los productos de prueba es sólo una forma de desarrollo de software. Como tales, se deben aplicar buenas prácticas de codificación (por ejemplo, el uso de directrices de codificación, análisis estático, revisiones de código). Incluso puede ser una buena idea usar desarrolladores de software (en lugar de ingenieros de prueba) para desarrollar ciertas partes del producto de prueba (por ejemplo, librerías).

- Evaluar los guiones existentes para su revisión/eliminación.

Varios guiones pueden ser problemáticos (por ejemplo, fallar de vez en cuando, o con altos costes de mantenimiento), y puede ser acertado volver a diseñar estos guiones. Otros guiones de prueba pueden ser eliminados del juego porque ya no aportan ningún valor añadido.

Ejecución de la Prueba

Cuando un juego de pruebas de regresión automatizadas no se termina en una noche, no debería ser una sorpresa. Cuando la prueba se prolonga demasiado tiempo, puede ser necesario realizarla de forma concurrente en diferentes sistemas, pero esto no siempre es posible. Cuando se utilizan sistemas objetivo¹⁷ de alto coste para las pruebas, puede ser una restricción que todas las pruebas se realicen en un único objetivo. Puede ser necesario dividir el conjunto de pruebas de regresión en varias partes, cada una de las cuales se ejecutará en un período de tiempo definido (por ejemplo, en una sola noche). Un análisis más detallado de la cobertura de la prueba automatizada puede revelar que hay duplicación. Eliminar la duplicación puede reducir el tiempo de ejecución y aumentar la eficiencia. Un análisis más detallado de la cobertura de la prueba automatizada puede revelar que hay duplicación. Eliminar la duplicación puede reducir el tiempo de ejecución y aumentar la eficiencia.

¹⁷ "sistema objetivo" es la traducción del término "target system".

Verificación

Antes de crear nuevas funciones de verificación, es conveniente adoptar un conjunto de métodos de verificación estándar para todas las pruebas automatizadas. Esto evitará que se vuelvan a implementar las acciones de verificación a través de múltiples pruebas. Cuando los métodos de verificación no son idénticos sino similares, el uso de la parametrización ayudará a que la función se pueda utilizar en múltiples tipos de objetos.

Arquitectura

Puede ser necesario cambiar la arquitectura para dar soporte a mejoras de la capacidad de ser probado del SSP. Estos cambios se pueden realizar 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 inversiones en el SSP/la SAP. Por ejemplo, si el SSP va a ser modificado para aportar Interfaces de Programación de Aplicaciones para las pruebas, entonces la SAP también debería ser refactorizada en consecuencia. Agregar este tipo de características en una etapa posterior puede ser bastante costoso; es mucho mejor pensar en esto al comienzo de la automatización (y en las etapas iniciales del desarrollo del SSP - ver Sección 2.3 Diseño para Capacidad de ser Probado y Automatización).

Preprocesamiento y postprocesamiento

Proporcionar tareas de preparación y desarmado estándar. También se conocen como preprocesamiento (preparación) y postprocesamiento (desarmado). Esto ahorra las tareas que se implementan repetidamente para cada prueba automatizada, no sólo reduciendo los costes de mantenimiento, sino también el esfuerzo necesario para implementar nuevas pruebas automatizadas.

Documentación

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

Prestaciones de la SAP

Añadir prestaciones y funciones adicionales a la SAP, como informes detallados, registros, integración con otros sistemas, etc. Añadir nuevas prestaciones sólo cuando se vayan a utilizar. Si se añaden prestaciones que no se utilizan, sólo aumenta la complejidad y disminuye la fiabilidad y la mantenibilidad.

Actualizaciones y mejoras de la SAP

Al actualizar¹⁸ o mejorar¹⁹ a nuevas versiones de la SAP, pueden aparecer nuevas funciones que pueden ser utilizadas por los casos de prueba (o pueden corregirse fallos). El riesgo es que la actualización del marco de trabajo (por la actualización de las herramientas de prueba existentes o introducción de otras nuevas) pueda tener un impacto negativo en los casos de prueba existentes. Probar la nueva versión de la herramienta de prueba ejecutando pruebas de muestra antes de desplegar la nueva versión. La muestra de las pruebas debe ser representativa 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 de Pruebas

Los cambios en una SAP existente requieren una planificación e investigación cuidadosas. Se ha invertido mucho esfuerzo en la creación de una SAP robusta que consiste en un MTAP y librerías de componentes. Cualquier cambio, por trivial que sea, puede tener un gran impacto en la fiabilidad y el rendimiento de la SAP.

