



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**GUÍA PARA EL DESARROLLO DE
APLICACIONES WEB BASADA EN EL
MODELO DE PROCESO PARA LA
INDUSTRIA DE SOFTWARE.**

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA (COMPUTACIÓN)

P R E S E N T A:

RUBÉN SCHAFFER LEVINE

DIRECTORA DE LA TESIS: DRA. HANNA OKTABA

MÉXICO, D.F.

2009.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**GUÍA PARA EL DESARROLLO DE
APLICACIONES WEB BASADA EN EL
MODELO DE PROCESO PARA LA
INDUSTRIA DE SOFTWARE.**

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA (COMPUTACIÓN)

P R E S E N T A:

RUBÉN SCHAFFER LEVINE

DIRECTORA DE LA TESIS: DRA. HANNA OKTABA

MÉXICO, D.F.

2009.

Índice

1	Introducción.....	4
1.1	Objetivo:	4
1.2	Antecedentes:.....	4
1.3	Motivación:	5
1.4	Metodología:.....	5
1.5	Estructura del trabajo:.....	7
2	Marco teórico.....	8
2.1	Componentes y estructura de los sistemas WEB	8
2.2	Características de los sistemas WEB.....	12
C1	Comunicación	12
C2	Interacción	12
C3	Estética y diseño gráfico preponderantes	12
C4	Intuitivos y autoexplicativos	13
C5	Vinculados	13
C6	Cambio continuo	13
C7	Tecnología diversa.....	13
C8	Usuarios simultáneos y diversos.....	13
C9	Múltiples involucrados con múltiples áreas de especialización.....	14
2.3	Dificultades en los desarrollos de sistemas WEB	15
D1	Necesidades de usuarios frecuentemente relegadas	15
D2	Requisitos altamente cambiantes	15
D3	Dificultades de comunicación	15
D4	Tecnología diversa y cambiante	15
3	Proceso de desarrollo WEB	17
3.1	Técnicas para desarrollo WEB.....	19
T1	Análisis de requerimientos enfocado a objetivos	19
T2	Prototipos Grises	21
T3	Prototipos operativos	21
T4	Técnicas para modelación	22
T5	Uso de patrones	23
T6	Técnicas para el diseño de arquitecturas	23

T7 Técnicas de Diseño gráfico	24
T8 Técnicas para el diseño de navegación	25
T9 Técnicas para presentar la información.....	26
T10 Técnicas para probar las aplicaciones WEB.....	26
4 MoProSoft	28
4.1 Introducción a MoProSoft.....	28
4.2 Categorías	29
4.3 Procesos	30
4.4 Niveles.....	31
4.5 MoProSoft y la guía de desarrollo WEB	31
5 Construcción de la guía	32
5.1 Proceso	32
5.2 Estructura	33
5.3 Notación, cambios y formato.....	35
5.4 Validación	37
6.- Conclusiones.....	38
7.- Trabajo a futuro	39
Referencias:	40
Anexo A - "Guía para el desarrollo de aplicaciones WEB. Basada en el modelo de procesos para la industria de software MoProSoft."	42

1 Introducción

1.1 Objetivo:

El presente trabajo de tesis consiste en construir una guía, basada en los procesos de Administración de Proyectos Específicos y Desarrollo y Mantenimiento de software de MoProSoft[1], que integre técnicas especializadas en desarrollo de sistemas WEB. Misma que estará enfocada en apoyar a equipos pequeños, de desarrollo de aplicaciones WEB, en formación. También puede ser útil a equipos que tienen metodologías de trabajo totalmente empíricas y desean migrar a la aplicación de modelos de procesos, en particular a MoProSoft.

1.2 Antecedentes:

Por su ubicación geográfica, su rol en Latinoamérica, el nivel educativo y la importancia de la industria en sí, el desarrollo de la industria de software es una oportunidad para México. Sin embargo, más del 90% de las empresas en el país y dicha industria son pequeñas o medianas. La mayoría de las empresas de este tipo tienen grandes dificultades para subsistir, pues no pueden mantener esquemas de operación competitivos. Bajo este contexto el gobierno y la industria de software mexicanos crearon el programa PROSOFT[11] para fomentar, organizar y canalizar proyectos e iniciativas que mejoren las circunstancias y oportunidades de las empresas de esta industria.

