

Evaluación de un producto de software

Se debe realizar una planificación para la evaluación de productos de software según el modelo de evaluación definido en la ISO/IEC 14598 y las características/métricas de la calidad de producto definidos en la ISO/IEC 9126.

Bierkings surgió a partir del deseo de un grupo de amigos de producir su propia cerveza. Con el tiempo empezaron a vender barriles a algunos despachos locales y a raíz del éxito acababan de obtener contratos para exportar a 30 países.

Necesitan sumar rápidamente una fuerza laboral enorme y no pueden asumir el costo de licencias de software por lo que van a instalar todas las computadoras con sistemas operativos y aplicaciones libres. Los usuarios serán 40 personas con edades entre los 30 y los 50 años y en particular en el área contable se desempeñarán personas con conocimientos apenas básicos de informática. La aplicación contable es web pero requiere la instalación de un servidor Microsoft SQL Server y el uso de una carpeta compartida donde se deben copiar algunos archivos para operar con la AFIP. Las personas necesitarán acceder a la web y a la carpeta compartida fácilmente. En base a estos requerimientos, solicitan que se realice una evaluación de calidad de producto con diferentes opciones de manejadores de ventanas para xorg en Linux y así realizar la elección fundamentada en los resultados obtenidos.

1. Seleccionar 3 productos a evaluar, teniendo en cuenta los requerimientos de la empresa. Explicar por qué fueron seleccionados y las características de cada uno de ellos (diferencias y similitudes).

Para desarrollar la comparación evaluaremos los siguientes manejadores de ventanas para xorg en Linux: Compiz, Fluxbox y Awesome.

Seleccionamos estos tres debido al tipo al que responden, siendo los dos primeros de tipo Stacking y el último de tipo Dynamic. Esto quiere decir, que son los tipos más parecidos al manejo que hace Windows de sus ventanas. Como nuestros usuarios finales, son personas que no están por demás o necesariamente familiarizadas con la informática, nos pareció importante que el nuevo sistema a utilizar sea lo más parecido posible a lo que ya conocen.

Por otra parte, Compiz y Fluxbox tienen licencia MIT; por su parte, Awesome tiene licencia GPL. La primera es una licencia del tipo "licencia de software permisiva", lo que significa que tiene pocas restricciones de reutilización. La segunda, es una licencia completamente libre lo que significa que el software que la utiliza puede ser reutilizado sin restricciones. Ambas licencias son compatibles y la mayoría de los productos que las utilizan son de uso gratuito, lo que es una importante aspiración para esta selección.

Algunas características particulares de los diferentes manejadores:

Compiz es un administrador de ventanas de composición, que utiliza hardware de gráficos 3D para crear efectos de escritorio de composición rápida para la administración de ventanas.

Awesome es altamente configurable, muy rápido, extensible y, como dijimos previamente, está bajo la licencia GNU GPLv2.

Fluxbox, por su parte, es muy ligero, consume pocos recursos y rápido, provee de varias herramientas útiles como tabulación y agrupación, además es fácil de configurar.

2. En base a lo definido anteriormente, para la característica Mantenibilidad de la ISO/IEC 9126-1 seleccionar al menos 5 métricas de la ISO/IEC 9126-2 que abarquen todas las subcaracterísticas de la misma.

Métricas según la ISO/IEC 9126-2:

❖ **Mantenibilidad:**

- Analizabilidad -> Eficiencia del análisis de fallas
- Cambiabilidad -> Modificabilidad de la parametrización
- Estabilidad -> Proporción de éxito de cambios
- Pruebabilidad -> Retesteo eficiente
- Cumplimiento de la mantenibilidad -> Cumplimiento de la capacidad de mantenimiento

3. Realizar la planificación de la evaluación completando los siguientes ítems:

7. Establecer los requisitos de la evaluación

7.1 Establecer el propósito de la evaluación

El propósito de la evaluación del producto es obtener la medida en que el producto cumple con respecto a su madurez, estabilidad, desempeño, y la frecuencia y solidez de sus actualizaciones, con el objetivo de poder orientar la adquisición con respecto al cumplimiento o no de las necesidades establecidas.