Identificar los cambios en los componentes del entorno de prueba

Evaluar qué cambios y mejoras se necesitan llevar a cabo. ¿Esto requiere cambios en el software de prueba, librerías de funciones personalizadas, sistema operativo? Cada uno de ellos tiene un impacto en el desempeño de la SAP. El objetivo general es asegurar que las pruebas automatizadas continúen ejecutándose de manera eficiente. Los cambios deben hacerse de forma gradual para que el impacto en la SAP pueda medirse a través de una ejecución limitada de guiones de prueba. Una vez que se encuentra que no existe un efecto perjudicial, los cambios pueden ser implementados de forma completa. La ejecución de una regresión completa es el paso final hacia la validación de que el cambio no afectó negativamente a los guiones automatizados. Durante la ejecución de estos guiones de regresión, se pueden detectar errores. Identificar la causa raíz de estos errores (a través de informes, registros, análisis de datos, etc.) proporcionará un medio para asegurar que no son el resultado de la actividad de mejora de la automatización.

Aumentar la eficiencia y la eficacia de las librerías de funciones básicas de la SAP.

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

Concentrarse en múltiples funciones que actúen en el mismo tipo de control para la consolidación

Una gran parte de lo que ocurre durante una ejecución de una prueba automatizada es la interrogación de los controles en la Interfaz Gráfica de Usuario. Este interrogatorio sirve para proporcionar información sobre ese control (por ejemplo, visible/no visible, habilitado/no habilitado, tamaño y

¹⁸ “actualizar” es la traducción del término “update”.

¹⁹ “mejorar” es la traducción del término “upgrade” cuando se encuentra junto a término “update”. En otros casos, “update” y “upgrade” se pueden traducir como “actualizar”.

dimensiones, datos, etc.). Con esta información, una prueba automatizada puede seleccionar un elemento de una lista desplegable, introducir 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 sólo funcione en las listas desplegables. Alternativamente, puede haber una función (o puede crearse y utilizarse una dentro de la SAP) que funcione con varias funciones especificando una función como uno de sus parámetros. Por lo tanto, un IAP puede utilizar varias funciones que pueden ser consolidadas en menos funciones, logrando los mismos resultados y minimizando la necesidad de mantenimiento.

Refactorizar la AAP para acomodar los cambios en el SSP

A lo largo de la vida de una SAP, será necesario realizar cambios para adaptarse a los cambios en el SSP. A medida que el SSP evoluciona y madura, la AAP subyacente tendrá que evolucionar también para asegurar que la capacidad está disponible para soportar al SSP. Se debe tener cuidado cuando se extienden las prestaciones de modo que no se implementen como elementos accesorios y secundarios, sino que se analicen y modifiquen a nivel arquitectónico de la solución automatizada. Esto asegurará que como la nueva funcionalidad del SSP requiere guiones adicionales, habrá componentes compatibles para adaptarse a estas nuevas pruebas automatizadas.

Convenciones y estandarización de nomenclatura

A medida que se introducen cambios, las convenciones de nomenclatura para las nuevas librerías de código y funciones de automatización deben ser coherentes con las normas previamente definidas (véase la Sección 4.3.2 Alcance y Enfoque).

Evaluación de los 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 de continuidad. Por ejemplo, si ciertas pruebas son complejas y su ejecución lleva mucho tiempo, descomponerlas en varias pruebas más pequeñas puede ser más práctico y eficiente. La selección de pruebas que se ejecutan con poca o ninguna frecuencia para su eliminación reducirá la complejidad de la SAP y aportará mayor claridad a lo que debe ser objeto de mantenimiento.

