

UNIDAD 1: ESTÁNDARES Y MODELOS DE CALIDAD APLICADOS AL SOFTWARE

1.1 ISO

La Organización Internacional de Normalización o ISO (del griego, ἴσος (isos), 'igual'), nacida tras la Segunda Guerra Mundial (23 de febrero de 1947), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación (tanto de productos como de servicios), comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones (públicas o privadas) a nivel internacional.

La ISO es una red de los institutos de normas nacionales de 163 países, sobre la base de un miembro por país, con una Secretaría Central en Ginebra (Suiza) que coordina el sistema. La Organización Internacional de Normalización (ISO), con sede en Ginebra, está compuesta por delegaciones gubernamentales y no gubernamentales subdivididos en una serie de subcomités encargados de desarrollar las guías que contribuirán al mejoramiento.

Las normas desarrolladas por ISO son voluntarias, comprendiendo que ISO es un organismo no gubernamental y no depende de ningún otro organismo internacional, por lo tanto, no tiene autoridad para imponer sus normas a ningún país. El contenido de los estándares está protegido por derechos de copyright y para acceder ellos el público corriente debe comprar cada documento.

Está compuesta por representantes de los organismos de normalización (ON) nacionales, que produce diferentes normas internacionales industriales y comerciales. Dichas normas se conocen como normas ISO y su finalidad es la coordinación de las normas nacionales, en consonancia con el Acta Final de la Organización Mundial del Comercio, con el propósito de facilitar el comercio, el intercambio de información y contribuir con normas comunes al desarrollo y a la transferencia de tecnologías.

La Organización ISO está compuesta por tres tipos:

Miembros simples, uno por país, recayendo la representación en el organismo nacional más representativo.

Miembros correspondientes, de los organismos de países en vías de desarrollo y que todavía no poseen un comité nacional de normalización. No toman parte

activa en el proceso de normalización pero están puntualmente informados acerca de los trabajos que les interesen.

Miembros suscritos, países con reducidas economías a los que se les exige el pago de tasas menores que a los correspondientes.

1.2 SPICE

El proyecto SPICE es el proyecto 07.29 del subcomité SC7. Es un documento compuesto de 9 partes que al momento de publicar el libro estaba en su primera revisión formal como Proposed Draft Technical Report (PDRT). El siguiente estado del proceso de estandarización es llegar al Draft Technical Report (DTR) y finalmente publicar Technical Report type 2 (TR-2). El tipo 2 significa que todavía existen dudas en la comunidad sobre la aceptación del documento. Después de dos años se realiza la última votación para que se vuelva el estándar internacional.

El Software Process Assessment (SPA) y el proyecto SPICE tienen sus orígenes en el creciente uso y dependencia de la Tecnología de Información que en consecuencia dió el incremento de frustración e incumplimiento de expectativas por parte de los desarrolladores y los usuarios de software.

Al principio de los 80's, los militares de E.U. y del Reino Unido se propusieron mejorar el mecanismo de selección de proveedores de software con el objetivo de detener el creciente costo de software, reducir riesgos en su desarrollo y mejorar la calidad de los productos de software.

En E.U., el departamento de defensa creó el Software Engineering Institute (SEI), con el objetivo de desarrollar el mecanismo de selección de proveedores. El modelo CMM y el trabajo e impacto de este instituto son bien conocidos.

Por su parte, en el Reino Unido, el comité conjunto del Gobierno, la Defense Industry Trade Association (DITA) y el Computing Policy Consultative Committee (CPCC) reconocieron la necesidad de abordar con mayor rigor el problema de selección de proveedores para los sistemas que dependen en gran medida de software (Software Intensive Systems). A la agencia Defense Evaluation Research Agency (DERA) se le encomendó la investigación de los métodos de evaluación de proveedores en la industria mundial.

1.3 CMMI

Capability Maturity Model Integration (CMMI) es un modelo de aseguramiento de la calidad que busca la mejora continua de las organizaciones mediante

el análisis y re-diseño de los procesos que subyacen en la organización. Fue creado por el SEI (Software Engineering Institute) de la Universidad de Carnegie-Mellon y patrocinado por el Ministerio de Defensa de los Estados Unidos. Con el propósito de lograr la mejora de los procesos, CMMI provee:

- Una forma de integrar los elementos funcionales de una organización [SEI07b].
- Un conjunto de mejores prácticas basadas en casos de éxito probado de organizaciones experimentadas en la mejora de procesos.
- Ayuda para identificar objetivos y prioridades para mejorar los procesos de la organización [SEI07b], dependiendo de las fortalezas y debilidades de la organización que son obtenidas mediante un método de evaluación.
- Un apoyo para que las empresas complejas en actividades productivas puedan coordinar sus actividades en la mejora de los procesos.
- Un punto de referencia para evaluar los procesos actuales de la organización

CMMI v1.2 corresponde a la tercera versión entregable del modelo CMMI, posterior a las versiones 1.02 (primera versión año 2000) y 1.1 (año 2002). Las versiones previas sirvieron como retroalimentación para que los propios usuarios, evaluadores y evaluados hicieran acotaciones sobre posibles mejoras, las cuales fueron estudiadas, refinadas y algunas incluidas en la versión 1.2. CMMI v1.2 para desarrollo, que corresponde a una de tres constelaciones de prácticas, es una guía que ayuda a manejar, medir y monitorear procesos utilizados en el desarrollo de productos y servicios de una organización, y contiene prácticas ligadas a la administración de proyectos, administración de procesos, ingeniería y soporte. Las otras dos constelaciones son CMMI para Adquisición que provee una guía para liderar la adquisición informada y decisiva, y CMMI para Servicios que proporciona una guía para la entrega de servicios a clientes internos y externos de la organización. Ambas constelaciones se encuentran aún en desarrollo.

Junto con CMMI se desarrolló y publicó el método de evaluación "Assessment Requirements for CMMI (ARC)" [SEI00] en el año 2000, el cual define los requerimientos considerados esenciales para realizar una evaluación de CMMI en una organización y "Standard CMMI Appraisal Method for Process Improvement", (SCAMPI) [SEI01], manual seguido por los evaluadores para medir el nivel de madurez de una organización. Estos dos documentos también se han actualizado como consecuencia de la retroalimentación de la comunidad involucrada en

CMMI, generando la última versión 1.2 de SCAMPI y ARC ambas publicadas el año 2006.

Representaciones

La representación usada en CMMI entrega una guía para efectuar las actividades de mejora de los procesos y es utilizada en el método de evaluación. Según el modelo se tienen dos formas para mejorar. Una forma es mejorar un proceso específico o un conjunto de ellos usando la Representación Continua (Continuous Representation) y la otra es la mejora de la organización completa según los procesos definidos y ocupados usando la Representación Escalonada o por Etapas (Staged Representation). En la Tabla 1 se muestran los niveles para estos dos tipos de representaciones.

Representación Continua

La representación continua se focaliza en la mejora de un proceso o un conjunto de ellos relacionado(s) estrechamente a un área de proceso en que una organización desea mejorar, por lo tanto una organización puede ser certificada para un área de proceso en cierto nivel de capacidad. Existen seis niveles de capacidad por donde transitan los procesos asociados a un área de proceso y cada nivel es construido sobre el nivel anterior, es decir para que un proceso alcance un nivel de capacidad necesariamente debe haber alcanzado el nivel anterior.

	<i>Representación Continua</i>	<i>Representación Escalonada</i>
	Nivel de Capacidad	Nivel de Madurez
Nivel 0	Incompleto	-
Nivel 1	Realizado	Inicial
Nivel 2	Manejado	Manejado
Nivel 3	Definido	Definido
Nivel 4	Manejado cuantitativamente	Manejado cuantitativamente
Nivel 5	Optimizando	Optimizando

Tabla 1: Niveles de Representación continua y escalonada.

Los niveles de capacidad son:

Nivel 0 - Incompleto: Un proceso es denominado "proceso incompleto" cuando una o más objetivos específicos del área de proceso no son satisfechos.

Nivel 1 – Realizado: Un proceso es denominado "proceso realizado" cuando satisface todos los objetivos específicos del área de proceso. Soporta y permite el trabajo necesario para producir artefactos [Chr06].

Nivel 2 – Manejado: Un proceso es denominado como "proceso manejado" cuando tiene la infraestructura base para apoyar el proceso. El proceso es planeado y ejecutado en concordancia con la política, emplea gente calificada los cuales tienen recursos adecuados para producir salidas controladas; involucra partes interesadas; es monitoreado, controlado y revisado; y es evaluado según la descripción del proceso.

Nivel 3 – Definido: Un proceso denominado "proceso definido" es adaptado desde el conjunto de procesos estándares de la organización de acuerdo a las guías de adaptación de la organización, y aporta artefactos, medidas, y otra información de mejora a los activos organizacionales.

Nivel 4 – Manejado cuantitativamente: Un proceso denominado "proceso manejado cuantitativamente" es controlado usando técnicas estadísticas y otras técnicas cuantitativas. Objetivos cuantitativos para la calidad y realización del proceso son establecidos y usados como criterios para manejar el proceso.

Nivel 5 – Optimización: Un proceso denominado "proceso optimización" es mejorado basado en el entendimiento de causas comunes de variación del proceso. Un proceso en optimización se focaliza en la mejora continua del proceso realizado a través de mejoras incrementales y usando innovación tecnológica.

Representación Escalonada

En la representación escalonada o por etapas se ofrece un método estructurado y sistemático de mejoramiento de procesos, que implica mejorar por etapas o niveles. Al alcanzar un nivel, la organización se asegura de contar con una infraestructura robusta en términos de procesos para optar a alcanzar el nivel siguiente. Por lo tanto es una organización la que puede ser certificada bajo un nivel, en este caso llamado nivel de madurez. Según esta representación un nivel de madurez está compuesto por áreas de procesos (ver Tabla 2) en donde los objetivos asociados a ese nivel deben ser cumplidos para que la organización pueda certificarse en aquel nivel de madurez. Hay cinco niveles de madurez, los que son descritos a continuación:

Nivel 1: Iniciado

En el nivel de madurez 1, la mayoría de los procesos son "ad-hoc" y caóticos. La organización usualmente no provee un ambiente estable para soportar los procesos. Éxitos en estas organizaciones se debe a la competencia y esfuerzos heroicos de la gente dentro de la organización y no al uso de procesos probados. A pesar de este caos, organizaciones pertenecientes al nivel de madurez 1 con frecuencia producen productos y servicios que funcionan; sin embargo, ellos frecuentemente exceden sus presupuestos y no cumplen sus planes. Estas organizaciones son caracterizadas por la tendencia a no cumplir sus compromisos, al abandono de procesos durante tiempos de crisis, y a la incapacidad para repetir sus éxitos. El Nivel 1 está caracterizado además por la realización de trabajo redundante, por personas que no comparten sus métodos de trabajo a lo largo de la organización y cuando una persona clave en un área de negocio específica dentro de la organización se marcha, su conocimiento se va con ella y se pierde para la organización. Es claro que el Nivel 1 es uno donde ninguna organización quiere estar y donde por lo general la mayoría que no tiene sus procesos definidos se encuentra.