En base a las mediciones obtenidas se realizará la comparación de los 3 productos evaluados, con la intención de ayudar y aconsejar en la elección del más adecuado.

7.2 Identificar el tipo de producto a evaluar

Se evaluará la versión más estable de cada uno de los tres productos finales a comparar, siendo todos productos de terceros con licencias libres (Compiz y Fluxbox tienen licencia MIT, y awesome licencia GPL). Los productos seleccionados son desarrollados por empresas y/u organizaciones externas, por lo tanto analizaremos la mantenibilidad en función de herramientas y repositorios de acceso público que nos brinden la información necesaria y pruebas de usuario por un tiempo determinado.

7.3 Especificar el modelo de calidad

Para el análisis de calidad de los distintos productos a comparar se utilizará el modelo definido en la ISO/IEC 9126-1. Esta describe un modelo de dos partes para la calidad del producto de software: calidad interna y externa, y calidad en uso.

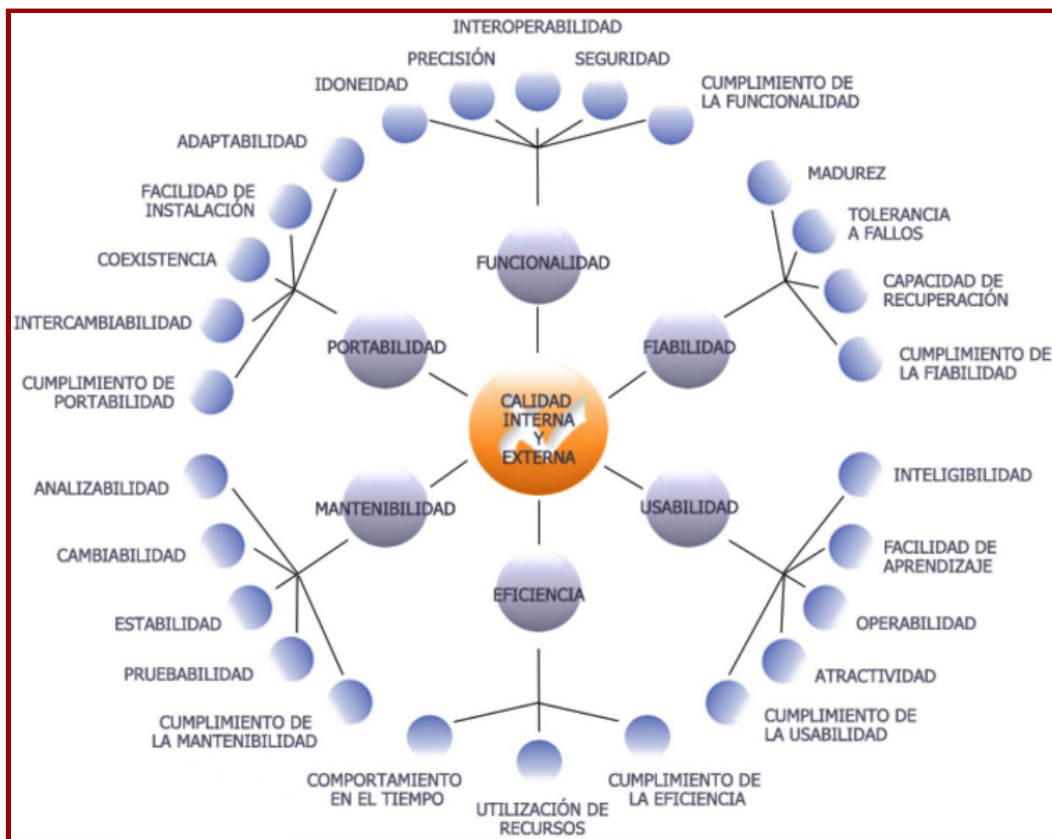
La calidad interna es medible a partir de las características intrínsecas, como el código fuente, y puede mejorar la calidad del producto de software por medio del rediseño del software.

