SERVICIO NACIONAL DE APRENDIZAJE SENA

APRENDIZ

ELKIN ALBERTO PEREZ GIL

CALIDAD DE SOFTWARE

INSTRUCTORA

MARIETH PERPIÑAN

2023

CUCUTA, NORTE DE SANTANDER

Elabore una investigación corta usando los materiales bibliográficos disponibles en la biblioteca o internet respecto a los fundamentos de calidad de software seleccionando, al menos tres fuentes que le permitan resolver los siguientes conceptos:

Definición de calidad de software.

La calidad se puede definir como el máximo estándar en un producto o servicio para el cliente, en este caso te presentaremos varias características que definen este concepto:

* Idoneidad funcional. El grado o nivel en que el software resultante proporciona funciones que satisfacen las necesidades establecidas para determinadas condiciones.
* Fiabilidad. El grado en que un sistema o componente software realiza sus funciones bajo determinadas condiciones y período de tiempo.
* Eficiencia en el desempeño. El rendimiento en relación con la cantidad de recursos utilizados en las condiciones establecidas.
* Usabilidad. El grado o nivel en que el software permite ser entendido, aprendido, usado y fácil para el usuario.
* Accesibilidad: el grado en el que un usuario bajo determinadas condiciones le da uso a un software sin complicaciones adversas.
* Seguridad. El grado de protección de la información y los datos para que las personas o los sistemas no autorizados no puedan leerlos o modificarlos y que permita el acceso y uso a las personas o sistemas autorizados.
* Compatibilidad. El grado en que dos o más sistemas o componentes pueden intercambiar información y/o realizar sus funciones.
* Mantenibilidad. El grado de eficacia y eficiencia con que el software puede ser. Modificado y/o actualizado.
* Transferibilidad. El grado o nivel en que un sistema o componente transferido de un hardware, software u otro entorno operativo a otro.

Después de haber leído las principales características del concepto de calidad de un software ahora nos enfocaremos a los objetivos principales de un software, que es una accesibilidad plena y una usabilidad disponible para todos los tipos de usuarios, en esta sesión también veremos cuales son las reglas junto con su significado.

**Reglas de usabilidad.**

1. Visibilidad del estado del sistema

El sistema (web, App o cualquier otro producto digital) debe siempre mantener informado al usuario de lo que está ocurriendo).

2. Relación entre el sistema y el mundo real

El sistema debe hablar el mismo lenguaje del usuario. Con esto me refiero a que los mensajes y frases que aparecen en el sistema deben ser familiares para el usuario. Usar términos fáciles y sencillos de entender ayudan a que este se sienta más familiarizado con el sistema.

3. Control y libertad del usuario

Es fundamental mostrar acciones que permitan al usuario regresar o salir de cualquier acción que tomó en el sistema. Las opciones de hacer y deshacer tienen que estar siempre visibles para que el usuario tenga el control de sus interacciones.

4. Consistencia y estándares

Los usuarios no deben cuestionarse si acciones, situaciones o palabras diferentes significan en realidad la misma cosa; sigue las convenciones establecidas.

5. Reconocer antes que recordar

El usuario no tiene por qué recordar todas las opciones que tiene nuestra aplicación o web, pero si podemos hacer que las reconozca haciéndolas siempre visibles en la aplicación.

6. Prevenir los errores

No se trata de ofrecer salidas ante los errores o páginas divertidas con diseños "bonitos" (para disimular el error) cuando ya ha tenido lugar. Se trata de que sea muy difícil que el usuario los cometa. Trabajar en un diseño simple y fácil de usar ayudará a que los usuarios tengan menos incidencia a cometer errores.

7. Flexibilidad y eficiencia de uso

Es importante ofrecer facilidades para los nuevos usuarios en la plataforma, pero también lo es ofrecer atajos de teclado o claves que hagan más rápida la navegación de usuarios más expertos. Permite que el sistema se adapte tanto a los usuarios principiantes como a los expertos.