Una de las mayores deficiencias en la industria mexicana de software es la implementación de procesos de trabajo que faciliten la producción de software a tiempo dentro del presupuesto planeado, y al costo estimado. Esto se debe a que mayoría de las normas, estudios y estrategias para lograr procesos eficientes y eficaces han sido desarrollados en países extranjeros para empresas de mayor tamaño. Para resolver este problema, uno de los proyectos más importantes es el Modelo de Procesos de la industria de Software (MoProSoft), desarrollado a iniciativa de PROSOFT. MoProSoft se creó con el propósito de dar una norma de operación a la industria mexicana de software. Adaptó las mejores prácticas para el desarrollo de software reflejadas en los diversos modelos y normas internacionales a las condiciones específicas de la industria de software mexicana.

MoProSoft ha probado ser un modelo sólido y adecuado a su propósito al grado que será tomado como base tanto por la organización internacional de estándares ISO[12], como por el programa iberoamericano COMPETISOFT[13], para desarrollar modelos enfocados en la pequeña y mediana empresa. Sin embargo el tener el modelo de procesos es sólo el primer paso para lograr que la industria de software opere con procesos eficientes. Es

necesario que dicho modelo de procesos sea conocido y adoptado por las empresas. Para esto pueden ayudar diferentes acciones, como son la formación de consultores y certificadores, o la creación de guías para facilitar la adopción del modelo de procesos dentro de las empresas.

Dentro de la industria del software una de las áreas más importantes es el desarrollo de sistemas de Internet. Esta red global de computadoras ha revolucionado la forma en que utilizamos las computadoras. La posibilidad de comunicar una computadora a otras ha multiplicado exponencialmente las posibilidades que estas herramientas nos brindan para organizar, crear, procesar y divulgar información. Una de las formas más comunes de transferir información por Internet son los sitios WEB, que están formados por un grupo de páginas en lenguaje de marcado de hipertexto (HTML por sus siglas en inglés). Las páginas se alojan en una computadora (servidor) permanentemente conectada a Internet y pueden ser accedidas por usuarios desde otras computadoras utilizando un navegador de Internet y la dirección electrónica (o URL) del servidor.

Anué técnicamente un sistema WEB es todo aquel sistema que utiliza internet para funcionar, comúnmente y en esta tesis se utiliza el término sistemas WEB se utiliza para referirse a sitios WEB cuyos contenidos e información son procesados automáticamente por diversos programas, según las solicitudes del usuario.

1.3 Motivación:

La adopción de MoProSoft es importante para dar a la industria de software Mexicana mejores niveles de eficiencia y calidad pues al generar un lenguaje y forma de trabajo organizada y común entre diferentes empresas permite armar cadenas de valor más exitosas. Por otra parte Internet está revolucionando todos los aspectos de la vida por lo que se ha generado una gran demanda de desarrollos de sistemas WEB Dada la importancia de ambos la presente tesis y la guía resultante que se presenta como anexo, se generaron para ayudar a fomentar la convergencia del desarrollo de sistemas WEB y MoProSoft.

1.4 Metodología:

La guía se construyó combinando la investigación bibliográfica con la asesoría de dos académicos expertos: Dra. Hanna Oktaba, directora del proyecto MoProSoft, y el Dr. Oscar Pastor, investigador de desarrollo WEB de la Universidad Politécnica de Valencia (UPV), y dos líderes de equipos de desarrollo WEB la Ing. Liliana León Azpiroz y la Ing. Diana Campos Chalot. Con la guía de los expertos académicos se desarrolló llevó a a cabo una investigación bibliográfica para ubicar los proceso y técnicas del estado del arte, probadas y propuestas, para desarrollo de sistemas WEB. Atraves de la consulta con los desarrolladores se seleccionaron de las técnicas y metodologías de desarrollo WEB, investigadas, las más apegadas a MoProSoft y los equipos pequeños de desarrollo.

