



TEST WORKFLOW

SQS TestWORKFLOW es una herramienta diseñada para asistir durante todo el proceso de validación de software, ayudando al usuario en la automatización de las pruebas y la gestión de requisitos, tests y documentación.

Esta herramienta integra la definición y gestión de requisitos, la organización del proceso de testing, la ejecución automática de casos de test, el análisis de resultados y la generación de múltiples informes.

Las relaciones existentes entre casos de test, requisitos y ejecuciones quedan registradas en todo momento, lo cual garantiza la trazabilidad durante todo el proceso.

La interfaz intuitiva de **SQS TestWORKFLOW** permite a cualquier tipo de usuario comprender el proceso de validación y familiarizarse con ella sin necesidad de ser un especialista. La herramienta se basa en una metodología que permite además estructurar los procesos de test.

Definición y gestión de requisitos

Una de las claves para que un proceso de validación tenga éxito radica en una buena gestión de los requisitos del sistema.

Los requisitos definen el comportamiento del sistema. Cada uno de ellos modela una función del mismo de forma que el conjunto completo de requisitos debe cubrir su funcionalidad completa.

Al realizar la validación de un sistema se debe garantizar que todos los requisitos sean comprobados; esto se consigue relacionando requisitos y Test Cases.

Con **SQS TestWORKFLOW** los requisitos se definen utilizando una plantilla en forma de tabla que contiene toda su información relevante permitiendo una sencilla gestión y administración. Los requisitos se pueden agrupar para mostrar similitudes o dependencias entre ellos y organizar mediante “drag and drop”.

La implementación de nuevos requisitos o la modificación de los existentes se realiza de una manera sencilla.

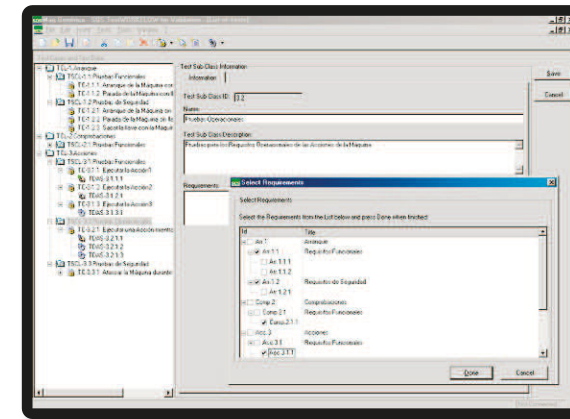
ID	Título	Descripción	Tipo	Dep.	Comentarios
Req 1	Inicio		Handing	Test Central	
Req 1.1	Requisito Funcional	El usuario debe poder iniciar la sesión de usuario.	Handing	Test Central	
Req 1.2	Requisito de Seguridad	Al iniciar el sistema se debe mostrar un mensaje de bienvenida.	Handing	Test Central	
Req 1.3	Requisito de Seguridad	El sistema debe mostrar un mensaje de bienvenida.	Handing	Test Central	
Comp 2.1	Compendio	Se debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2	Inicio		Handing	Test Central	
Req 2.1	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.2	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.3	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.4	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.5	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.6	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.7	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.8	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.9	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.10	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.11	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.12	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.13	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.14	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.15	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.16	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.17	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.18	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.19	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.20	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.21	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.22	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.23	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.24	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.25	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.26	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.27	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.28	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.29	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.30	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.31	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.32	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.33	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.34	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.35	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.36	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.37	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.38	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.39	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.40	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.41	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.42	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.43	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.44	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.45	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.46	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.47	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.48	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.49	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	
Req 2.50	Requisito Funcional	El sistema debe poder acceder a los datos de los usuarios.	Handing	Test Central	

Organización de Tests

Los tests se organizan en una estructura jerárquica arborescente que consta de los siguientes elementos: Test Class, Test Sub Class, Test Case y Test Data.

Test Class: La Test Class es el elemento situado en el nivel más alto de la estructura. Contiene una o varias Test Sub Classes. Una Test Class representa una agrupación lógica de requisitos del sistema.

Test Sub Class: La Test Sub Class es un subconjunto dentro de una Test Class. En un sistema complejo la cantidad de requisitos es demasiado elevada para que una división en Test Classes resulte suficientemente precisa. Por ello, las Test Sub Classes se utilizan para hacer un refinamiento de los requisitos en subconjuntos más pequeños.