8. Estética y diseño minimalista

El sistema debe apuntar a tener una interfaz sencilla y fácil de entender, quita todo lo que sobra de tu web, todo lo que compita con otra información, cuantos menos elementos, mejor. Esto ayudará al usuario a enfocarse en lo importante.

9. Ayudar en el diagnóstico del error

Los mensajes de error del sistema deben tener un lenguaje claro y conciso que ayuden al usuario a identificar cuál fue el error y darle opciones para solucionarlo rápidamente.

10. Ayuda y documentación

Aunque es mejor que el sitio web o aplicación pueda ser usado sin ayuda, puede ser necesario proveer cierto tipo de ayuda. En este caso, la información debe ser fácil de localizar, especificar los pasos necesarios y no ser muy extensa.

**Reglas de la accesibilidad.**

Perceptible

Los usuarios, incluidos los que dependen de tecnología de apoyo, deben poder fijarse en la información que muestra el sitio web y utilizarla.

Añadir alternativas textuales a las imágenes permite que los lectores de pantalla las describan a alguien con problemas de visión, ayudándote a cumplir este principio.

Operable

Todo el mundo debe poder navegar y utilizar el sitio web. Este principio abarca diversos aspectos de la navegación y el control, garantizando que todos los usuarios puedan utilizar el sitio.

Puedes fomentar este principio permitiendo a los usuarios navegar por el sitio web con un teclado si no tienen o no pueden acceder a un ratón.

Además, no muestres en el sitio web elementos parpadeantes o de reproducción automática que puedan provocar convulsiones, para garantizar que tu sitio es seguro para las personas con discapacidad.

Comprensible

La información y el funcionamiento del sitio web deben ser fáciles de entender. En otras palabras, el sitio web debe presentar un contenido legible en una estructura organizada y una navegación predecible.

Robusto

Este principio dicta que el contenido del sitio sea accesible a través de diferentes navegadores web y agentes de usuario, incluidas las soluciones de tecnología de asistencia.

Puedes conseguirlo manteniéndote al día de los últimos estándares de las tecnologías de la información y la comunicación, como HTML5 y CSS3.

¿Cuál es el concepto de usabilidad en software?

La usabilidad es un aspecto controlado del diseño de la experiencia de usuario que garantiza que el usuario final no se esfuerce o encuentre problemas con el uso de la interfaz de usuario de un producto o sitio web. Un diseñador de experiencia de usuario puede controlar la accesibilidad, la interfaz de usuario, la arquitectura de la información y la usabilidad para adaptarse a los aspectos no controlados, como los objetivos, el estilo de vida y los hábitos del usuario.

¿Cuáles son los estándares de calidad de software?

ISO 9001 – Gestión de la Calidad

Esta norma para la implementación de un método o Sistema de Gestión de la Calidad (SGC) se ha convertido desde el año 2002 en un referente global con más de un millón de empresas certificadas. Acredita ante cualquier parte interesada la capacidad de una organización de satisfacer los requisitos del cliente. Es el certificado ISO más común y mejor reconocido y sus requisitos son genéricos y aplicables a cualquier organización.

ISO 10005:2018 Sistemas de Gestión de la calidad. Directrices para los planes de la calidad

Esta norma, que sustituye a ISO 10005:2015, ofrece las directrices para gestionar un plan de calidad que incremente la confianza, el control y la oportunidad de mejora durante todo el ciclo de vida. Se enfoca en el desarrollo, revisión, aceptación, aplicación y revisión del plan para adaptarlo a procesos, productos, proyectos o contratos. Las fases de desarrollo de un plan de la calidad son:

* Identificación de la necesidad de un plan de calidad
* Identificación de las entradas
* Alcance
* Preparación
* Contenido
* Revisión, aceptación e implementación
* Normas enfocadas al Aseguramiento de la Calidad (Quality Assurance)

ISO 33000 Calidad de los procesos de desarrollo de software

Esta norma sustituye a ISO 15504 Evaluación y Mejora de la capacidad y madurez de procesos. También llamada Process Improvement and Capability Determination (SPICE), aporta unas líneas de trabajo para la evaluación de procesos software. Proporciona una base que permite evaluar el punto en el que se encuentra una empresa. De esta manera, es posible comprobar la evolución en el tiempo, hacer un seguimiento de la situación respecto a la competencia y determinar posibles estrategias de mejora.