En cambio, la calidad externa es medible en el comportamiento del producto, es decir, cuando el software es ejecutado. Es medida y evaluada mientras se prueba en un ambiente con datos simulados y usando métricas externas.

Por su parte, la calidad en uso es medible durante la utilización efectiva por parte del usuario. La calidad en uso se refiere a la perspectiva del usuario con relación a la calidad del producto al ser utilizado en un ambiente y contexto específico, es decir, si los usuarios pueden lograr sus metas usando el producto.

Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación más completo.

Atributos de la calidad interna y externa



Las capacidades que debe cumplir un producto software se describen en los siguientes atributos, estos deben ser evaluados cuando el producto del software es utilizado bajo condiciones especificadas:

- ❖ **Funcionalidad:** capacidad para proveer funciones que cumplan con necesidades específicas o implícitas.
- ❖ **Fiabilidad:** capacidad para mantener un nivel especificado de rendimiento.
- ❖ **Usabilidad:** capacidad para ser atractivo, entendido, aprendido y utilizado por el usuario.
- ❖ **Eficiencia:** capacidad para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados.
- ❖ **Mantenibilidad:** capacidad para ser modificado.
- ❖ **Portabilidad:** capacidad para ser transferido de un ambiente a otro.

Atributos de la calidad de uso:

Las capacidades que debe cumplir un producto software se describen en los siguientes atributos, estos deben ser evaluados en un contexto específico:

- ❖ **Efectividad:** capacidad para permitir a los usuarios alcanzar objetivos especificados con exactitud y completitud.
- ❖ **Productividad:** capacidad para permitir a los usuarios gastar una cantidad adecuada de recursos con relación a la efectividad alcanzada.
- ❖ **Seguridad física:** capacidad para alcanzar niveles aceptables del riesgo de hacer daño a personas, al negocio, al software, a las propiedades o al medio ambiente.
- ❖ **Satisfacción:** capacidad para satisfacer a los usuarios.

En base a la norma ISO 9126-1 se seleccionó el atributo de mantenibilidad, la cual es la capacidad del producto para ser modificado, y será analizada a través de las métricas externas definidas en la ISO/IEC 9126-2.

Características de la mantenibilidad:

- ❖ **Analizabilidad:** capacidad para serle diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas.
- ❖ **Cambiabilidad:** capacidad del producto software que permite que una determinada modificación sea implementada.
- ❖ **Estabilidad:** capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software.

- ❖ **Pruebabilidad:** capacidad del producto software que permite que el software modificado sea validado.
- ❖ **Cumplimiento de la mantenibilidad:** capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.

8. Especificar la evaluación

8.1 Selección de las métricas

Analizabilidad

| | |
|---|---|
| Nombre | Eficiencia del análisis de fallas (Failure analysis efficiency) |
| Propósito | ¿Puede el usuario analizar de manera eficiente la causa del fallo? (El usuario a veces realiza el mantenimiento configurando parámetros). ¿Puede el encargado de mantenimiento encontrar fácilmente la causa del fallo? ¿Qué tan fácil es analizar la causa del fallo? |
| Metodo de aplicacion | Observe el comportamiento del usuario o mantenedor que intenta resolver fallas. |
| Medición, fórmula | $X = \text{Sum}(T)/N$ $T = T_{out} - T_{in}$ T_{out} = Hora en la que se descubren las causas de la falla (o se informa al usuario) T_{in} = Hora a la que se recibe el informe de falla N = Número de fallos registrados |
| Interpretación | $0 \leq X$ Cuanto más corto, mejor. |
| Tipo de escala | Proporción |
| Tipo de medida | $T = \text{Time}$ $T_{in}, T_{out} = \text{Time}$ $N = \text{Count}$ $X = \text{Time}/\text{Count}$ |
| Fuente de medición | Informe de resolución de problemas Informe de operación |
| ISO/IEC 12207 SLCP Reference | Prueba de calificación Operación Mantenimiento |
| Audiencia | Desarrollador Mantenedor Operador |
| NOTA: 1. Se recomienda medir el tiempo máximo del peor de los casos y la duración del tiempo (ancho de banda) para representar la desviación. 2. Se recomienda excluir el número de fallas cuyas causas aún no se han encontrado cuando se realiza la medición. Sin embargo, la proporción de tales fallos desconocidos también debe medirse y presentarse en conjunto. 3. Desde el punto de vista del usuario individual, el tiempo es motivo de preocupación, mientras que el esfuerzo también | |