Nivel 2: Manejado

En el nivel de madurez 2 se ordena el caos. En el nivel 2 las organizaciones se enfocan en tareas cotidianas referentes a la administración. Cada proyecto de la organización cuenta con una serie de procesos para llevarlo a cabo, los cuales son planeados y ejecutados de acuerdo con políticas establecidas; los proyectos utilizan gente capacitada quienes disponen de recursos para producir salidas controladas; se involucran a las partes interesadas; son monitoreados, controlados y revisados; y son evaluados según la descripción del proceso. La disciplina del proceso reflejada por el nivel de madurez 2 ayuda a asegurar que existen prácticas y los proyectos son realizados y manejados de acuerdo a los planes documentados. En el nivel de madurez 2 el estado de los artefactos y la entrega de los servicios siguen planes definidos. Acuerdos son establecidos entre partes interesadas y son revisados cuando sea necesario [Chr06]. Los artefactos y servicios son apropiadamente controlados. Estos además satisfacen sus descripciones especificadas, estándares, y procedimientos.

Nivel 3: Definido

En el nivel de madurez 3, procesos son caracterizados y entendidos de buena forma, y son descritos en estándares, procedimientos, herramientas, y métodos. El conjunto de procesos estándares de la organización, los cuales son la base para el nivel de madurez 3, es establecido y mejorado continuamente. Estos procesos

estándares son usados para establecer consistencia a través de la organización. Los proyectos establecen sus procesos adaptando el conjunto de procesos estándares de la organización de acuerdo a guías de adaptación.

Una diferencia importante entre el nivel 2 y 3 es el alcance de los estándares: la descripción de procesos y los procedimientos. En el nivel de madurez 2, los estándares pueden ser un poco diferentes en cada instancia específica del proceso (por ejemplo sobre un proyecto particular). En el nivel de madurez 3, los estándares, descripción de procesos y procedimientos para un proyecto, son adaptados desde un conjunto de procesos estándares de la organización a un particular proyecto o unidad organizacional y así son más consistentes. Otra distinción crítica es que el nivel de madurez 3, los procesos son típicamente descritos más rigurosamente que en el nivel 2. Un proceso definido claramente plantea el propósito, entradas, criterios de entrada, actividades, roles, medidas, pasos de verificación, salidas y criterios de salida. En el nivel de madurez 3, procesos son manejados más proactivamente entendiendo las interrelaciones de las actividades y medidas detalladas del proceso, sus artefactos y sus servicios.

Nivel 4: Manejado cuantitativamente

En el nivel de madurez 4, la organización y proyectos establecen objetivos cuantitativos para medir la calidad y realización de los procesos y los usa como criterios en el manejo de ellos. Los objetivos cuantitativos son definidos en base a las necesidades de clientes, usuarios finales, organización, y actores de los procesos. La calidad y realización de procesos son entendidos en términos estadísticos y son manejados durante todo el ciclo de vida del proceso. Para subprocesos seleccionados, se recolectan y analizan estadísticamente medidas sobre la realización de procesos. Estas métricas son incorporadas en el repositorio de métricas de la organización para apoyar la toma de decisiones. Causas especiales de variación de procesos son identificadas y, cuando sea necesario, las fuentes de estas causas son corregidas para prevenir futuras ocurrencias.

Una diferencia importante entre los niveles 3 y 4 es la capacidad de predicción de la realización del proceso. En el nivel de madurez 4, la realización de procesos es controlada usando técnicas estadísticas y cuantitativas, y el proceso es cuantitativamente predecible, en cambio en el nivel de madurez 3 la realización del proceso es sólo predecible cualitativamente.

Nivel 5: Optimizado

En el nivel de madurez 5, una organización mejora continuamente sus procesos basándose en el conocimiento de las causas comunes de variación inherente en los procesos. El nivel de madurez 5 se focaliza sobre la mejora continua de los procesos a través de mejoras continuas, incrementales y tecnológicas. Los objetivos de mejora cuantitativa de procesos para la organización son establecidos, continuamente revisados para reflejar cambios en los objetivos del negocio y usados como criterio en la mejora de procesos. Los efectos del empleo de las mejoras de procesos son medidos y evaluados contra los objetivos de mejora cuantitativa del proceso.

Una diferencia importante entre el nivel de madurez 4 y 5 es el enfoque de la variación de los procesos. En el nivel de madurez 4, la organización está orientada a encontrar causas especiales de variación y proveer una predicción estadística de los resultados. Sin embargo, los resultados pueden ser insuficientes para alcanzar los objetivos establecidos. En el nivel de madurez 5 la organización está enfocada en las causas comunes de variación de procesos y modificar los procesos afectados para mejorar la realización de ellos y alcanzar los objetivos cuantitativos de mejora de procesos.

Dado a que la organización con que se trabajará quiere certificarse en forma organizacional en Nivel de madurez 3, en adelante sólo se detallará el modelo según la Representación Escalonada.

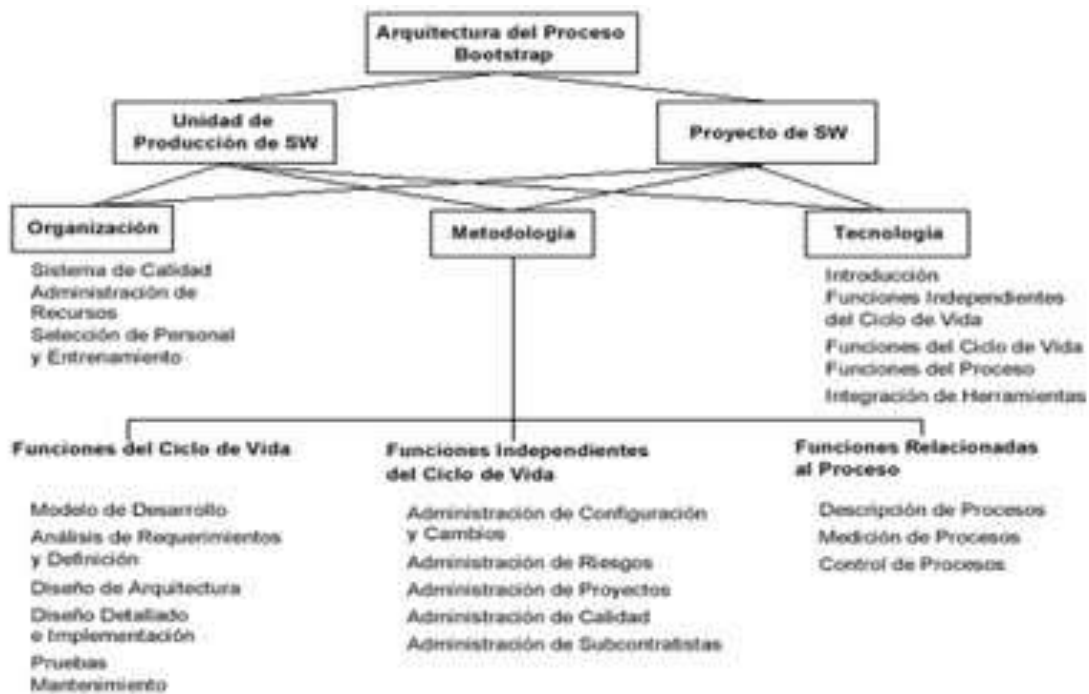
1.4 BOOTSTRAP

El Estándar Europeo para Evaluación y Mejoras de Procesos de Desarrollo de Software (Bootstrap) es otra de las iniciativas para resolver la crisis del desarrollo de software; mediante prácticas, herramientas y estándares de calidad internacional; mide, evalúa y propone mejoras al proceso de desarrollo de SW que siguen las Unidades de Producción de Software (UPS) de las empresas.

OBJETIVOS DE BOOTSTRAP

- Proporcionar apoyo a la evaluación de capacidad de proceso entre las mejores prácticas de una reconocida ingeniería de software.
- Incluir estándares de software reconocida internacionalmente como fuentes para identificar las mejores prácticas.
- Apoyar la evaluación de como el estándar de referencia ha sido puesta en práctica en la organización.

- Asegurar la fiabilidad de la evaluación.
- Identificar, en la organización, procesos fuertes y débiles.
- Apoyar planificación de mejora con resultados convenientes y confiables.
- Apoyar el logro de los objetivos de la organización planeando acciones de mejora.
- Ayudar a aumentar la eficiencia de los proceso poniendo en práctica la exigencias de estándares en la organización.



1.5 MOPROSOFT

Modelo de Procesos para la Industria del Software. Modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Desarrollado por la Asociación Mexicana para la Calidad en Ingeniería de Software ([1]) a través de la Facultad de Ciencias de la Universidad Nacional Autónoma de México (UNAM) y a solicitud de la Secretaría de Economía para obtener una norma mexicana que resulte apropiada a las características de tamaño de la gran mayoría de empresas mexicanas de desarrollo y mantenimiento

de software. Moprosoft es el nombre del modelo en la comunidad universitaria y profesional, y la norma técnica a la que da contenido es la NMX-059/01-NYCE-2005 que fue declarada Norma Mexicana el 15 de agosto de 2005 con la publicación de su declaratoria en el Diario oficial de la Federación.

Moprosoft considera que los modelos de evaluación y mejora CMMI e ISO/IEC 15504 no resultan apropiados para empresas pequeñas y medianas de desarrollo y mantenimiento de software. Sobre las áreas de procesos de los niveles 2 y 3 del modelo SW-CMM e inspirándose en el marco de ISO/IEC 15504 se ha desarrollado este modelo.

Criterios empleados

Se han aplicado los siguientes criterios para la elaboración de este modelo de procesos:

La estructura de procesos resultante debe ser acorde a la estructura generalmente empleada por las organizaciones de la industria del software (alta dirección, gestión y operación)

La alta dirección tiene un papel importante a través de la planificación estratégica. Debe actuar como promotor del buen funcionamiento de la organización a través de su implicación en la revisión y mejora continua del modelo.

El modelo considera a la gestión como proveedora de recursos, procesos y proyectos; así como responsable de la vigilancia del cumplimiento de los objetivos estratégicos de la organización.

El modelo considera a la operación como ejecutora de los proyectos de desarrollo y mantenimiento de software.