Test Case: Un Test Case sirve para comprobar una funcionalidad específica del sistema. Todo requisito debe estar contenido en al menos un Test Case, si bien un Test Case puede cubrir más de un requisito.

Test Data: Un Test Data consiste en unos datos concretos para un Test Case, imprescindibles para la ejecución de éste. Todo Test Case, entendido como la definición lógica de un requisito, contiene un conjunto de Test Datas. El conjunto de Test Datas debe cubrir

cualquier posible combinación del Test Case al que pertenece. Los valores específicos de un Test Data dependen directamente de la implementación del sistema.

Los elementos ejecutables de un Test Data se denominan **Steps**. Cada Step lleva a cabo una acción en el sistema como, por ejemplo, pulsar un botón, generar una entrada o solicitar una información del sistema. En **SQS TestWORKFLOW** existen Steps de inicialización y de ejecución.

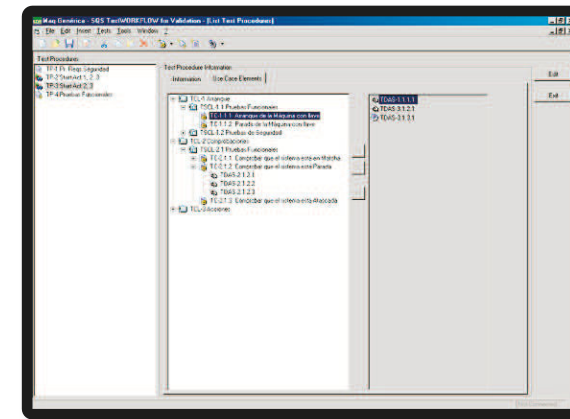
Partiendo de un estado inicial del sistema que, en caso necesario, puede alcanzarse ejecutando los Steps de inicialización correspondientes, los Test Data lanzan los Steps de ejecución que llevarán al sistema a su estado final.

Los Steps pueden agruparse en Macros. Una **Macro** es una secuencia de Steps que suele ejecutarse con frecuencia. Un ejemplo típico de Macro sería el de una parte de una secuencia de inicialización que debe ser ejecutada de manera idéntica para la mayor parte de los Test Data.

Los Test Data pueden agruparse en series y almacenarse como un **Test Procedure**.

Existen dos tipos de Test Procedures: Collections y Use Cases.

Una **Collection** consiste en una serie de Test Datas independientes entre sí.



Puede tratarse de **Test Collections**, en cuyo caso los Test Data son seleccionados desde un punto de vista orientado al testing, o **Requirements Collections**, en donde los Test Data son seleccionados en base a los requisitos.

Un **Use Case** consiste en una serie de Test Datas que simulan el comportamiento de un usuario. Estos Test Datas ejecutan las acciones que un usuario llevaría a cabo en el sistema siguiendo un orden concreto.

Ejecución

Una vez definidos los Test Data se puede lanzar la ejecución de los tests.

Los Tests pueden ejecutarse automáticamente aunque el usuario tiene la opción de controlar la ejecución de forma manual (ejecución semiautomática).

En ambos casos **SQS TestWORKFLOW** crea un fichero de 'logs' para almacenar los resultados de los tests.

Para una ejecución puede seleccionarse un Test Data individual, aunque lo más habitual será que una ejecución esté formada por un conjunto de varios Test Datas seleccionados para testear un conjunto de Test Cases o Requirements.

En **SQS TestWORKFLOW** existen los siguientes tipos de ejecuciones:

- Ejecución por Tests
- Ejecución por Test Procedures
- Ejecución por Requisitos

Análisis de resultados

SQS TestWORKFLOW también evalúa los resultados de la ejecución de los tests.

En una ejecución automática la herramienta evalúa los resultados obtenidos para determinar si éstos coinciden con los esperados.

En una ejecución semi-automática la herramienta muestra los resultados generados por cada Test Data para que el usuario pueda decidir si cumplen con los esperados.