Las actividades a realizadas y sus objetivos fueron:

Investigación bibliográfica para:

- Identificar las características y fases del proceso de desarrollo WEB.
- Conocer las técnicas del estado del arte, probadas y propuestas, para el desarrollo de sistemas WEB.
- Encontrar ejemplos de productos de trabajo de sistemas WEB, que cumplan los objetivos de los productos del proceso de desarrollo y mantenimiento de software de MoProSoft.

Consulta con líderes de equipos desarrollo para seleccionar las técnicas cuya aplicación fuera posible a los equipos pequeños de desarrollo.

Consulta y trabajo con expertos para mantener la investigación y construcción de la guía bien enfocados y generar resultados relevantes y de calidad.

Redacción de la guía formada por:

- Una integración, coherente del proceso de desarrollo y mantenimiento de software de MoProSoft y algunas técnicas de desarrollo WEB.
- Selección de ejemplos de productos de trabajo de sistemas WEB para cada producto del proceso de “Desarrollo y Mantenimiento de Software” de MoProSoft.
- Estancia de investigación con el Dr. Oscar Pastor en la UPV para guiar la investigación bibliográfica y validar la guía en su conjunto.

1.5 Estructura del trabajo:

Capítulo 2 Marco teórico: Explica la estructura y componentes de Internet y los sistemas WEB. Describe las características de los sistemas WEB y las dificultades que estas causan en su desarrollo.

Capítulo 3 Proceso de desarrollo WEB: Describe las similitudes y diferencias entre los diferentes tipos de procesos de desarrollo de sistemas y su convergencia con MoProSoft y lo planteado en la Guía.

Capítulo 4 MoProSoft: Explica que es MoProSoft y su relación con la guía.

Capítulo 5 Construcción de la guía: Describe como se construyó la guía. Se explica su estructura y el porqué de la misma se señalan las similitudes y diferencias de notación con MoProSoft y el proceso de Validación que hubo de la misma.

Capítulo 6 Conclusiones: Se describe lo que el desarrollo de la presente tesis implicó y enseñó al autor.

Capítulo 7 Trabajo a futuro: Describe que líneas de trabajo se pueden desarrollar a partir de esta tesis.

Anexo A “Guía para el desarrollo de aplicaciones WEB. Basada en el modelo de procesos para la industria de software MoProSoft.” : Contiene la guía producto de esta tesis.

2 Marco teórico

2.1 Componentes y estructura de los sistemas WEB

Los sistemas WEB pueden definirse como todos los sistemas de software cuya operación depende de Internet. Sin embargo en el presente trabajo de tesis se usa el término “sistemas WEB” para referirnos al grupo más numeroso de estos: Se trata de los sitios o portales WEB cuyas páginas son generadas dinámicamente en respuesta a las acciones y peticiones de los usuarios. Los componentes y el funcionamiento de estos sistemas influyen en sus propias características y las de su desarrollo.

Internet en términos materiales, es un conjunto de computadoras y dispositivos electrónicos muy diversos. Estas computadoras y dispositivos están interconectados de forma tal que es posible enviar y recibir información desde una computadora a cualquier otra dentro de la red. Los componentes de Internet son muy diversos en origen, forma, funcionamiento, diseño, etc. Las referencias [14],[24],[25] y [26] muestran que Las claves para que funcionen en conjunto son el uso de lenguajes, formatos y protocolos estandarizados, como [23], [22] especificados por la W3C (World Wide Web Consortium)¹[14] , y la organización de todas las acciones en sub-tareas que son resueltas por diversos sistemas y procesos mucho más simples que Internet en sí. Algunas de estas normas, estándares y protocolos son: Ethernet, WiFi, T1, etc. que especifican cómo se transmite la información por cables, ondas de radio, troncales telefónicas. Protocolos como el IP, DNS, TCP e UDP establecen cómo determinar rutas de envío y direcciones. Finalmente formatos de archivos como jpg, mov y mp3 y lenguajes como HTML, XML, JavaScript y CSS se usan para codificar y decodificar la información que perciben los usuarios a través de las computadoras.