El modelo integra con claridad y consistencia los elementos indispensables para la definición de los procesos y las relaciones entre ellos.

El modelo integra los elementos para realizar la administración de proyectos desde un sólo proceso.

El modelo integra los elementos para realizar la ingeniería de productos de software en un único marco que incluya los procesos precisos de soporte (verificación, validación, documentación y control de la documentación).

El modelo destaca la importancia de la gestión de recursos, con especial relevancia en aquellos que componen el conocimiento de la organización: productos generados por proyectos, datos de los proyectos, mediciones, documentación de procesos y datos cosechados a partir del uso y de las lecciones aprendidas.

Moprosoft se basa en los modelos de procesos ISO 9001:2000, en las áreas de procesos de los niveles 2 y 3 de CMM-SW: CMM-SW v.1.1., en el marco general ISO/IEC15504 y en prácticas y conceptos de PMBOK Y SWEBOK.

PROSOFT representa un campo diferente de apoyo a los empresarios de las tecnologías de la información, es un sector diverso para hacer negocios y generar fuentes de empleo dignas”

El Plan Nacional de Desarrollo 2001-2006 plantea el fomento a la industria y el mercado De Tecnologías de la Información (TI) como estrategia para aumentar la competitividad del País. Dado el gran potencial con que cuenta México para desarrollar esta industria, la Secretaría de Economía, en coordinación con organismos empresariales y empresas del Sector, diseñó el PROSOFT.

UNIDAD 2: INGENIERÍA DEL SOFTWARE PARA COMERCIO ELECTRÓNICO

2.1 INTRODUCCIÓN

El comercio electrónico brinda a las empresas la posibilidad de expandir su mercado a niveles internacionales haciendo uso del internet como punto de partida. Hoy en día, las barreras geográficas, las fronteras entre países importan poco frente a la versatilidad que ofrece este medio a través de sus videoconferencias, chats, foros que comunican proveedores, empresas y potenciales clientes de todo el mundo. Con el objetivo de proveer a su empresa de la plataforma tecnológica ideal para responder a las necesidades de este nuevo mercado, Softelligence ofrece sus servicios en el desarrollo de software para comercio en línea contando con un grupo de expertos y las más avanzadas tecnologías digitales.

Actualmente, el desarrollo de software para comercio en línea genera una serie de beneficios. Una de las grandes ventajas de este software es que puede automatizar procesos que si se hacen de forma manual o tradicional tomaría una considerable cantidad de tiempo. Usted puede llevar su negocio a un mayor nivel

de eficiencia teniendo un software que permita a sus clientes cotizar en línea, comprar productos de manera fácil y rápida o encontrar la información que buscan dentro de su página de forma sencilla y clara.

Es por eso que sería útil el uso de diversas plataformas en materia de Ingeniería de Software como software a la medida, sistema bolsa de trabajo, registro de usuarios, sistema de reservaciones on line, software de gestión, motor bienes raíces, outsourcing de programadores, servidores, aplicación server provider, AJAX, SQL server, Visual Studio, soluciones integrales en marketing online, administrador de contenidos, cotizador, inventarios en línea, renta de tiendas en línea, entre otros.

Recuerde que tanto en el internet como en la mesa de negocios la rapidez en la atención al cliente sin perder calidad es una de las características que los usuarios y potenciales clientes valoran mucho al tomar su decisión de a quién comprarle. El desarrollo de software para el comercio electrónico puede brindar una solución integral que combine una atractiva identidad gráfica de su empresa, incluso impactante, a la par de un contenido inteligente y con procesos de ventas realmente agradables y fáciles de entender, como un carrito de compras a la medida, por ejemplo. Uno de los logros posibles con el desarrollo de software para comercio en línea, es darle al sitio un toque moderno y ergonómico, impactando visualmente a los usuarios.

Un software para comercio electrónico realiza diversas funciones traducidas en ventajas, entre ellas la de ayudar a administrar negocio, ya que puede realizar por sí mismo control de inventarios, avisando mediante reportes especializados de cuántas ventas hubo en el día, que producto está vendiéndose más, cuántas unidades quedan en stock ó saber cuántos días tardará el proveedor en surtirle nuevamente. Como parte de las ventajas del desarrollo de software para comercio en línea, puede realizar la atención a sus clientes ofreciendo a estos usuarios atractivas ventajas como la personalización de su producto, donde el sistema deja que el cliente decida sobre algunas características del artículo en cuestión como el color, tamaño y número de piezas que desea ordenar, siempre al margen de lo que tiene en existencia.

Otra de las ventajas en de las Tecnologías de la Información, es la gran capacidad de adaptabilidad, donde no sólo responde los requerimientos presentes sino que previene las necesidades futuras, siendo un ahorro bastante significativo en cuestión de inversión, ya que en vez de comprar un nuevo software cada cierto tiempo (como suele pasar con los software adaptables existentes en el mercado)

El software para comercio electrónico puede ser una útil herramienta y la mejor plataforma para hacer de su negocio, una empresa exitosamente competitiva.

2.2 SISTEMAS DISTRIBUIDOS

Un sistema distribuido se define como una colección de computadores autónomos conectados por una red, y con el software distribuido adecuado para que el sistema sea visto por los usuarios como una única entidad capaz de proporcionar facilidades de computación.

El desarrollo de los sistemas distribuidos vino de la mano de las redes locales de alta velocidad a principios de 1970. Mas recientemente, la disponibilidad de computadoras personales de altas prestaciones, estaciones de trabajo y ordenadores servidores ha resultado en un mayor desplazamiento hacia los sistemas distribuidos en detrimento de los ordenadores centralizados multiusuario. Esta tendencia se ha acelerado por el desarrollo de software para sistemas distribuidos, diseñado para soportar el desarrollo de aplicaciones distribuidas. Este software permite a los ordenadores coordinar sus actividades y compartir los recursos del sistema – hardware, software y datos.

Los sistemas distribuidos se implementan en diversas plataformas hardware, desde unas pocas estaciones de trabajo conectadas por una red de área local, hasta Internet, una colección de redes de área local y de área extensa interconectados, que en lazan millones de ordenadores.

Las aplicaciones de los sistemas distribuidos varían desde la provisión de capacidad de computo a grupos de usuarios, hasta sistemas bancarios, comunicaciones multimedia y abarcan prácticamente todas las aplicaciones comerciales y técnicas de los ordenadores. Los requisitos de dichas aplicaciones incluyen un alto nivel de fiabilidad, seguridad contra interferencias externas y privacidad de la información que el sistema mantiene. Se deben proveer accesos concurrentes a bases de datos por parte de muchos usuarios, garantizar tiempos de respuesta, proveer puntos de acceso al servicio que están distribuidos geográficamente, potencial para el crecimiento del sistema para acomodar la expansión del negocio y un marco para la integración de sistema usados por diferentes compañías y organizaciones de usuarios.

2.3 COMERCIO ELECTRÓNICO

El comercio electrónico, también conocido como e-commerce (electronic commerce en inglés), consiste en la compra y venta de productos o deservicios a

través de medios electrónicos, tales como Internet y otras redes informáticas. Originalmente el término se aplicaba a la realización de transacciones mediante medios electrónicos tales como el Intercambio electrónico de datos, sin embargo con el advenimiento de la Internet y laWorld Wide Web a mediados de los años 90 comenzó a referirse principalmente a la venta de bienes y servicios a través de Internet, usando como forma de pago medios electrónicos, tales como las tarjetas de crédito.

La cantidad de comercio llevada a cabo electrónicamente ha crecido de manera extraordinaria debido a Internet. Una gran variedad de comercio se realiza de esta manera, estimulando la creación y utilización de innovaciones como la transferencia de fondos electrónica, la administración de cadenas de suministro, el marketing en Internet, el procesamiento de transacciones en línea (OLTP), el intercambio electrónico de datos (EDI), los sistemas de administración del inventario y los sistemas automatizados de recolección de datos.

La mayor parte del comercio electrónico consiste en la compra y venta de productos o servicios entre personas y empresas, sin embargo un porcentaje considerable del comercio electrónico consiste en la adquisición de artículos virtuales (software y derivados en su mayoría), tales como el acceso a contenido "premium" de un sitio web.

- Tipos de comercio electrónico:
 - Business to Business (B2B):
Comercio realizado entre proveedores y clientes empresariales, o sea, de empresa a empresa;
 - Business-to-consumer o business-to-customer (B2C)
Es el comercio entre la empresa productora, vendedora o prestadora de servicios y el consumidor final;
 - Consumer to Consumer (C2C)
Comercio electrónico entre usuarios particulares de Internet. La comercialización de bienes o servicios no involucra a productores y sí a un consumidor final con otro consumidor final, sin intermediarios;
 - Government to consumers (G2C)
Comercio entre gobiernos ,provincial, federal o municipal y consumidores, por ejemplo, el pago de impuestos, multas y tarifas públicas;

- Government to Business (G2B)

Negocios entre gobierno y empresas, por ejemplo: las compras del Estado a través de internet por medio de licitaciones, concurso de precios, etc.

Un sistema de Administración de comercio electrónico o e-Commerce por su nombre en inglés es una aplicación desarrollada con el fin permitir el intercambio comercial (compra y venta) a través de medios electrónicos tales como internet, intranets corporativas y otras redes informáticas.

Los sistemas de comercio electrónico son usados como herramientas para:

Creación de canales nuevos de marketing y ventas.

Acceso interactivo a catálogos de productos, listas de precios y folletos publicitarios.

Venta directa e interactiva de productos a los clientes.

Soporte técnico ininterrumpido, permitiendo que los clientes encuentren por sí mismos, y fácilmente, respuestas a sus problemas mediante la obtención de los archivos y programas necesarios para resolverlos.

Mejorar los procesos de distribución.

Optimizar procesos operacionales y sistemas de comunicación entre las partes que intervienen.

Facilitar el procedo de fidelización de clientes.

Los sistemas de Comercio Electrónico implementan una serie de etapas que se corresponden con el proceso de compra tradicional:

Se identifican las necesidades.

Elección del producto a comprar.

Elección del vendedor, a quien se le va realizar la compra.

Negociación.

Compra y envío.

Uso y evaluación del producto adquirido.

Para realizar dicho proceso, existen sistemas de agentes, a continuación vamos a hablar sobre la función que realizan en el proceso y ejemplos de cada uno de ellos:

Agentes notificadoros

Su misión es avisar a sus usuarios cuando encuentran un producto acorde al que dicho usuario necesita.

Jango (<http://www.jango.com>): Utiliza Internet, para mediante el ingreso del nombre de un producto, buscar automáticamente en las tiendas y sitios más relevantes. Muestra al usuario un informe con todos los datos e incluso ayuda a cumplimentar el formulario de pedido, en caso de que se realice la compra.