ISO 12207 Modelos de Ciclos de Vida del Software

Se trata de un estándar para los procesos de ciclo de vida del software, entendidos como un conjunto de actividades y tareas relacionadas. Hace el recorrido desde que surge la necesidad o nueva idea hasta la retirada del software. La conformidad se mide según el nivel de adecuación de los procesos al estándar definido en la norma.

ISO IEC IEEE 12207 Procesos de ciclo de vida de software

Recomienda​ un marco común para los procesos de Ciclo de Vida del Software, desde la necesidad hasta la retirada. El propósito es proporcionar requisitos uniformes mínimos aceptables para la preparación y el contenido de los planes de aseguramiento de la calidad del software.

IEEE 730 – 2002 Standard for Software Quality Assurance Plans

Este estándar define lo que es el software de alta calidad y es una recomendación para elaborar un Plan de Aseguramiento de la calidad de software (SQAP). Es utilizado en las fases de desarrollo y mantenimiento del ciclo de vida del software.

Normas enfocadas al Control de la Calidad (Quality Control)

ISO IEC 25000

Esta norma sustituye a ISO 9126 e ISO/IEC 14598. También conocida como SQuaRE (System and Software Quality Requirements and Evaluation), se trata de una familia de normas para evaluar la calidad del producto software. Está compuesta por las siguientes divisiones:

* Gestión de calidad
* Modelo de calidad
* Mediciones de calidad
* Requisitos de calidad
* Evaluación de la calidad
* Estándares de extensión

Sirve para medir ocho características de los productos software, que pueden descomponerse en su característica para un análisis en detalle:

* Adecuación funcional
* Fiabilidad
* Usabilidad
* Eficiencia
* Compatibilidad
* Seguridad
* Mantenibilidad
* Portabilidad

ISO 5055 Software Quality Standards

Calcula medidas de calidad basadas en la cantidad de debilidades críticas en el software. Analiza cuatro características de calidad relacionadas con las mencionada ISO IEC 25000:

* Seguridad
* Confiabilidad
* Eficiencia de rendimiento
* Mantenibilidad

Normas enfocadas a las Pruebas o al Testing.

ISO IEC IEE 29119 Norma para la documentación de prueba de software. Standard for Software Test Documentation.

Sustituye a IEE 829 – 1998. Está enfocada a la relación de las pruebas con las metodologías de desarrollo y el ciclo de vida software. Describe el papel de las pruebas en la gestión de la calidad y cómo parte de la verificación y validación del software. Menciona las pruebas estáticas y dinámicas y pone de manifiesto uno de los principios del testing de ISTQB: la imposibilidad de realizar pruebas exhaustivas sobre un producto.

Destaca la importancia de realizar las pruebas por terceras partes independientes al equipo de desarrollo. Define cómo diseñar estrategias, gestionar, priorizar y enfocar las pruebas. Además, marca la necesaria distinción entre niveles, tipos y técnicas de diseño de prueba.

Se presentan varios marcos de prueba, métricas, y modelos de documentación, gestión de configuración y soporte de herramientas. Define cómo se deben diseñar y ejecutar las pruebas, gestionar los datos necesarios para llevarlas a cabo y documentar e informar de los resultados obtenidos. Hace la necesaria diferenciación entre pruebas manuales y automatizadas.