Mientras los dispositivos y sistemas sigan los protocolos, el proceso que usen internamente para lograrlo es irrelevante para el resto de los componentes. Esto hace posible la interacción de sistemas muy diversos y la fácil sustitución de cualquiera de ellos.

Específicamente para el caso de los sistemas de software en Internet se necesita que dos programas intercambien información usando los mismos formatos, lenguajes y protocolos. La mayoría de los sistemas opera bajo el esquema de Cliente/Servidor. Esto implica que existe un programa “servidor” que recibe y atiende peticiones de uno o varios programas clientes. El intercambio de información en estos esquemas es de uno a muchos. La otra posibilidad son los sistemas uno a uno (peer to peer), donde no existe un servidor y los programas que

¹ Principal organización internacional de estándares para Internet.

intercambian mensajes o información son equivalentes en sus funciones. La figura 2-1 muestra el flujo de información en ambos esquemas.

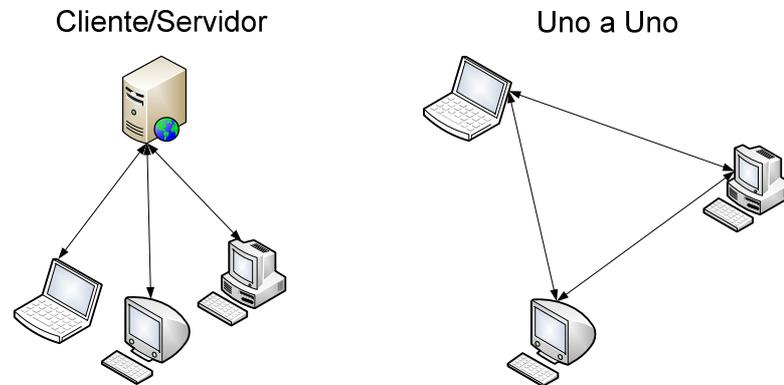


Figura 2-1: Esquemas de funcionamiento de las aplicaciones en Internet

Los sistemas WEB (objeto de estudio de esta tesis), así como los de correo electrónico, Google Earth y muchos otros. Funcionan bajo el esquema Cliente/Servidor. Sistemas como los programas de mensajería instantánea (chat) y los sistemas para compartir archivos (bit torrent), operan bajo el esquema uno a uno.

Los sistemas WEB utilizan los lenguajes HTML, XHTML, XML, XSL y CSS para contener la información en texto y describir su organización en la pantalla. Adicionalmente el uso de diversos formatos de archivos como jpg, gif, swf, mov, wav, midi, mp3, etc. permite integrar imágenes, sonido y video dentro del texto representado.

HTML, XHTML, XML no son lenguajes de programación. Son lenguajes que describen la información. Están basados 100% en texto y dentro utilizan “etiquetas” o señalamientos para que las aplicaciones distingan entre el texto y la información que “califica” al texto. Las etiquetas se presentan en pares que rodean al texto que califican. Se distinguen por estar enmarcadas por los símbolos < >. Existen etiquetas que abren y etiquetas que cierran, a estas últimas se les agrega una diagonal / después del símbolo inicial. Así los navegadores pueden distinguir que en una cadena como esta: <p> texto cualquiera </p>, el texto así delimitado constituye un párrafo. O que en la cadena: <libro> <titulo>La fundación</titulo> <autor> Isaac Asimov</autor> </libro> indica que hay un libro de titulo *La fundación* cuyo autor es Isaac Asimov.

Java Script es un lenguaje de programación que permite a los programadores hacer que los navegadores ejecuten ciertas tareas de procesamiento de información. Con JavaScript se puede, por ejemplo, hacer que el navegador no envíe un formulario hasta que el usuario registre todos los datos que se le solicitan. A diferencia del HTML y sus similares, no contiene datos contiene instrucciones.

El término *servidor* se utiliza simultáneamente para referirse a las computadoras que desempeñan esta función como a los programas que éstas ejecutan. Estos sistemas también operan bajo la premisa de ser independientes entre sí. Los programas servidores de páginas pueden operar en una gran variedad de computadoras. A su vez, las computadoras servidor pueden ejecutar una gran variedad de programas.