Agentes de recomendación

Se basan en el perfil y el conocimiento del contexto de negocio, para poder recomendar a los usuarios que productos son más de su interés.

FireFly (<http://www.firefly.com>): Permite mediante el uso de encuestas e índices de popularidad determinar las preferencias de los usuarios, por ejemplo para el sector de música y obtiene una lista con los cd's que podrían ser de su interés. Utiliza algoritmos para automatizar el proceso de conocer opiniones de personas con similares gustos musicales. Yahoo utiliza esta tecnología para su generador de listas top música y películas. Uno de los inconvenientes es que está limitado a un comerciante, por lo que no realiza comparaciones.

Agentes de compra comparativa

Buscan al comerciante que ofrece las mejores condiciones para la compra del producto que desea el usuario.

Bargain Finder (<http://bf.cstar.ac.com>): Agente que busca en varias tiendas los precios más baratos para CD's y cintas. Para realizar la búsqueda en un determinado comerciante, este debe estar suscrito y pagar por aparecer como referencia a los clientes. La búsqueda se realiza por el nombre exacto del artista o álbum, lo cual hace imprescindible que el usuario sepa exactamente lo que desea comprar.

Agentes de negociación

Se encargan, mediante la existencia de agentes compradores y vendedores de llegar a acuerdos para la compra de un producto, como los procesos de negociación que existen para realizar una transacción comercial.

Kasbah: Sistema web donde el usuario puede crear agentes tanto compradores como vendedores. Ayuda al usuario en la búsqueda del producto deseado, y en la negociación del mejor precio posible

Agentes de subasta

Permite a los compradores el pujar en la red. Un ejemplo de estos agentes es AuctionBot.

2.4 TECNOLOGÍAS UTILIZADAS PARA EL COMERCIO ELECTRÓNICO

1. Conexiones (Sockets)

Un socket es un tipo de conducto que se utiliza para conectarse a una computadora, a su vez, conectada a una red basada en TCP/IP. El socket se configura de tal manera que los datos pueden ser bajados desde el cliente y devueltos al mismo. Los lenguajes de programación modernos, como Java, proporcionan servicios de alta calidad, por medio de los cuales un socket se puede conectar mediante “programación” a una computadora cuya dirección de Internet sea conocida, y donde los datos se pueden enviar por este conducto. La programación necesaria para esto, normalmente no es más complicada que la programación que se requiere para escribir y leer datos de un archivo. Los sockets son una implementación a bajo nivel de la conectividad.

Dentro de las aplicaciones típicas de un socket, están las aplicaciones de conferencia, donde una entrada a

Una conferencia se enviaría al servidor de conferencia que utiliza una configuración de sockets en el servidor.

Los sockets son mecanismos de bajo nivel, pero una forma muy eficiente de comunicar datos en un sistema distribuido que ejecuta el protocolo TCP/IP.

2. Objetos Distribuidos

Un objeto distribuido es aquel que reside en una computadora, normalmente un servidor, en un sistema distribuido. Otras computadoras del sistema pueden enviar mensajes a este objeto, como si residiera en su propia computadora. El software del sistema se hará cargo de varias funciones: localizar el objeto, recoger los

datos que se requieren para el mensaje y enviarlos a través del medio de comunicación que se utiliza para el sistema. Los objetos distribuidos representan un nivel más alto de abstracción que las conexiones (sockets), a partir de algún código de inicialización, el programador no es consciente del hecho de que el objeto reside en otra computadora.

Las tecnologías de objetos distribuidos usualmente empleadas son:

- RMI. Esta es una tecnología asociada al lenguaje de programación Java. Es un enfoque Java puro en el que sólo los programas escritos en ese lenguaje se pueden comunicar con un objeto RMI distribuido. Es la tecnología ideal para sistemas cerrados de Java; estos sistemas generalmente tendrán pocas conexiones o ninguna con otros sistemas.
- DCOM. Esta es una tecnología desarrollada por la compañía Microsoft y permite que, programas escritos en lenguajes tales como Visual Basic y Visual J++ (variedad de Java desarrollada por Microsoft), se comuniquen con los objetos que están en computadoras remotas.
- CORBA. Es la tecnología de objetos distribuidos más sofisticada. La característica más importante del enfoque CORBA es que es multilenguaje, donde los programadores pueden utilizar diferentes lenguajes de programación para enviar mensajes a objetos CORBA.

La principal ventaja de los objetos distribuidos sobre los sockets es, el hecho de que como abarca enteramente el paradigma de orientación a objetos, se pueden emplear los mismos métodos de análisis y diseño que se utilizan para la tecnología de objetos convencional.

3. Espacios

Esta es una tecnología que se encuentra en un nivel de abstracción, incluso más alto que los objetos distribuidos.

La tecnología de espacios concibe un sistema distribuido sobre la base a un gran almacén de datos persistentes, donde las computadoras de un sistema distribuido pueden leer o escribir. No concibe el sistema distribuido como una serie de programas que pasan mensajes a los demás, utilizando un mecanismo como los sockets, o como una serie de objetos distribuidos que se comunican utilizando métodos. Por el contrario, la tecnología de espacios conlleva procesos como escribir, leer o copiar datos a partir de un almacén persistente. Un programador que utiliza esta tecnología, no se preocupa por detalles como dónde están

almacenados los datos, qué proceso va a recoger los datos y cuándo los va a recoger.

4. CGI

El término CGI (Common Gateway Interface) significa, interfaz común de pasarela. Es la interfaz con el servidor Web al cual se puede acceder, mediante los programas que se ejecutan en el servidor. Gran parte de la interactividad asociada a las páginas Web, se implementa programando el acceso a la CGI. Por ejemplo, cuando el usuario de un navegador accede a una página que contiene un formulario, éste lo rellena y lo envía al servidor Web, programa el acceso a la CGI, procesa el formulario, y lleva a cabo la funcionalidad asociada al formulario.

5. Contenido Ejecutable

Es el término que se aplica a la inclusión en una página Web de un programa que se ejecuta, cuando la página es recuperada por un navegador. Este programa puede llevar a cabo un número diverso de funciones, entre las que se incluyen, la animación y la presentación de un formulario al usuario para insertar datos. Existen varias tecnologías que proporcionan servicios para insertar contenido ejecutable en una página Web, entre ellas se incluyen: applets, ActiveX y Javascript.

6. Paquetes Cliente / Servidor

Este término describe las colecciones de software que normalmente llevan a cabo algún tipo de procesamiento de sistemas. A continuación, se muestran ejemplos típicos de paquetes de software:

- Paquetes de reproducción de datos (Replicación). Este tipo de software realiza una transacción en la base de datos y la aplica a un número de bases de datos reproducidas, evitando así acceder a estas bases de datos hasta que todas estén sincronizadas.
- Paquetes de seguridad. Estos son paquetes que monitorizan el tráfico dentro de un sistema distribuido y avisan al administrador de sistemas de la aparición de cualquier violación posible en la seguridad.
- Monitores de transacciones. Estos son paquetes de software que administran las transacciones que tienen lugar dentro de un sistema distribuido y aseguran que se devuelvan los datos correctos como resultado de una transacción y en el orden correcto. Muchas de las funciones de estos monitores tienen que ver con asegurar

que los resultados correctos se devuelvan, incluso, en el entorno en donde podrían surgir errores de hardware o de transmisión.

2.5 INGENIERIA DE SEGURIDAD

La Ingeniería de la seguridad es una rama de la ingeniería, que usa todo tipo de ciencias para desarrollar los procesos y diseños en cuanto a las características de seguridad, controles y sistemas de seguridad. La principal motivación de esta ingeniería ha de ser el dar soporte de tal manera que impidan comportamientos malintencionados.

El campo de esta ingeniería puede ser muy amplio, podría desarrollarse en muchas técnicas:

Equipos: Como el diseño de cerraduras, cámaras, sensores,...

Procesos: políticas de control, procedimientos de acceso,...

Informático: control de passwords, criptografía,...

UNIDAD 3: INGENIERÍA WEB

3.1 ATRIBUTOS DE LOS SISTEMAS Y APLICACIONES WEB

¿Que es la usabilidad?

Técnicas que ayudan a los seres humanos a realizar tareas en entornos gráficos de ordenador.

Tres conceptos muy importantes. Trabajamos para seres humanos, que quieren realizar una tarea de una forma sencilla y eficaz y en este caso particular, la deben realizar frente a un ordenador en un entorno gráfico, la Web. La usabilidad ayuda a que esta tarea se realice de una forma sencilla analizando el comportamiento humano, y los pasos necesarios para ejecutar la tarea de una forma eficaz. Quizás suene un poco a trabajo industrial, pero al hacer referencia al término "tarea", estamos haciendo referencia a buscar un programa que queremos ver en la tele, a ver información sobre mi equipo de fútbol favorito, a comprar un libro sobre artes marciales.

¿Realmente hace falta la usabilidad para hacer una Web?

Si. Sea tu Web un portal o una página personal, no olvides que la gente que entra en tu Web es por que básicamente busca algo. Hacerlo sencillo es decisión tuya, pero seguro que entre todos podemos hacer de Internet un sitio mejor donde la gente no se sienta perdida o extraviada o sencillamente defraudada.

¿Cuales son las 8 reglas de Oro sobre la usabilidad?

Quizás mañana sean 10 reglas.

1. En Internet el usuario es el que manda.

Esto quiere decir que sin usuarios tu página muere, así que más vale que los cuides y les des lo que te piden, de lo contrario te quedarás solo.

2. En Internet la calidad se basa en la rapidez y la fiabilidad.

En Internet cuenta que tu página sea más rápida que bonita, fiable que moderna, sencilla que compleja, directa.

3. Seguridad.

Si en el mundo real a veces nos cuesta fiarnos del banco de la esquina, imagínate como se siente la gente en Internet cuando llega a tu site. Procura que todo funcione como un reloj para que la gente pueda fiarse de tu site.

4. La confianza es algo que cuesta mucho ganar y se pierde con un mal enlace.

Esto quiere decir que tal y como esta la competencia en Internet, no puedes perder ni un solo visitante por tener un enlace mal hecho. Es mejor salir con algo sencillo e irlo complicando poco a poco, que salir con todo y ver “que es lo que pasa”. Versiones 1.0 son buenas mientras lo que este puesto este bien asentado y genere confianza. Poco a poco y con el feedback de los usuarios, podrás ir complicando la pagina. Pero asegura antes de arriesgar.