9. Referencias

9.1 Estándares

Los estándares para la automatización de pruebas incluyen pero no se limitan a:

- La notación Testing and Test Control Notation (TTCN-3) del ETSI (European Telecommunication Standards Institute) e ITU (International Telecommunication Union) está compuesta por:
 - ES 201 873-1: TTCN-3 Core Language
 - ES 201 873-2: TTCN-3 Tabular Presentation Format (TFT)
 - ES 201 873-3: TTCN-3 Graphical Presentation Format (GFT)
 - ES 201 873-4: TTCN-3 Operational Semantics
 - ES 201 873-5: TTCN-3 Runtime Interface (TRI)
 - ES 201 873-6: TTCN-3 Control Interface (TCI)
 - ES 201 873-7: Using ASN.1 with TTCN-3
 - ES 201 873-8: Using IDL with TTCN-3
 - ES 201 873-9: Using XML with TTCN-3
 - ES 201 873-10: TTCN-3 Documentation
 - ES 202 781: Extensions: Configuration and Deployment Support
 - ES 202 782: Extensions: TTCN-3 Performance and Real-Time Testing
 - ES 202 784: Extensions: Advanced Parameterization
 - ES 202 785: Extensions: Behaviour Types
 - ES 202 786: Extensions: Support of interfaces with continuous signals
 - ES 202 789: Extensions: Extended TRI
- The Automatic Test Markup Language (ATML) by IEEE (Institute of Electrical and Electronics Engineers) consisting of
 - IEEE Std 1671.1: Test Description
 - IEEE Std 1671.2: Instrument Description
 - IEEE Std 1671.3: UUT Description
 - IEEE Std 1671.4: Test Configuration Description
 - IEEE Std 1671.5: Test Adaptor Description
 - IEEE Std 1671.6: Test Station Description
 - IEEE Std 1641: Signal and Test Definition
 - IEEE Std 1636.1: Test Results
- The ISO/IEC/IEEE 29119-3

- The UML Testing Profile (UTP) by OMG (Object Management Group) specifying test specification concepts for
 - Test Architecture
 - Test Data
 - Test Behavior
 - Test Logging
 - Test Management

9.2 Documentos ISTQB

<u>Identificador</u>	<u>Referencia</u>
ISTQB-AL-TM	ISTQB Certified Tester, Advanced Level Syllabus, Test Manager, Versión 2012, disponible en [ISTQB-Web]
ISTQB-AL-TTA	ISTQB Certified Tester, Advanced Level Syllabus, Technical Test Analyst, Versión 2012, disponible en [ISTQB-Web]
ISTQB-EL-CEP	ISTQB Advanced Level Certification Extension, disponible en [ISTQB-Web]
ISTQB-EL-Modules	ISTQB Advanced Level Modules Overview, Versión 1.2, 23 de Agosto de 2013, disponible en [ISTQB-Web]
ISTQB-EL-TM	ISTQB Advanced Level – Test Management syllabus, Versión 2011, disponible en [ISTQB-Web]
ISTQB-FL	ISTQB Foundation Level Syllabus, Versión 2011, disponible en [ISTQB-Web]
ISTQB-Glossary	ISTQB Glossary of terms, Versión 2.4, 4 de Julio de 2014, disponible en [ISTQB-Web].

9.3 Marcas Registradas

En este documento se utilizan las siguientes marcas registradas y marcas de servicio:
ISTQB® es una marca registrada de International Software Testing Qualifications Board.

9.4 Referencias bibliográficas

<u>Identificador</u>	<u>Referencia del libro</u>
-----------------------------	------------------------------------

Versión 2016

Página 100 de 104

21 de octubre de 2016

© International Software Testing Qualifications Board

- [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 Web

Identificador

Referencia

ISTQB-Web

Sitio web del International Software Testing Qualifications Board. Refiérase a este sitio web para el último Glosario y programas de estudio del ISTQB.
www.istqb.org

10. Aviso a los Proveedores de Formación

10.1 Tiempos de Formación

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

Es posible que los proveedores de formación dediquen más tiempo del indicado y que los candidatos vuelvan a dedicar más tiempo a la lectura e investigación. El desarrollo de un curso no tiene que seguir el mismo orden que el programa de estudio. No es necesario realizar el curso en un bloque de tiempo continuo.

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

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

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

10.2 Ejercicios Prácticos en el Área de Trabajo (Curso)

No se han definido ejercicios que se puedan realizar en el lugar de trabajo.

10.3 Normas para Aprendizaje Electrónico

Todas las partes de este plan de estudio se consideran adecuadas para su implementación como aprendizaje electrónico.

11. Índice terminológico



El índice terminológico se desarrollará en una próxima versión de este programa de estudio.