Los programas clientes para los sitios o portales WEB son mejor conocidos como “navegadores” de Internet y tienen las instrucciones para acceder a los servidores y enviarles las solicitudes de los usuarios. Posteriormente, reciben e interpretan lo que los servidores les responden y despliegan la información en las computadoras de los usuarios. Hay navegadores para una gran cantidad de computadoras desde las más tradicionales como las PC y las Macs hasta las consolas de video juegos y los teléfonos celulares.

Los sistemas WEB son los que operan dentro de un servidor y contienen diversos archivos y programas que los servidores ejecutan para atender las peticiones que los usuarios hacen a través de los navegadores. Los programas del sistema WEB almacenan, procesan y leen información para generar una respuesta en código HTML, XML, JavaScript, CSS, XSL y en algunos casos, otros formatos que son los lenguajes y datos que pueden interpretar los navegadores. La respuesta que reciben los navegadores siempre debe estar en estos lenguajes. Si es así, el proceso mediante el cual se generaron los contenidos resulta irrelevante para el navegador. Los sistemas WEB pueden estar programados en una gran variedad de lenguajes como son PHP, JSP, ASP, C, Perl, etc. Todos estos lenguajes se usan para generar HTML, XHTML, XML, XSL y CSS y los otros formatos que entiende un navegador por lo tanto el lenguaje de programación específico no afecta forma alguna a los navegadores.

La misma propiedad pero en sentido inverso permite que mientras los navegadores puedan hacer peticiones e interpretar HTML, XHTML, XML, XSL y CSS para el servidor sea completamente irrelevante cómo o con qué lenguaje se programaron los clientes. Sin embargo los navegadores son la interfaz entre la información en lenguajes de marcado y la información que recibimos los usuarios humanos. Esto ocasiona que se deban considerar las diferencias con que los distintos navegadores interpretan un mismo código HTML, XHTML, XML, XSL, CSS o JavaScript para evitar que se generen resultados diferentes a los deseados en algún navegador.

El diagrama de la figura 2-2 muestra la estructura de los diferentes componentes involucrados en la operación de los sistemas WEB y el flujo de la información. Mientras se sigan protocolos y estándares, en teoría cualquiera de los componentes puede ser modificado o sustituidos sin alterar en forma alguna a los otros y el funcionamiento del conjunto.