5. Si quieres hacer una pagina decente, simplifica, reduce, optimiza.

La gente no se va a aprender tu site por mucho que insistas, así que por lo menos hazlo sencillo, reutiliza todos los elementos que puedas, para que de este modo los usuarios se sientan cómodos y no se pierdan cada vez que necesiten encontrar algo en tu site.

6. Pon las conclusiones al principio.

El usuario se sentirá más cómodo si ve las metas al principio. De esta forma no tendrá que buscar lo que necesita y perderá menos tiempo en completar su tarea. Si completa su tarea en menos tiempo se sentirá cómodo y quizás se dedique a explorar tú site o quizás se lo recomiende a un amigo.

7. No hagas perder el tiempo a la gente con cosas que no necesitan.

Cuidado con cruzar promociones, si lo haces por lo menos hazlo con cuidado. Procurar que la selección de productos a cruzar sea consecuente y no lo quieras “vender todo” en todas las paginas. Según avance el usuario en su navegación procura dejarle mas espacio libre. Puede ocurrir que cuando este punto de comprar algo vea una oferta que le distraiga y pierdas esa venta.

8. Buenos contenidos.

Escribir bien para Internet es todo un arte. Pero siguiendo las reglas básicas de (1) poner las conclusiones al principio y (2) escribir como un 25% de lo que pondrías en un papel, se puede llegar muy lejos. Leer en pantalla cuesta mucho, por lo que, en el caso de textos para Internet, reduce y simplifica todo lo que puedas.

9. Evaluación del contenido.

A diferencia del software de aplicación que evoluciona continuamente, las aplicaciones Web involucran una de manera continua. Un seguimiento continuo y detallado de la información implícita en la aplicación Web. Todo esto hace que la aplicación Web sirva al usuario fina.

3.2 ESTRATOS DE LA INGENIERÍA WEB

El desarrollo de aplicaciones Web incorpora métodos de proceso especializados, métodos de ingeniería de software adaptados a características de desarrollo de las aplicaciones Web y un conjunto de importantes tecnologías que permitan un correcto desarrollo de las mismas.

Los procesos, métodos y tecnologías (herramientas) proporcionan un enfoque en estratos de la IWeb que es conceptualmente idéntico a los estratos de la ingeniería de software.

Proceso

Los modelos de procesos Web adoptan la filosofía de desarrollo ágil. El desarrollo ágil enfatiza un enfoque de desarrollo riguroso que incorpora rápidos ciclos de desarrollo.

Es importante reconocer que el problema todavía debe analizarse, debe desarrollarse un diseño, la implementación debe preceder en una forma incremental y se debe iniciar un enfoque organizado de prueba. Dichas actividades de marco de trabajo se deben definir dentro de un proceso que:

- Adopte el cambio
- Aliente la creatividad y la independencia del equipo de desarrollo y fortalezca la interacción con el usuario.
- Construya sistemas que utilicen pequeños equipos de desarrollo.
- Subraye el desarrollo evolutivo o incremental mediante el uso de cortos ciclos de desarrollo.

Métodos

Los métodos de la IWeb abarcan un conjunto de labores técnicas que permiten al ingeniero Web comprender, caracterizar y luego construir una aplicación Web de alta calidad. Los métodos de la IWeb se pueden categorizar de la siguiente manera:

Métodos de comunicación:

Métodos de comunicación: Definen el enfoque con que se facilita la comunicación entre:

Las técnicas de comunicación son importantes durante la recolección de requisitos.

Métodos de análisis de requisitos:

proporcionan una base para comprender el contenido que se entregará a las WebApps, la función que proporcionará al usuario final y los modos de interacción de cada clase de usuario requerirá mientras ocurra la navegación por medio de las WebApps

Métodos de diseño:

Abarcan una serie de técnicas de diseño que abordan el contenido, la aplicación y la arquitectura de información, así como el diseño de interfase y la estructura de navegación de las WebApps.

Método de prueba:

Incorporan revisiones técnicas formales tanto de contenido y el modelo de diseño como de una amplia variedad de técnicas de pruebas que abordan conflictos al nivel de componentes. Pruebas de la navegación, pruebas de facilidad de uso , pruebas de seguridad y pruebas de configuración.

Herramientas y tecnologías

Las tecnologías abarcan un amplio conjunto de descripción de contenido y lenguaje de modelación por ejemplo: HTML, VRML, XML, etc. lenguajes de programación por ejemplo java, php, jsp, etc. recursos de desarrollo basados en componentes por ejemplo corba, com, activeX, .net, etc. navegadores, herramientas multimedia, herramientas de auditoria de sitio, herramientas de conectividad de base de datos, herramientas de seguridad, servidores y utilidades de servidor, y herramientas de administración y análisis de sitio.

3.3 EL PROCESO DE INGENIERÍA WEB

En un proceso tan rápido como es el proceso de Ingeniería Web, donde los tiempos de desarrollo y los ciclos de vida de los productos son tan cortos, ¿merece la pena el esfuerzo requerido por la gestión? La respuesta es que dada su complejidad es imprescindible. Entre los aspectos que añaden dificultad a la gestión destacamos: - alto porcentaje de contratación a terceros, - el desarrollo incluye una gran variedad de personal técnico y no técnico trabajando en paralelo,

- el equipo de desarrollo debe dominar aspectos tan variados como, software basado en componentes, redes, diseño de arquitectura y navegación, diseño gráfico y de interfaces, lenguajes y estándares en Internet, test de aplicaciones Web, etc, lo que hace que el proceso de búsqueda y contratación de personal sea arduo.

1.4.1 Definir el marco de trabajo

La efectividad de cualquier proceso de ingeniería depende de su adaptabilidad. Esto es, la organización del equipo de trabajo del proyecto, los modos de comunicación entre miembros del equipo, las actividades de ingeniería y las tareas que deben realizarse, la información que se recolecte y se cree, y los métodos empleados para producir un producto de alta calidad deben estar adaptados a la gente que realiza el trabajo, el plazo y las restricciones del proyecto, y al problema que se quiere resolver.

Antes de definir un marco de trabajo de proceso para la IWeb se debe reconocer que:

Las WebApps con frecuencia se entregan de manera incremental.

Los cambios ocurrirán frecuentemente.

Los plazos son cortos.

Comunicación con el cliente: La comunicación con el cliente se caracteriza por medio de dos grandes tareas: el análisis del negocio

y la formulación. El análisis del negocio define el contexto empresarial-organizativo para las WebApps y otras aplicaciones de negocio. La formulación es una actividad de recopilación de requisitos que involucran a todos los participantes.

Planeación:

Se crea el plan del proyecto para el incremento de la WebApp. El plan consiste de una definición de tareas y un calendario de plazos respecto al período establecido para el desarrollo del proyecto.

Modelado:

Las labores convencionales de análisis diseño de la ingeniería del software se adaptan al desarrollo de las WebApp, se mezclan y luego se funden en una actividad de modelado de la IWeb. El intento es desarrollar análisis rápido y

modelos de diseño que definan requisitos y al mismo tiempo representen una WebApp que los satisfará.

Construcción:

Las herramientas y la tecnología IWeb se aplican para construir la WebApp que se ha modelado. Una vez que se construye el incremento de WebApp se dirige a una serie de pruebas rápidas para asegurar que se descubran los errores en el diseño.

Despliegue:

Las WebApp se configuran para su ambiente operativo, se entrega a los usuarios finales y luego comienza un período de evaluación. La retroalimentación acerca de la evaluación para realizar los procesos respectivos.

Refinamiento del marco de trabajo

La definición de tareas de ingeniería requerida para refinar cada actividad del marco de trabajo se dejan a discrecional del juicio del equipo de ingeniería Web.

Es importante destacar que las tareas asociadas con la actividad del marco de trabajo IWeb pueden modificarse, eliminarse o extenderse con base en las características del problema, el producto, el proyecto y la gente en el equipo de ingeniería Web.

3.4 MEJORES PRÁCTICAS EN LA INGENIERÍA WEB.

Tomar tiempo para entender las necesidades del negocio y los objetivos del producto, incluso si los detalles de la WebApp son vagos.

Describir cómo interactuarán los usuarios con la WebApp aplicando un enfoque basado en escenarios.

Desarrollar un plan del proyecto, incluso si es muy breve

Utilizar algún tiempo para modelar lo que se construirá.

Revisar la consistencia y calidad de los modelos.

Utilizar herramientas y tecnología que permitan construir el sistema con tantos componentes reutilizables.

No apoyarse en usuarios anteriores para depurar la webApp

3.5 FORMULACIÓN DE SISTEMAS BASADOS EN WEB

La formulación se enfoca sobre el gran cuadro en las necesidades y objetivos del negocio y el la información relacionada. Los clientes y los ingenieros Web quieren definir el contenido requerido, discutir la funcionalidad específica, características específicas e identificar la forma en que los usuarios finales interactuarán con las WebApp, todo esto es formulación y recopilación de requisitos.

Preguntas de formulación:

- En esta etapa es bueno preguntarse y responderse lo siguiente:
- ¿Cuál es la principal motivación para las WebApp?
- ¿Cuáles son los objetivos que deben satisfacer las WebApp?
- ¿Quién usará la WebApp?

Si tenemos una respuesta correcta a estas preguntas implican metas específicas para el sitio Web.

En general se identificas dos categorías de metas:

- Metas informativas: indican una intención de proporcionar contenido información específicos al usuario final.
- Metas aplicables: indican la habilidad para realizar alguna tarea dentro de la WebApp.

Recopilación de requisitos para WebApps

Los objetivos globales de la recopilación de requisitos propuestos para la ingeniería de software permanecen inalterados. Adaptados para las WebApp, dichos objetivos se convierten en:

- Identificar requisitos de contenido.
- Identificar requisitos funcionales.
- Definir escenarios de interacción para diferentes clases de usuarios.

Los siguientes pasos de la recopilación de requisitos se dirigen para lograr estos objetivos

1. Pedir a los clientes que definan una categoría de usuario y describan cada categoría.
2. Comunicarse con los clientes para definir los requisitos básicos para las WebApp.
3. Analizar la información recopilada y utilizar la información para realizar un seguimiento a los clientes.
4. Definir casos de uso que describan escenarios de interacción para cada clase de usuario.

El puente hacia el modelado de análisis

Toda la información recolectada y tratada se modela con la utilización de casos de uso y notación UML.

3.6 PLANEACIÓN DE PROYECTOS DE INGENIERÍA WEB

Dada la inmediatez de las WebApps es razonable preguntar: ¿En realidad se necesita gastar tiempo en la planeación y administración de un esfuerzo WebApp? ¿No solo se debería dejar involucrar naturalmente a las WebApp, con poco o ninguna gestión explicativa?

Mas de un desarrollador Web optaría por poca o ninguna gestión, pero eso no hace que estén en lo correcto.