puede ser motivo de preocupación desde el punto de vista del encargado del mantenimiento. Por lo tanto, se pueden utilizar horas-persona en lugar de tiempo.

Cambiabilidad

| | |
|------------------------------|--|
| Nombre | Modificabilidad de la parametrización (Parameterised modifiability) |
| Propósito | ¿Puede el usuario o el encargado de mantenimiento cambiar fácilmente los parámetros para cambiar el software y resolver problemas? |
| Metodo de aplicacion | Observe el comportamiento del usuario o del mantenedor al intentar cambiar el software. De lo contrario, investigue el informe de resolución de problemas o el informe de mantenimiento. |
| Medición, fórmula | $X = 1 - (A / B)$ A = Número de casos en los que el mantenedor falla en cambiar el software mediante el uso de parámetros B = Número de casos en los que el mantenedor intenta cambiar el software utilizando el parámetro |
| Interpretación | $0 \leq X \leq 1$ Cuanto más cerca de 1.0, mejor. |
| Tipo de escala | Absoluto |
| Tipo de medida | A = Count B = Count X = Count/Count |
| Fuente de medición | Informe de resolución de problemas Reporte de mantenimiento Informe de operación |
| ISO/IEC 12207 SLCP Reference | Prueba de calificación Operación Mantenimiento |
| Audiencia | Desarrollador Mantenedor Operador Usuario |

Estabilidad

| | |
|----------------------|---|
| Nombre | Proporción de éxito de cambios (Change success ratio) |
| Propósito | ¿Puede el usuario operar el sistema de software sin fallas después del mantenimiento? ¿Puede el mantenedor mitigar fácilmente las fallas causadas por los efectos secundarios del mantenimiento? |
| Metodo de aplicacion | Observe el comportamiento del usuario o encargado del mantenimiento que esté operando el sistema de software después del mantenimiento. Cuenta las fallas que el usuario o el encargado del mantenimiento encontraron durante la operación del software antes y después del mantenimiento. De lo contrario, investigue el informe de resolución de problemas, el informe de funcionamiento o el informe de mantenimiento. |

| | |
|---|--|
| Medición, fórmula | $X = Na / Ta$ $Y = \{(Na / Ta) / (Nb / Tb)\}$ Na = Número de casos en los que el usuario encuentra fallas durante la operación después de que se cambió el software Nb = Número de casos en los que el usuario encuentra fallas durante la operación antes de que se cambie el software Ta = tiempo de funcionamiento durante el período de observación especificado después de que se cambia el software Tb = tiempo de funcionamiento durante el período de observación especificado antes de que se cambie el software |
| Interpretación | $0 \leq X, Y$ Cuanto más pequeño y más cercano a 0, mejor. |
| Tipo de escala | Proporción |
| Tipo de medida | Na, Nb = Count Ta, Tb = Time $X = \text{Count/Time}$ $Y = [(\text{Count/Time}) / (\text{Count/Time})]$ |
| Fuente de medición | Informe de resolución de problemas Reporte de mantenimiento Informe de operación |
| ISO/IEC 12207 SLCP Reference | Prueba de calificación Operación Mantenimiento |
| Audiencia | Desarrollador Mantenedor Operador |
| <p>NOTA:</p> <ol style="list-style-type: none"> 1. X e Y implican "frecuencia de encontrar fallas después del cambio" y "frecuencia fluctuante de encontrar fallas antes / después del cambio". 2. El usuario puede necesitar un período específico para determinar los efectos secundarios de los cambios de software, cuando se introduce la revisión del software para resolver problemas. 3. Se recomienda comparar esta frecuencia antes y después del cambio. Si se identifica una función modificada, se recomienda determinar si las fallas encontradas se detectan en la propia función modificada o en las demás. La extensión de los impactos puede clasificarse para cada falla. | |