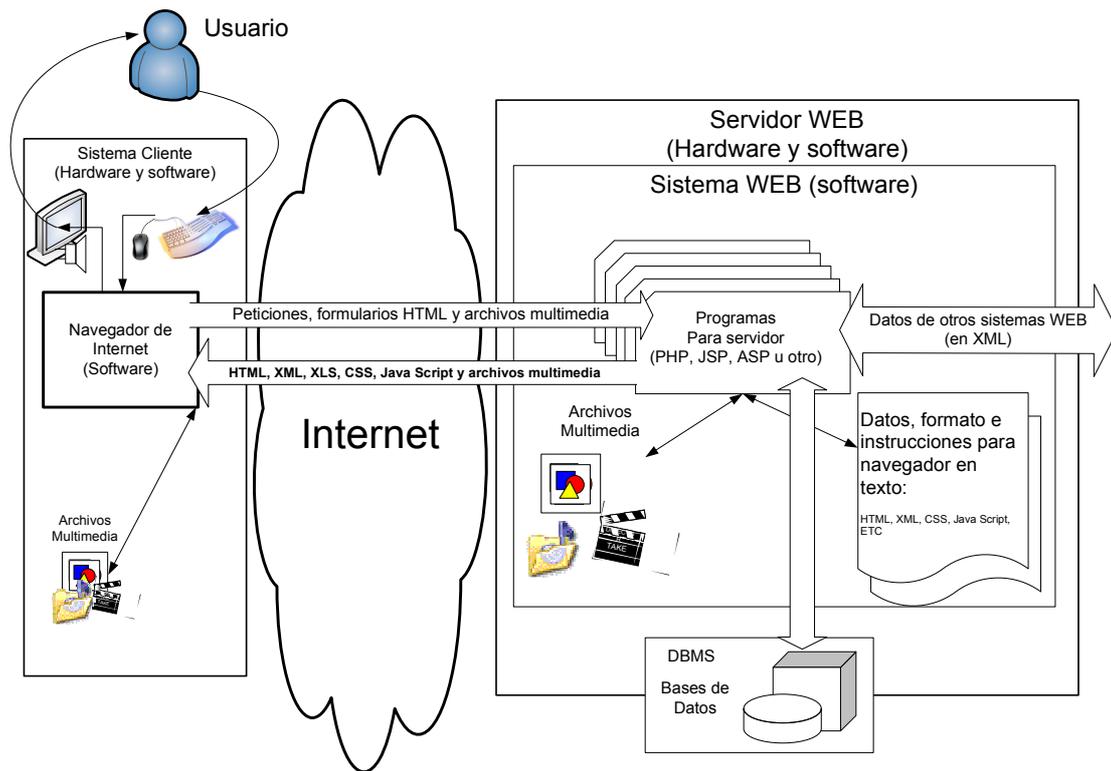


Figura 2-2: Estructura de los diferentes componentes involucrados en la operación de los sistemas WEB

Esta estructura también permite que la información que utiliza un sistema WEB provenga, no solo del servidor y los usuarios sino además de otros sistemas WEB. Gracias a esto están surgiendo sistemas WEB especializados que permiten agregar, de forma relativamente simple, información valiosa y compleja a los sistemas que se estén desarrollando.

2.2 Características de los sistemas WEB

Cada sistema es único sin embargo hay características que se repiten en casi todos los sistemas WEB. Estas características están directamente relacionadas con la naturaleza de la propia Internet, con las funciones y usos de los sistemas así como con las dificultades en el desarrollo de los mismos.

C1 Comunicación

La función intrínseca de todos los componentes de Internet es transmitir y recibir información, o en otros términos: comunicar. “La comunicación es obviamente el factor subyacente del éxito de la WEB a la vez que la mayor fuente de complejidad y retos para el desarrollo WEB” [2, pag:38]. Los sistemas de software tradicionales almacenan, procesan y organizan la información, pero en comparación con los sistemas WEB, sus posibilidades de transmitirla son limitadas. Los sistemas WEB pueden o no cumplir alguna o todas las funciones “tradicionales”, pero inevitablemente deben transmitir información. En este contexto la generación, actualización, confiabilidad y adecuación de la información de un sitio o sistema WEB es un factor de singular importancia.

C2 Interacción

Un factor adicional, que da aún más versatilidad y complejidad a la WEB, es la bidireccionalidad del flujo de la información. Los usuarios no solo reciben información, también pueden enviarla. Esto permite que los usuarios puedan asumir una amplia gama de roles al usar un sistema. Pueden ser desde simples lectores, como en el caso de un portal de noticias, hasta los responsables casi absolutos de los contenidos de los sistemas, como ocurre en Wikipedia. La tendencia general es involucrar a los usuarios en algún grado en todos los sistemas. Los sitios de noticias permiten a los lectores hacer comentarios sobre las mismas, los de ventas permiten evaluar los productos, etc, etc.

C3 Estética y diseño gráfico preponderantes

El esquema Cliente/Servidor implica que el cambio de un sistema a otro es simple e inmediato para los usuarios. Internet da a los sistemas instantáneamente millones de usuarios en potencia, pero a la vez miles de sistemas competidores. La utilidad, la facilidad de uso, el precio, la accesibilidad, la oportunidad y el atractivo visual son los factores más significativos para retener usuarios. Ante la gran oferta, los usuarios hacen juicios velozmente basándose a veces en un solo vistazo. Esto hace que la estética y el diseño gráfico sean más importantes que en los sistemas tradicionales, pues resultan claves para retener al usuario lo suficiente para que decida evaluar las otras características.

C4 Intuitivos y autoexplicativos

Los mismos factores que causan C3 también fuerzan que los sitios sean fáciles de usar e intuitivos. Los usuarios que decidan usar el sitio, lo abandonarán si no logran lo que pretenden fácilmente.