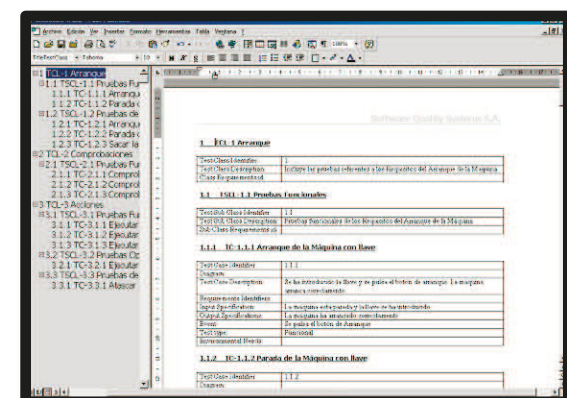
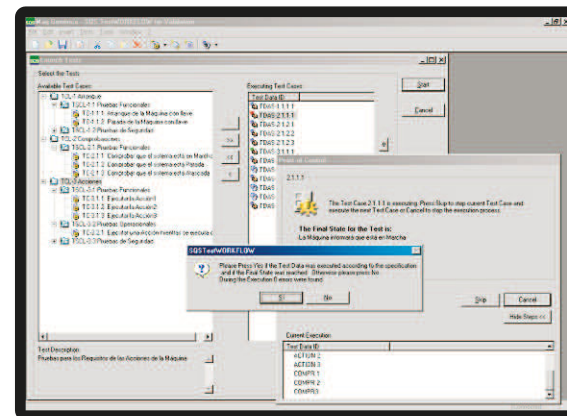
El conjunto completo de tests se evalúa después globalmente de acuerdo con los criterios de aceptación del sistema.

Informes

El usuario dispone de diversos informes con los que obtener en cualquier momento información acerca del estado actual de la organización y la ejecución de los tests.

Estos informes muestran información detallada acerca del sistema y facilitan la trazabilidad.

Con el objetivo de proporcionar un documento sencillo y estandarizado con el que cualquier usuario pueda trabajar estos informes son exportados a Microsoft Word.



Algunos informes típicos son:

Test Design Specification, detalla la estructura de los tests en Test Classes, Test Sub Classes y Test Cases.

Requirements Specification Report, muestra todos los requerimientos del sistema y sus relaciones.

Executed Tests Report, lista todos los elementos de un test y la ejecución de los Test Datas asociados. También proporciona estadísticas acerca del sistema incluyendo un recuento de los Test Data y sus estados asociados: 'passed', 'failed' o 'not executed'.

Existe también la posibilidad de que el usuario defina y personalice sus propios informes.

Glosario

Validación: Es el proceso de testear un sistema para asegurar que cumple con los requisitos que definen su comportamiento.

Test Class: Ocupa el nivel más alto en la estructura de los tests y representa a un grupo lógico de requisitos del sistema. Puede contener una o varias Test Sub Classes.

Test Sub Class: Es un subconjunto dentro de un Test Class. Se utiliza para refinar un conjunto de requisitos en otros más pequeños ya que en los sistemas complejos el número de requisitos es muy elevado y la división en Test Classes no es lo suficientemente precisa.

Test Case: Un Test Case sirve para comprobar una funcionalidad específica del sistema. Todo requisito del sistema debe estar contenido en al menos un Test Case, aunque un Test Case puede cubrir más de un requisito.

Test Data: Cada Test Case, que coincide con la definición lógica de un requisito, contiene un conjunto de Test Datas. Un Test Data es un test ejecutable con unos valores concretos que dependen directamente de la implementación del sistema. El conjunto de Test Datas debe cubrir todas las posibles combinaciones del Test Case al que pertenece.

Requisito: Los requisitos especifican el comportamiento exigido al sistema. Cada requisito modeliza una función del sistema de manera que la funcionalidad completa de éste debe quedar totalmente cubierta por el conjunto de requisitos definido. Para validar un sistema debe garantizarse que todos y cada uno de los requisitos han sido comprobados; esto puede garantizarse gracias a la relación entre requisitos y Test Cases.

Test Procedure: Serie de Test Datas que pueden ser almacenadas. Existen dos tipos de Test Procedures: Collections y Use Cases.

Use Case: Serie de Test Data que simulan el comportamiento de un usuario.

Informe: Documento que contiene descripciones acerca del estado actual de la organización y ejecución de los tests. Los informes proporcionan información extensa acerca del sistema y facilitan la trazabilidad.

Entorno: Todas las estructuras de tests, requisitos, informes y otras informaciones relacionadas con un proyecto se almacenan en un fichero denominado 'entorno'.

Para cada proyecto, entendido éste como un conjunto de programas de software y sus requisitos y/o especificaciones para una versión/release concreta, deberá definirse un entorno.