3.7 EL EQUIPO DE INGENIERÍA WEB

Un equipo de ingeniería Web exitoso mezcla una amplia variedad de talentos que deben trabajar como equipo en un ambiente de proyecto con alta presión. Los plazos son cortos, los cambios son inexorables y la tecnología continúa cambiando. La creación de un equipo que se consolide no es asunto sencillo.

- Los Actores
- Desarrolladores/proveedores de contenido
- Editores de Web
- Ingeniero Web
- Expertos en dominios empresariales
- Especialista en Soporte
- Administrador

- Construcción del equipo

Para la construcción del equipo tiene particular relevancia cuando una organización se forma un equipo de IWeb

- Se debe establecer un conjunto de directrices del equipo.
- El liderazgo fuerte es una obligación
- El respeto hacia los talentos individuales es crucial
- Cada miembro del equipo se debe comprender
- Es fácil comenzar, lo difícil es mantener el espíritu.

3.8 CONFLICTOS DE GESTIÓN DE PROYECTO PARA INGENIERÍA WEB

Una vez realizada la formulación y que se han identificado los requisitos básicos de la WebApp, la empresa debe elegir una de dos opciones de ingeniería Web:

- La WebApp es subcontratada: la ingeniería Web la realiza un tercer proveedor con experiencia, talento y recursos con los cuales no cuenta la empresa.
- Las WebApp la desarrollan en casa ingenieros Web que sean empleados de la empresa.

Planeación de WebApp: subcontratación

En este caso un negocio(el cliente) pide un precio fijo para desarrollar la WebApp de uno o más proveedores, evalúa los precios competitivos y luego elige un proveedor para efectuar el trabajo. Pero ¿Qué busca la organización contratante? ¿Cómo se determina la competencia de un proveedor de WebApp? ¿Cómo se sabe si una cotización es razonable?

¿Qué grado de planeación, programa de trabajo y valoración de riesgos se pueden esperar conforme una organización se embarca en un esfuerzo por desarrollar una gran WebApp?

Estas preguntas no siempre son fáciles de contestar, pero vale la pena considerar algunos lineamientos.

Selección de los subcontratistas candidatos:

Con la finalidad de elegir desarrolladores Web candidatos, el contratante debe realizar algunas tareas obligadas:

- Entrevistar a los clientes antiguos para determinar el profesionalismo del vendedor Web, así como su habilidad para cumplir con compromiso de plazos y costos y su destreza para comunicarse efectivamente.
- Determinar el nombre de ingeniero(s) Web jefe de la empresa subcontratista para buscar proyectos anteriores exitosos.
- Examinar cuidadosamente ejemplos de trabajo del subcontratista que sean similares en apariencia y sentido a la WebApp que será contratada.

Planeación de WebApp: Ingeniería Web en casa

Los pasos siguientes se recomiendan para proyectos IWeb pequeños y de tamaño moderado.

- Entender el ámbito, las dimensiones del cambio y las restricciones del proyecto.
- Definir una estrategia del proyecto incremental
- Realizar análisis de riesgo
- Desarrollar una estimación rápida
- Elegir un conjunto de tareas
- Establecer un programa
- Definir mecanismos de rastreo del proyecto
- Establecer un enfoque de gestión de cambios

3.9 MEDIACIÓN PARA INGENIERÍA WEB Y WEBAPPS

La mediación del software ofrece una base para mejorar el proceso de software, lo que aumenta la precisión de las estimaciones del proyecto, incrementa el rastreo del proyecto y mejora la calidad del software.

Las mediciones tienen tres metas principales:

- Proporcionar un indicador de calidad de las WebApp desde el punto de vista técnico.

- Proporcionar una base para la estimación del esfuerzo
- Proporcionar una indicación del éxito de la WebApp desde el punto de vista empresarial.

UNIDAD 4: ANÁLISIS DE APLICACIONES WEB

4.1 REQUISITOS PARA EL ANÁLISIS DE LAS WEBAPP

El análisis de requisitos para las WebApp abarca tres grandes tareas:

- Formulación
- Recopilación de requisitos
- Modelado de análisis

La jerarquía de usuario

Las categorías de los usuarios finales se identifican como parte de las tareas de formulación y de recopilación de requisitos. Las categorías de usuarios son relativamente limitados y no necesitan una representación UML. Sin embargo, cuando crece el número de categorías de usuario, a veces es aconsejable desarrollar una jerarquía de usuarios.

Desarrollo de casos de uso

Los casos de uso se desarrollan para cada categoría de usuario descrita en la jerarquía de usuario. En el contexto de la ingeniería Web, el caso de uso en si mismo es relativamente informal: un párrafo narrativo que describe una interacción específica entre el usuario y la WebApp.

Afinación del modelo de caso de uso

A la par que se crean los diagramas de casos de uso para cada categoría de usuario, se desarrolla una vista superior de los requisitos de la WebApp observables de manera externa.

Cada paquete se valora para garantizar que es:

- Comprensible.
- Cohesivo
- Libremente acoplados
- Jerárquicamente superficial.

Puesto que el análisis y modelado de actividades son actividades iterativas.

4.2 EL MODELADO DE ANÁLISIS PARA WEBAPPS

Se basa en la información que contienen los casos de uso desarrollados para la aplicación. Se identifican el contenido que presentará la WebApp y se extraen las funciones que se desarrollarán a partir de las descripciones de caso de uso. Finalmente los requisitos específicos de la implementación se deben desarrollar de modo que el ambiente y la infraestructura que apoyan la WebApps puedan construirse.

Cuatro actividades de análisis, cada una con su aporte a la creación de un modelo de análisis completo son:

- Análisis de contenido
- Análisis de interacción
- Análisis de funciones
- Análisis de configuración

El modelo es si mismo contiene elementos estructurales y dinámicos. Los elementos estructurales identifican las clases de análisis y los objetivos de

contenido que se requieren para crear una WebApp que satisfaga las necesidades del cliente.

Los elementos dinámicos del modelo de análisis describen como interactúan los elementos estructurales, entre ellos y con los usuarios finales.

4.3 EL MODELO DE CONTENIDO

El modelo de contenido contiene elementos estructurales que proporcionan una importante visión de los requisitos de contenido para una WebApp. Además incluye todas las clases de análisis: entidades visibles para el usuario que se crean o manipulan conforme éste interactúa con la WebApp.

Al igual que otros elementos del modelo de análisis, el modelo de contenido se deriva a partir de un examen cuidadoso de los casos de uso desarrollados para la WebApp.

Definición de objetos de contenido

Un objeto de contenido puede ser una descripción textual de un producto, un artículo que describa un evento noticioso.

Los objetos de contenido se extraen en los casos de uso al examinar la descripción del escenario para referencias directas e indirectas al contenido.

Relaciones de jerarquía de contenido

El modelo de contenido puede contener diagramas de relación de entidades o árboles de datos que bosquejan las relaciones entre los objetos de contenido o la jerarquía de éste que mantiene una WebApp.

4.4 EL MODELO DE INTERACCIÓN

Este modelo de interacción lo comprenden cuatro elementos:

- Casos de uso
- Diagramas de secuencia
- Diagramas de estado
- Prototipo de interfaz de usuario

Casos de Uso

Un caso de uso se modela para todos los procesos que la WebApp debe llevar a cabo. Los procesos se describen dentro del caso de uso por una descripción textual o una secuencia de pasos ejecutados. Los Diagramas de Actividad se pueden usar también para modelar escenarios gráficamente. Una vez que el comportamiento de la aplicación está captado de esta manera, los casos de uso se examinan y amplían para mostrar qué objetos se interrelacionan para que ocurra este comportamiento.

Diagrama de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

Diagramas de Estado

El comportamiento en tiempo real de cada clase que tiene comportamiento dinámico y significativo, se modela usando un Diagrama de Estado. El diagrama de actividad puede ser usado también aquí, esta vez como una extensión del diagrama de estado, para mostrar los detalles de las acciones llevadas a cabo por los objetos en respuesta a eventos internos. El diagrama de actividad se puede usar también para representar gráficamente las acciones de métodos de clases.

Prototipo de interfaz de usuario

Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

Puesto que las herramientas de desarrollo de la WebApp son abundantes y funcionalmente poderosas, es mejor crear prototipos de la interfaz mediante tales herramientas.

4.5 EL MODELO FUNCIONAL

Este modelo funcional aborda dos elementos de procesamiento de la WebApp y cada uno representa un grafico diferente de la abstracción de procedimiento:

- Funcionalidad observable respecto al usuario y que entrega al usuario final de WebApp.
- Las operaciones dentro de las clases de análisis que implementan comportamientos asociados con la clase.

La funcionalidad observable para el usuario comprende cualesquiera funciones de procesamiento que éste indicia directamente. Por ejemplo, un sitio Web financiero puede implementar una variedad de funciones financieras.

4.6 EL MODELO DE CONFIGURACIÓN

Las WebApps se deben diseñar e implementar de forma que se acomoden a una diversidad de ambientes, tanto del lado del servidor como del cliente. Se deben especificar el hardware del servidor y el ambiente del sistema operativo. Si las WebApp deben tener acceso a una gran base de datos o ínter operar con las aplicaciones corporativas existentes en el lado del servidor, se deben especificar las interfaces apropiadas, los protocolos de comunicación y la información.

Las WebApp deben someterse a una amplia prueba de cada configuración de navegador que se especifique como parte del modelo de configuración.

Análisis relación – navegación

El análisis relación-navegación proporciona una serie de pasos de análisis que luchan por identificar relaciones entre los elementos descubiertos como parte de la creación del modelo de análisis.

El enfoque de ARN se organiza en cinco pasos:

- Análisis de los participantes

- Análisis de los elementos
- Análisis de relaciones
- Análisis de navegación
- Análisis de evaluación.

Análisis de relaciones: preguntas claves

En este análisis se formulan una serie de preguntas que nos ayudará a comprender mas la relación, para ello debe acudir al libro guía pagina 516.

Análisis de navegación

Uno de los aspectos más importantes en los sistemas de información en las WebApp es el de la navegación. La gran mayoría de las propuestas metodológicas para sistemas WebApp resaltan este aspecto ofreciendo modelos que permitan diseñarlo e implementarlo asegurando la calidad del resultado. Sin embargo, analizando dichos modelos y técnicas y viendo los resultados de diferentes estudios comparativos, se puede observar que este aspecto, en la mayoría de las propuestas, se trata solamente en las últimas fases del ciclo de vida, principalmente en diseño e implementación.

Los mecanismos de navegación se definen como parte del diseño. En esta etapa, los desarrolladores deben considerar requisitos de navegación globales.