C5 Vinculados

La abundancia de sitios y la facilidad de cambiar de uno a otro no representa únicamente retos, también es una oportunidad. Los sistemas WEB pueden valerse de otros para atraer visitantes y cubrir funciones y contenidos si gastar en implementarlos o mantenerlos.

C6 Cambio continuo

La tecnología de Internet permite el cambio continuo y además el mercado lo exige. La información es un bien perecedero, y el cambio continuo una herramienta para atraer a más usuarios. Estos factores refuerzan la necesidad de cambios muy frecuentes a todos los niveles en la mayoría de los sistemas WEB.

C7 Tecnología diversa

La estandarización de los protocolos y la tecnología Cliente/Servidor permite la integración de una red muy heterogénea. El hecho de que la interoperabilidad sea posible fomenta la diversidad de los sistemas. La variedad de sistemas existe tanto en características relacionadas al hardware como al software. En cuanto a hardware existen una gran variedad de tipos de computadoras, tamaños y resoluciones de la pantalla así como diversos dispositivos de entrada como ratones, pantallas táctiles, teclados, lápices apuntadores, etc. En el terreno del software varía el navegador, su versión, y los plugins (complementos) que tenga para interpretar formatos adicionales de archivos.

Los estándares permiten que sistemas muy diversos puedan inter operar y ser reemplazados sin alterar a los sistemas restantes pero no cambia las diferencias entre un sistema y otro. Así aunque un celular un una PC puedan acceder a un sistemas WEB específico el celular nunca podrá presentar en su pequeña la información del mismo modo en que se esta se presenta en la pantalla de PC. En el caso del software es posible que en dos sistemas idénticos en hardware y con el mismo navegador uno pueda desplegar animaciones multimedia y otro no. Esto puede ocurrir si el plugin para tal fin se instaló sólo en un navegador.

C8 Usuarios simultáneos y diversos

La diversidad de los usuarios es igual o más compleja que la de la tecnología. Hay que considerar factores variables en cada individuo como el idioma, la familiaridad con la información del sistema, el tiempo disponible, etc. También es necesario evaluar si la

información es sensible al acceso simultáneo. En sitios dependientes de operaciones con fechas y horas se debe atender las confusiones y problemas que puede causar el acceso desde lugares con otro uso horario. La meta generalmente es satisfacer lo mejor posible a un amplio número de miembros de un segmento de mercado específico, pues satisfacer plenamente a todos es imposible.

C9 Múltiples involucrados con múltiples áreas de especialización

La transferencia de información es crítica en Internet y no es un elemento puramente tecnológico. Intervienen muchos otros factores relativos a la experiencia y cognición humana. Este fenómeno ocasiona la necesidad de roles e involucrados en los sistemas WEB que son inexistentes o muy poco frecuentes en los sistemas tradicionales. Debe haber quienes generen la información, quienes diseñen como organizarla para hacerla accesible, quienes la actualicen, quienes la validen y evidentemente quienes la consulten. Para cumplir estas tareas suelen ser necesarias personas de diversas especialidades.

2.3 Dificultades en los desarrollos de sistemas WEB

Las características comunes de los sistemas WEB hacen que a su vez se presenten con frecuencia diversas dificultades en casi todos los desarrollos de sistemas WEB. Las más significativas de estas son:

D1 Necesidades de usuarios frecuentemente relegadas

Los criterios para diseñar los sistemas suelen considerar a las preferencias, necesidades y capacidades de los usuarios. El uso a distancia, que permite Internet, implica que no sea fácil o viable un contacto presencial con ellos. Por esta razón es común que los usuarios estén físicamente ausentes durante el desarrollo de los sistemas WEB. Esta ausencia presencial ocasiona muchas veces que se omitan algunos de los aspectos importantes. Por la misma razón muchos otros aspectos se diseñen con base en suposiciones, interpretaciones y preferencias, de otros involucrados, como el cliente patrocinador, con mayor injerencia directa en el proyecto. Generalmente dichos involucrados reflejan sus gustos personales y no las necesidades de quienes serán los verdaderos usuarios.