ISO IEC 20246 Ingeniería de Software

Esta norma sustituye a IEE 1028. Establece un marco genérico para revisiones de productos de trabajo. Cualquier artefacto producido por un proceso puede ser considerado un producto de trabajo. Esta norma es utilizada por cualquier organización para la gestión, desarrollo, pruebas y mantenimiento de sistemas y software. Contiene un proceso genérico con actividades, tareas, técnicas de revisión y plantillas de documentación que se aplican durante la revisión.

Define revisiones de productos de trabajo para cualquier fase del ciclo de vida. Está destinado a managers de proyecto, desarrollo, calidad y pruebas, así como a analistas comerciales, desarrolladores, probadores, clientes y todos aquellos involucrados en el desarrollo, las pruebas y el mantenimiento de sistemas y software.

En Hiberus contamos con certificaciones que acreditan nuestra competencia para ofrecer servicios dentro de las normas y estándares de calidad preestablecidos por organismos reguladores como la Organización Internacional de Normalización (ISO) y el Gobierno de España, responsable del Esquema Nacional de Seguridad (ENS), cuyo objetivo es reforzar las capacidades de defensa frente a las ciber amenazas sobre el sector público y certificar a los proveedores tecnológicos de la administración que garantizan unas condiciones mínimas de seguridad de la información.

Diferencias y estándares de aplicaciones web y móvil.

Características de una aplicación web.

1. No ocupa espacio en el dispositivo, no es necesario instalarla ni descargarla, ya que permanecen en línea.
2. Tiene una estructura de código más fácil de entender y mantener.
3. No necesita aprobación de una plataforma de aplicaciones, lo que la hace visible sin necesidad de autorizaciones.
4. Su velocidad es limitada, ya que consta de servidores para atender los usuarios, y estos son estrictamente limitados.
5. Para darle uso se necesita la conexión a una señal de internet.

Características de la aplicación independiente o móvil.

1. Tiene acceso a los recursos del sistema, por lo que puede funcionar rápidamente.
2. Se puede usar sin conexión a internet.
3. Tiene muchas mas funcionalidades y opciones que una app web.
4. Es mucho mas segura ya que necesita pasar diversos estándares de calidad antes de estar en una tienda de aplicaciones.
5. Se necesita mucha más dedicación e inversión para crear una App independiente.

Ahora que hemos entendido cuales son sus características y por ende identificar sus diferencias fácilmente ahora veremos cuales con los estándares básicos a las cuales estos softwares se deben adaptar.

Estándares de software de una aplicación web.

**W3C Web Content Accessibility Guidelines (WCAG):** Es el estándar internacional para la accesibilidad de sitios web y aplicaciones web. Define cuatro principios básicos: perceptibilidad, operabilidad, comprensibilidad y robustez.

**Responsive Web Design (RWD):** Conjunto de prácticas para diseñar páginas web que se adapten a diferentes tamaños de pantalla, desde computadoras de escritorio hasta dispositivos móviles.

**Web Application Security Standards:** Normas y buenas prácticas para proteger las aplicaciones web de ataques cibernéticos, como OWASP Top 10 y PCI DSS.

Estándares de software de una aplicación independiente o móvil.

Las aplicaciones móviles tienen una excepción, ya que funcionan en distintos dispositivos con sistemas operativos que dominan el mercado, esos son iOS y Android, por eso hay similitudes, pero notables diferencias en estos dos diferentes sistemas operativos de móviles en sus estándares de aplicaciones independientes.

Estándar de Android.

Android Material Design: Conjunto de principios y componentes de diseño para crear interfaces de usuario consistentes y atractivas en aplicaciones Android.

Google Play Developer Policy: Normas que deben cumplir las aplicaciones para ser publicadas en Google Play Store. Estas normas incluyen requisitos de seguridad, contenido y privacidad.

Estándar de iOS.

**Apple Human Interface Guidelines (HIG):** Pautas de diseño de interfaz de usuario para aplicaciones iOS, que establecen principios como la coherencia, la simplicidad y la usabilidad.