UNIDAD 5: DISEÑO PARA APLICACIONES WEB

5.1 TEMAS DE DISEÑO PARA INGENIERÍA WEB

Cuando se aplica el diseño dentro del contexto de la ingeniería Web, se deben considerar cuestiones tanto genéricas como específicas. El diseño debe ser muy específico.

Diseño de calidad de una WebApp

En la práctica, la calidad de los sitios web se ha evaluado de una manera “ad hoc”, basada en el sentido común y en la experiencia de los desarrolladores. El estudio de la calidad de productos y procesos de desarrollo para la Web es muy reciente y todavía no se dispone de métodos de evaluación ampliamente difundidos para este tipo de entorno.

Por lo tanto, existe la gran necesidad de metodologías efectivas para la obtención de aplicaciones Web de calidad. Actualmente, hay dos vertientes de metodologías de desarrollo para la Web: la comunidad de Ingeniería de Software y la comunidad de Hipermedia. Estas metodologías carecen sin embargo de métricas que puedan ser aplicadas a los modelos intermedios (Ej. Modelo de Objetos, Modelo Navegacional) utilizados en el proceso de desarrollo orientado a la Web.

Otras características de diseño de calidad Web son:

- Seguridad
- Disponibilidad
- Escalabilidad
- Tiempo en el mercado

Metas de diseño

Una de las metas de diseño de una WebApp es abarcar lo siguiente:

- Simplicidad
- Consistencia
- Identidad
- Robustez
- Navegabilidad
- Apariencia visual
- Compatibilidad

5.2 PIRÁMIDE DEL DISEÑO IWEB

El diseño conduce a un modelo que contiene una mezcla adecuada de estética, contenido y tecnología.



Ilustración 5 Pirámide de diseño para la Ingeniería Web.

5.3 DISEÑO DE LA INTERFAZ DE LA WEBAPP

Toda interfaz de usuario debe presentar las siguientes características:

- Fácil de usar
- Fácil de aprender
- Fácil de navegar
- Intuitiva
- Consistente
- Eficiente
- Libre de errores
- Funcional

Debe ofrecer al usuario final una experiencia satisfactoria y gratificante.

Los principios y directrices esenciales del diseño de una WebApp se pueden mencionar:

- Uso equitativo
- Flexibilidad en el uso
- Uso sencillo e intuitivo
- Información perceptible
- Tolerancia al error
- Esfuerzo físico reducido

- Tamaño y espacio para acercarse y usar

En un momento de nuestra vida, cualquiera de nosotros puede llegar a experimentar una transición física. Ya sea una minusvalía temporal, como un hueso roto, o una discapacidad permanente, nos veremos enfrentados al cambio.

El simple proceso de envejecimiento nos lleva a ser dependientes de otros. El diseño universal nos permite transitar por esos cambios y seguir disfrutando de igualdad de oportunidades, autodeterminación y calidad de vida.

Técnicas de diseño

Diseño fijo

Diseño que utiliza dimensiones de tamaño absoluto: inches (in), centímetros (cm), milímetros (mm), puntos (pt), picas (pc). Su efecto es bloquear el tamaño de un elemento a un tamaño fijo.

Diseño líquido (o fluido)

Diseño que utiliza cómo unidad de medida el porcentaje (%). Su efecto es la adaptación de los elementos y su disposición aprovechando todo el ancho de pantalla que será calculada en base a la resolución del dispositivo utilizado.

Diseño elástico (o flexible)

Diseño que utiliza cómo unidad de medida em. Su efecto es adaptar el ancho de los elementos y su disposición en base al tamaño de texto configurado por el usuario.

Diseño híbrido

Diseño que utiliza unidades de medida absolutas, porcentaje y em.

- No hay un diseño más accesible que otro.
- Se elije el tipo de diseño en base a las necesidades (proyecto).

Flujo de trabajo en el diseño de la interfaz

Las siguientes tareas representan un flujo de trabajo rudimentario para el diseño de la interfaz WebApp.

1. Revisar la información contenida en el modelo de análisis y refinarla conforme se requiera.

2. Desarrollar un bosquejo aproximado de la plantilla de la interfaz de la WebApp.
3. Correlacionar los objetivos del usuario con acciones específicas de la interfaz.
4. Definir un conjunto de tareas de usuario que estén asociadas con cada acción.
5. Elaborar bosquejos con imágenes de la pantalla para cada acción de la interfaz.
6. Refinar la plantilla de la interfaz y los bosquejos con el uso de entradas desde el diseño estético.
7. Identificar los objetos de la interfaz del usuario que se requieren para implementarla.
8. Desarrollar una representación de procedimiento de la interacción del usuario con la interfaz.
9. Desarrollar una representación del comportamiento de la interfaz.
10. Describir la plantilla de la interfaz par acada estado.
11. Refinar y revisar el modelo de diseño de la interfaz.

5.4 DISEÑO ESTÉTICO

El diseño estético, también llamado diseño grafico, es un esfuerzo artístico que complementa los aspectos técnicos de la ingeniería Web. Sin él una WebApp puede ser funcional, pero sin atractivo.

5.5 DISEÑO DE CONTENIDO

El diseño de contenido se enfoca en dos asuntos de diseño diferentes, cada uno lo abordan individuos con distintos conjuntos de habilidades. El diseño de contenido desarrolla una representación de diseño para los objetos de contenido y representa los mecanismos que se requieren para que establezcan sus relaciones uno con otro.

Objetos de contenido

La relación entre objetos de contenido, defina como parte del modelo de análisis WebApp y los objetos de diseño que representan contenido es análoga a la relación entre las clases de análisis y los componentes de diseño.

Un objeto de contenido tiene atributos que incluyen información específica de contenido y atributos específicos de implementación que se especifican como parte del diseño.

5.7 DISEÑO DE NAVEGACIÓN

El diseñador debe definir las rutas de navegación que habiliten para la ruta de los usuarios el acceso al contenido y las funciones de las WebApp. Para ello se debe:

- Identificar la semántica de navegación para diferentes usuarios del sitio
- Definir la mecánica que logra la navegación.

Semántica de navegación

Durante las etapas iniciales del diseño, de navegación se valora la arquitectura de contenido de la WebApp para determinar una o mas FdN para cada caso de uso. Como se anotó anteriormente, una FdN identifica los nodos de navegación y los vínculos que permiten la navegación entre ellos.

Sintaxis de navegación

Conforme el diseño se lleva a cabo se define la mecánica de navegación, entre sus posibilidades tenemos:

- Vinculo de navegación individual
- Barra de navegación horizontal
- Columna de navegación vertical
- Pestañas
- Mapas del sitio

5.8 METRICAS DE DISEÑO PARA WEBAPPS

Las métricas deben ofrecer respuestas cuantitativas a las siguientes preguntas

¿La interfaz de usuario promueve la facilidad de uso?

¿La estética de la WebApp es apropiada para el dominio de la aplicación y confortable al uso?

¿El contenido esta diseñado en una forma que proporciona mayor informacion con el menor esfuerzo?

¿La navegación es eficiente y directa?

¿La arquitectura de la WebApp se ha diseñado para acomodar las metas y objetivos especiales de los usuarios de la WebApp la estructura de contenido y funcionalidad y el flujo de navegación requerido para usar el sistema de manera efectiva?

¿Los componentes estándar diseñado en una forma que reduce la complejidad y aumenta la exactitud, la confiabilidad y el desempeño?

Hasta el momento no existe un conjunto de métricas cuantitativas por lo que estos deben ser tratados de manera cualitativa

UNIDAD 6: VERIFICACIÓN Y VALIDACIÓN DE APLICACIONES WEB

6.1 PLANIFICACIÓN DE LA VERIFICACIÓN Y VALIDACIÓN

- * La verificación y validación es un proceso caro.
- * Para algunos sistemas, más de la mitad del presupuesto para desarrollar el sistema se invierte en V & V.
- * Es necesaria una planificación cuidadosa para obtener el máximo provecho de las inspecciones y pruebas y controlar los costos del proceso de V & V.
- * Debe comenzarse la planificación de la V & V del sistema en etapas tempranas del proceso de desarrollo.
- * Dentro del proceso de planificación de V & V, hay que buscar un equilibrio entre las aproximaciones estáticas y dinámicas de V & V.
- * También hay que pensar procedimientos para las inspecciones y pruebas del software, establecer listas de comprobación para conducir las inspecciones de programas y definir el plan de pruebas del software.

- * El esfuerzo destinado a las inspecciones y las pruebas depende del tipo de sistema a desarrollar y de los expertos de la organización en V & V con que se cuente.
- * Como regla general, cuanto más crítico sea el sistema, debe dedicarse más esfuerzo a las técnicas de verificación estáticas.
- * Los planes de pruebas, además de ayudar a asignar recursos y estimar el calendario de pruebas, son útiles para los ingenieros del software implicados en el diseño y la realización de las pruebas del sistema.
- * Ayudan a obtener una panorámica general de las pruebas del sistema y ubicar su propio trabajo en este contexto.

6.2 INSPECCIONES DE SOFTWARE

- * Proceso de V & V estático donde un software se revisa para encontrar errores, omisiones y anomalías.
- * Las inspecciones se suelen centrar en el código fuente, pero puede inspeccionarse cualquier representación legible del software.
- * Hay 3 ventajas fundamentales de la inspección sobre las pruebas:

Durante las pruebas, los errores pueden ocultar otros errores.

- * Cuando se descubre un error, nunca se puede estar seguro de si otras anomalías de salida son debidas a un nuevo error o son efectos laterales del error original.
- * Como la inspección es un proceso estático, no hay que preocuparse de las interacciones entre errores.
- * Así, una única sesión de inspección puede descubrir muchos errores en un sistema.

Pueden inspeccionarse versiones incompletas de un sistema sin costos adicionales.

- * Si un programa está incompleto, entonces se necesita desarrollar software de pruebas, para probar las partes disponibles.
- * Esto, obviamente, añade costos al desarrollo del sistema.

Además de buscar los defectos en el programa, una inspección también puede considerar otros atributos de calidad más amplios: cumplimiento con los estándares, portabilidad y mantenibilidad.

- * Puede buscarse ineficiencias, algoritmos no adecuados y estilos de programación que podrían tornar al sistema difícil de mantener y actualizar.
- * Muchos autores observaron que la revisión estática de código era más efectiva y menos costosa que las pruebas de defectos, a la hora de encontrar fallas en los programas.

6.3 PRUEBAS DE LAS APLICACIONES WEB

Las pruebas que por cierto son de diferentes tipos.