Pruebabilidad

| | |
|----------------------|---|
| Nombre | Retesteo eficiente (Re-test efficiency) |
| Propósito | ¿Pueden el usuario y el encargado de mantenimiento realizar fácilmente pruebas operativas y determinar si el software está listo para funcionar o no? |
| Metodo de aplicacion | Observe el comportamiento del usuario o mantenedor que está probando el sistema de software después del mantenimiento. |
| Medición, fórmula | $X = \text{Sum}(T) / N$ T = Tiempo dedicado a probar para asegurarse de que se resolvió o no la falla informada N = Número de fallas resueltas |
| Interpretación | $0 < X$ |

| | |
|---|--|
| | Cuanto más pequeño, mejor. |
| Tipo de escala | Proporción |
| Tipo de medida | T = Time N = Count X = Time/Count |
| Fuente de medición | Informe de resolución de problemas Informe de operación |
| ISO/IEC 12207 SLCP Reference | Prueba de calificación Operación Mantenimiento |
| Audiencia | Desarrollador Mantenedor Operador |
| NOTA: X implica "tiempo (esfuerzo) promedio para probar después de la resolución de la falla". Si las fallas no se resuelven o reparan, excluirlas y medir por separado la proporción de dichas fallas. | |

Cumplimiento de la mantenibilidad

| | |
|--|--|
| Nombre | Cumplimiento de la capacidad de mantenimiento (Maintainability compliance) |
| Propósito | Cuán compatible es la capacidad de mantenimiento del producto con las regulaciones, estándares y convenciones aplicables. |
| Metodo de aplicacion | Cuente la cantidad de elementos que requieren cumplimiento que se han cumplido y compárelos con la cantidad de elementos que requieren cumplimiento en la especificación. |
| Medición, fórmula | $X = 1 - (A / B)$ A = Número de elementos que cumplen la mantenibilidad especificados que no se han implementado durante las pruebas B = Número total de elementos de cumplimiento de mantenibilidad especificados |
| Interpretación | $0 \leq X \leq 1$ Cuanto más cerca de 1.0 es mejor. |
| Tipo de escala | Absoluta |
| Tipo de medida | A = Count B = Count X = Count/Count |
| Fuente de medición | Descripción del producto (manual de usuario o especificación) de conformidad y normas, convenciones o regulaciones relacionadas Especificación de prueba e informe |
| ISO/IEC 12207 SLCP Reference | Prueba de calificación Validación |
| Audiencia | Proveedor Usuario |
| NOTA: Puede ser útil recopilar varios valores medidos a lo largo del tiempo, analizar la tendencia de aumentar los | |

elementos de cumplimiento satisfecho y determinar si están completamente satisfechos .

8.2 Establecer los niveles de puntuación para las métricas

| Eficiencia del análisis de fallas (Failure analysis efficiency) | |
|---|---|
| Excede los requerimientos | $X < 30$ minutos |
| Rango objetivo | $30 \text{ minutos} \leq X < 2 \text{ horas}$ |
| Mínimamente aceptable | $2 \text{ horas} \leq X < 4 \text{ horas}$ |
| Inaceptable | $X \geq 4 \text{ horas}$ |