**Apple App Store Review Guidelines:** Pautas que deben cumplir las aplicaciones para ser publicadas en App Store. Estas normas incluyen requisitos de seguridad, contenido, privacidad y diseño.

¿Qué tecnologías existen para implementar calidad?

Existen diversas tecnologías y herramientas que se utilizan para implementar y asegurar la calidad del software durante todo su ciclo de vida:

Herramientas de Gestión de Requisitos:

Estas herramientas permiten gestionar y rastrear los requisitos del software, asegurando que se cumplan y se integren correctamente en el desarrollo del software. Algunas herramientas populares incluyen IBM DOORS, JIRA y RequisitePro.

Herramientas de Pruebas Automatizadas:

Estas herramientas permiten automatizar las pruebas del software, mejorando la eficiencia y la cobertura de las pruebas. Algunas herramientas populares incluyen Selenium, JUnit, TestComplete y HP UFT.

Herramientas de Análisis Estático de Código:

Estas herramientas analizan el código fuente del software en busca de errores, vulnerabilidades y problemas de calidad. Algunas herramientas populares incluyen SonarQube, Checkmarx y Coverity.

Herramientas de Análisis Dinámico de Código:

Estas herramientas analizan el comportamiento del software en tiempo de ejecución para identificar errores y problemas de rendimiento. Algunas herramientas populares incluyen Dynatrace, AppDynamics y New Relic.

Herramientas de Gestión de Configuración:

Estas herramientas permiten gestionar y controlar las versiones del software y los cambios realizados durante el desarrollo del software. Algunas herramientas populares incluyen Git, SVN y Mercurial.

Herramientas de Integración Continua y Entrega Continua:

Estas herramientas automatizan el proceso de integración, prueba y entrega del software, mejorando la eficiencia y la calidad del software. Algunas herramientas populares incluyen Jenkins, Travis CI, CircleCI y GitLab CI/CD.

Herramientas de Monitoreo y Registro:

Estas herramientas permiten monitorear el comportamiento y el rendimiento del software en tiempo real, identificando problemas y mejorando la calidad del software. Algunas herramientas populares incluyen ELK Stack (Elasticsearch, Logstash, Kibana), Splunk y Graylog.

Herramientas de Gestión de Defectos y Problemas:

Estas herramientas permiten gestionar y rastrear los defectos y problemas identificados durante el desarrollo y la prueba del software, asegurando que se resuelvan de manera efectiva. Algunas herramientas populares incluyen JIRA, Bugzilla y Mantis.

Elementos para tener en cuenta:

Se deben conocer las reglas de usabilidad y accesibilidad.

Describir cada una de las reglas.

Diferenciar entre una aplicación independiente y una aplicación web.

Describir las características de cada tipo de aplicación.

# Bibliografía.

Conceptos/características de calidad de software. © EIP | International Business School 2010-2024. <https://eiposgrados.com/blog-ciberseguridad/caracteristicas-de-calidad-del-software/> .

Usabilidad de software. Techlib Blog [¿Qué es la usabilidad del software? – Techlib Blog](https://techlib.net/blog/que-es-la-usabilidad-del-software/).

10 reglas de usabilidad (UX). LinkedIn Autor: Jacob Nielsen, escrito por: Ruuben viera. Jul 2021. [10 reglas de la usabilidad (UX) - LinkedIn](https://es.linkedin.com/pulse/10-reglas-de-la-usabilidad-ux-ruub%C3%A9n-viera-arroyo).

Características de las aplicaciones móviles y aplicaciones web. Anincubator Autor: Luis Herazo-desarrollador de tecnología. [Cuál es la diferencia entre la aplicación móvil y las ... - Anincubator](https://anincubator.com/cual-es-la-diferencia-entre-las-aplicacion-movil-y-las-aplicacion-web/).

Estándares de calidad de software. Hiberus. <https://www.hiberus.com/crecemos-contigo/los-estandares-de-calidad-del-software-mas-importantes/>.