- El contenido: evaluación sintáctica (ortografía y gramática de contenido) y semántica (exactitud de la información presentada y consistencia entre objetos de contenido y objetos relacionados)
- La función: comprobar la concordancia con los requisitos del cliente
- Estructura: se comprueba que se entregue adecuadamente el contenido y la funcionalidad de la WebApp.
- Facilidad de uso: para cada categoría de usuario, que va relacionado con la sintaxis y semántica de la navegación
- Navegabilidad: comprobar la sintaxis y semántica de navegación así como comprobar que esté controlado todo flujo, por ejemplo: vínculos rotos, vínculos inadecuados, etc.)
- Desempeño: se prueba en una diversidad de condiciones operativas, configuraciones, y cargas para saber que el software responde adecuadamente a cargas extremas
- Compatibilidad: se comprueba que la WebApp funciona en varias configuraciones huésped tanto del lado del cliente como del servidor e infraestructura de seguridad
- Interoperabilidad: se prueba para asegurar que la WebApp realiza interfaces adecuadas con otras aplicaciones y/o bases de datos, y que estas sean con un buen desempeño
- Seguridad: se prueba la vulnerabilidad potencial.

EJEMPLOS DE PRUEBAS PARA WEBAPPS

Prueba de Contenido

- Sintáctica (ortografía y gramática de contenido)
- Semántica (exactitud de la información presentada y consistencia entre objetos de contenido y objetos relacionados).

Esta evaluación que se hace para cada objeto de contenido debe el examinador responder a las siguientes preguntas:

- ¿La información realmente es precisa (concisa y exacta)?
- ¿La plantilla del objeto de contenido es fácil de entender para el usuario?
- ¿La información anidada en un objeto de contenido se encuentra con facilidad?
- ¿La información presentada internamente es consistente con la información de otros objetos de contenido?
- ¿El contenido es ofensivo, engañoso o abre la puerta a pleitos?
- ¿El contenido infringe derechos de autor o marcas registradas?
- El estilo estético del contenido entra en conflicto con el estilo estético de la interfaz global

Un grupo de pruebas adicionales en cuanto a la semántica es para los WebApp que construyen objetos de contenido dinámicos que se crean en tiempo real en base a una base de datos. Para esto se deben observar varios factores:

- Pruebas que descubran errores cometidos al traducir la solicitud del usuario en una forma que pueda procesar la DBMS
 - Pruebas para comprobar errores en la comunicación con la base de datos
 - Validar que los datos brutos provenientes de la DBMS hacia el servidor y este hacia el cliente estén correctamente formateados para crear los objetos de contenido
 - Probar el formato y compatibilidad de presentación de los objetos de contenido dinámico al usuario con diferentes configuraciones de ambiente del cliente
- Estructurales/arquitectura (que se entregue adecuadamente el contenido y la función de la WebApp. Que pueda ser extensible en contenido y funcionalidad) CMS.

Prueba de Interfaz del Usuario

La verificación y validación de la interfaz del usuario se lo hace en tres puntos del proceso, en el análisis (formulación y análisis de requisitos), en el diseño (al diseñar la interfaz que garantice la calidad) y durante las pruebas donde básicamente se prueban ejecutando la aplicación.

Los objetivos que se obtienen con las pruebas de interfaz son:

- Las características de la interfaz se prueban para asegurar que las reglas del diseño, la estética y el contenido visual relacionado están a disposición del usuario sin error alguno
- Los mecanismos individuales de la interfaz se prueban en forma unitaria (por ejemplo se prueba HTML dinámico, cgi, carrito de compras, etc.)
- La interfaz se prueba frente cada caso de uso (USN) para descubrir errores de semántica, y facilidad de uso
- La interfaz se prueba dentro de una diversidad de ambientes para asegurar su compatibilidad.

Pruebas de semántica de la interfaz. Una vez que se ha probado de forma unitaria cada mecanismo de interfaz, se procede con la prueba semántica que evalúa cuán bien el diseño se ocupa de los usuarios, ofreciendo una dirección clara, manteniendo consistencia de lenguaje y enfoque.

Para esto se sugiere probar cada caso de uso (categorías de usuario) versus el diseño de la interfaz para confirmar que no hay errores y si los hubiere como los maneja y su recuperación.

Pruebas de compatibilidad. Como sabemos, el usuario puede tener diferentes ambientes (sistemas operativos, navegador, plataformas de hardware), por lo que se debe hacer pruebas de compatibilidad para descubrir errores asociados con un ambiente específico. Para esto se debe definir un conjunto de configuraciones de computadoras cliente más comunes y en función de estas armar las pruebas.

Prueba al nivel de Componentes

Conocidas como pruebas de función, que intentan descubrir errores en la funcionalidad de la WebApp. Para esto existen algunos métodos de prueba:

- Partición de equivalencia. Cuyo dominio de entrada se divide en categorías o clases de entrada y a partir de estos se derivan los casos de prueba. Se hacen pruebas con cada categoría mientras las otras permanecen constantes, por ejemplo en una página que importe el código postal, aquí se hace pruebas posibles con el código postal mientras los otros valores permanecen constantes
- Análisis de valores límite. Probar los valores máximos o mínimos de campos o atributos que maneja el componente o la función
- Pruebas de ruta. Para comprobar y garantizar que se ha ejercitado toda ruta posible en su funcionamiento.

Para cada caso se debe especificar la entrada y salida esperada.

Existen circunstancias en las que la funcionalidad es tan extensa que complica probar todas las posibles situaciones, por lo que una sugerencia sería acudir a un análisis de riesgo para obtener los casos de prueba de mayor riesgo, como por ejemplo:

- ¿Cuál funcionalidad es la más importante para el propósito
- ¿Cuales áreas del sitio amerita más dura interacción con la DB?
- ¿Cuales mecanismos específicos (cgi, applets, ActiveX) son los más complejos?
- ¿Qué tipos de problemas causara la mayoría de las quejas o la peor publicidad?
- ¿Qué áreas del sitio serán las más populares?
- ¿Qué partes del sitio tendrán mayores riesgos de seguridad?

Pruebas de Navegación

Las pruebas de navegación garantizan que todos los mecanismos que permiten al usuario de la WebApp viajar a través de ella sean funcionales y validar que cada USN pueda ser alcanzado por la categoría de usuario adecuada.

La primera prueba de navegación inicia con la prueba de interfaz. Estas pruebas inicialmente son probadas por los Ingenieros Web para luego pasar que prueben los usuarios.

Prueba de sintaxis de navegación. Con cada mecanismo de navegación se prueba: vínculos navegación, redirigir (destinos removidos), mapas de sitio, motores de búsqueda.

Prueba de semántica de navegación. Se deben probar los USN (unidad semántica de navegación). Si no se han hecho los USN en el análisis, se puede tomar los casos de uso para armar casos de prueba de navegación respondiendo a las siguientes preguntas:

Pruebas de Configuración

En las pruebas de compatibilidad en pruebas de interfaz usuario se probaron solo del lado del cliente, mientras que en estas pruebas de configuración se prueban un conjunto de probables configuraciones de cliente y servidores.

Del lado del servidor:

- ¿La WebApp es totalmente compatible con el sistema operativo del servidor?
- ¿Los archivos de sistema, directorios y datos de sistema relacionados se crean correctamente cuando la WebApp esta operativa?
- ¿Las medidas de seguridad del sistema permiten a la WebApp ejecutarse y dar el servicio a los usuarios sin interferencia y a un buen desempeño?
- ¿La WebApp se ha probado con la configuración de servidor distribuido (clúster o balanceo de carga)?
- ¿La WebApp está integrada adecuadamente con la base de datos (incluyendo diferentes versiones de DB)?
- ¿Si se usan proxys, las diferencias en configuraciones se han abordado?

Del lado del cliente:

- Hardware: CPU, memoria, dispositivos de impresión, monitor
- Sistemas operativos: normales y especiales para equipos móviles
- Software de navegación
- Componentes de interfaz del usuario: ActiveX, applets java, otros
- Plug in: QuickTime, real player, adobe, otros
- Conectividad: cable, DSL, Modem, proxys, otros
- Aplicaciones que corren al mismo tiempo.

Pruebas de Desempeño

Se aplican para descubrir problemas de desempeño que se presentan debido a la falta de recursos en el lado del servidor, ancho de banda de red inapropiada, capacidades inadecuadas de bases de datos, capacidades/defectuosos de sistemas operativos, funcionalidad mal diseñada, u otros conflictos de hardware o software (seguridad) que impidan un buen desempeño.

Esto conducirá a comprender como el sistema responde a cierta carga (numero de usuarios activos, numero de transacciones, volumen de datos, etc.) y a poder recolectar métricas para poder modificar el diseño o acoplar soluciones (hardware/software apropiadas) para mejorar el rendimiento.

Las pruebas deben responder a las siguientes preguntas:

- ¿En qué punto (usuarios, transacciones, carga de datos) el desempeño se vuelve inaceptable?
- ¿Que componentes del sistema son responsables de la reducción del desempeño?
- ¿Cuál es el tiempo de respuesta promedio para los usuarios en una variedad de condiciones de carga?
- ¿Que ocurre cuando se aplican cargas que rebasan la capacidad máxima del servidor?

Para contestar a estas preguntas se hacen las pruebas de desempeño comunes (carga y tensión).

Pruebas de carga. Para esta prueba se consideran variables:

- N: número de usuarios concurrentes
- T: numero de transacciones en línea por usuario por unidad de tiempo
- D: la carga de datos procesada por el servidor por transacción.

De estas variables el Ingeniero Web obtiene valores promedios aceptables y si en el momento de la prueba encuentra una disminución precipitada de desempeño se debe revisar el porqué.

Pruebas de tensión. Es una continuación de la prueba de carga, en donde las variables N, T y D se fuerzan para poder superar los límites operativos. Con esta prueba se responden a las siguientes preguntas:

- ¿El sistema se degrada o el servidor se desconecta cuando se rebasa su capacidad?
- ¿El software emite mensajes al cliente y al soporte técnico del sitio que se no puede alcanzar el servidor?
- ¿El servidor pone en cola las peticiones que no alcanza a resolverlas para ir atendiendo poco a poco como su capacidad lo permita?
- ¿Las transacciones se pierden conforme se rebasa los límites?
- ¿La integridad de datos se pierde cuando se rebasa los límites?
- ¿Si el sistema falla, cuánto tarda en estar en línea de nuevo?

- ¿Ciertas funciones del WebApp se discontinúan cuando el servidor llega a sus umbrales (memoria, procesador, disco)?

Una variación de pruebas de tensión son las llamadas pruebas pico, que son las mismas pruebas de tensión pero se lo hace muy repentino llegar o sobrepasar el límite volviendo a la normalidad y luego nuevamente sobrepasando el límite para efectos de poder comprobar cuán bien que el sistema/servidor se recupera regresando a su normalidad.