| Modificabilidad de la parametrización (Parameterised modifiability) | |
|---|---------------------|
| Excede los requerimientos | $1 \geq X \geq 0,9$ |
| Rango objetivo | $0,9 > X \geq 0,7$ |
| Mínimamente aceptable | $0,7 > X \geq 0,4$ |
| Inaceptable | $0,4 > X \geq 0$ |

| Proporción de éxito de cambios (Change success ratio) | |
|---|---------------------|
| Excede los requerimientos | $X, Y < 0.5$ |
| Rango objetivo | $0.5 \leq X, Y < 1$ |
| Mínimamente aceptable | $1 \leq X, Y < 1.5$ |
| Inaceptable | $X, Y \geq 1.5$ |
| Nota: para establecer el nivel se toma el menor valor entre X e Y | |

| Retesteo eficiente (Re-test efficiency) | |
|---|---|
| Excede los requerimientos | $X < 30$ minutos |
| Rango objetivo | $30 \text{ minutos} \leq X < 4 \text{ horas}$ |
| Mínimamente aceptable | $4 \text{ horas} \leq X < 12 \text{ horas}$ |
| Inaceptable | $X \geq 12 \text{ horas}$ |

| | |
|--|----------------------|
| Cumplimiento de la capacidad de mantenimiento (Maintainability compliance) | |
| Excede los requerimientos | $1 \geq X \geq 0,95$ |
| Rango objetivo | $0,95 > X \geq 0,85$ |
| Mínimamente aceptable | $0,85 > X \geq 0,75$ |
| Inaceptable | $0,75 > X \geq 0$ |

8.3 Establecer los criterios de evaluación

| MANTENIBILIDAD | ANALIZABILIDAD | CAMBIABILIDAD | ESTABILIDAD | PRUEBABILIDAD | CUMPLIMIENTO DE MANTENIBILIDAD |
|-----------------------|-----------------------|----------------|----------------|-----------------------|--------------------------------|
| Excede | Mínimamente aceptable | Excede | Excede | Excede | Rango objetivo |
| Rango objetivo | Mínimamente aceptable | Rango objetivo | Excede | Rango objetivo | Mínimamente aceptable |
| Mínimamente aceptable | Inaceptable | Rango objetivo | Rango objetivo | Mínimamente aceptable | Mínimamente aceptable |
| Inaceptable | Inaceptable | Inaceptable | Inaceptable | Inaceptable | Inaceptable |

9. Diseñar la evaluación

9.1 Elaborar el plan de evaluación (incluir casos de prueba)

Se realizarán dos planes de evaluación diferentes en función de las métricas seleccionadas.

Plan 1. Estudiar el historial

Investigar el informe de resolución de problemas, el informe de funcionamiento o el informe de mantenimiento

- ❖ Se analizarán los issues reportados en los repositorios oficiales de los distintos gestores de ventana en el último año, con el fin de detectar todas las fallas que hayan surgido debido a incompatibilidades al actualizar Linux o el gestor de ventanas correspondiente.
- ❖ Se revisará cualquier tipo de incompatibilidad surgida por actualizaciones de otros softwares o generadas por elementos del gestor que hayan quedado obsoletos.
- ❖ Se tomarán el 20% de los issues reportados como bugs y se obtendrá el tiempo promedio que tardan en responderse, por un lado y resolverse, por otro.
- ❖ Se tomarán el 25% de los issues reportados como features y se obtendrá el tiempo promedio que tardan en responderse, por un lado e implementarse, por otro.

Plan 2. Utilizar el producto

Observar el comportamiento del usuario o mantenedor que está probando el sistema de software después del mantenimiento o intentando resolver fallas.

- ❖ Se instalará el producto en un total de 10 equipos para que los usuarios realicen las pruebas por un período de 6 meses (podría ser extendido, en caso de considerarse necesario).
- ❖ Se recabará información de un total de 10 usuarios, a los cuales se les brindará una computadora con el software previamente instalado.
- ❖ Se observará y se llevará un registro de la capacidad de cambiar las opciones de configuración y la cantidad de tareas fallidas.