D2 Requisitos altamente cambiantes

Aunado al factor de cambio constante de Internet, los propios requisitos de los sistemas suelen cambiar antes de que estos estén terminados. El iniciar con objetivos muy generales, las nuevas ideas que surgen a partir del aprendizaje que causa la interacción de los diversos involucrados y el mismo número de involucrados, son solo algunos ejemplos de posibles causas. En cualquier caso el cambio de requisitos es muy frecuente y sus efectos son muy importantes en los desarrollos de sistemas WEB.

D3 Dificultades de comunicación

Las comunidades generan sus propios sub-lenguajes. Dan usos particulares a expresiones generales, tienen expresiones propias y utilizan términos que desconocen otros grupos de personas. En los desarrollos WEB, aun en el caso de sistemas pequeños, intervienen personas de diversas disciplinas. Esto ocasiona dificultades de comunicación que suelen ser muy costosas.

D4 Tecnología diversa y cambiante

La arquitectura de Internet y el uso de estándares permiten la heterogeneidad pero no eliminan todos los problemas que ésta causa. Estos incluyen desde diferencias en características físicas (como el tamaño de la pantalla) hasta pequeñas sutilezas en la forma y detalle con que cada navegador interpreta un estándar. Aun sobre las bases de los estándares más generales como HTML, los navegadores tienen algunas diferencias que pueden causar serias dificultades

en algunos sistemas. Además hay un gran número de tecnologías complementarias que permiten desarrollar funciones complejas a bajo costo, pero que no están ampliamente difundidas en los sistemas clientes. La oferta de dispositivos, protocolos, formatos, navegadores y servidores cambian constantemente. En ocasiones el sistema que estamos desarrollando o manteniendo se puede ver afectado por un cambio en un componente que no tiene efecto en el 99.9% de los sistemas. La dificultad para programar ciertas funciones o acceder a otros sistemas de apoyo como bases de datos, varía entre los diversos lenguajes con que se puede programar el sistema en el servidor. Nuevas tecnologías pueden ser muy sencillas de implementar pero no ser utilizadas por la mayoría de los clientes objetivo. La diversidad de navegadores y servidores puede volver compleja y crítica la selección de los lenguajes y tecnologías utilizados.

3 Proceso de desarrollo WEB

A pesar de todas sus particularidades los sistemas WEB son sistemas de software. Además el desarrollo de sistemas WEB es una disciplina relativamente nueva. La mayoría de quienes desarrollan sistemas WEB actualmente, han sido formados como desarrolladores de software “tradicional”. Por estas razones los trabajos que indagan sobre las mejores prácticas para desarrollos WEB suelen ser evaluaciones del desempeño de los procesos tradicionales [3,pags:197-218] en un proyecto WEB o propuestas puntuales para resolver alguna de las dificultades particulares de los sistemas WEB[5][7].

En el capítulo 3 de [3] se hace una comparación detallada entre el desempeño de eXtreme Programing (como ejemplo de un método ágil) y RUP (como ejemplo de un método formal) para desarrollar sistemas WEB. Según los autores, la cercanía con el cliente que los métodos ágiles proponen es muy efectiva en sistemas WEB pequeños. También afirman que dada la complejidad de sistemas más grandes en estos resulta mejor usar metodologías más robustas como RUP. Finalmente, sugieren que lo mejor es un proceso adaptativo que según la complejidad del sistema, combine elementos de ambos tipos de prácticas. En conclusión especifican que mientras más complejo sea un sistema WEB más formal debe ser el proceso utilizado, como se muestra en la figura 2-3. La línea punteada expresa la relación directa y lineal que proponen entre complejidad de la aplicación y formalidad del proceso. Mientras que propone un metaproceso que desencadene la aplicación de una u otra metodología o una fase de transición entre ambas conforme cambia la complejidad del sistema. Estas conclusiones no son muy diferentes a aquellas referentes a los sistemas tradicionales y es precisamente esto lo que permite la convergencia de MoProSoft y el desarrollo de sistemas WEB.

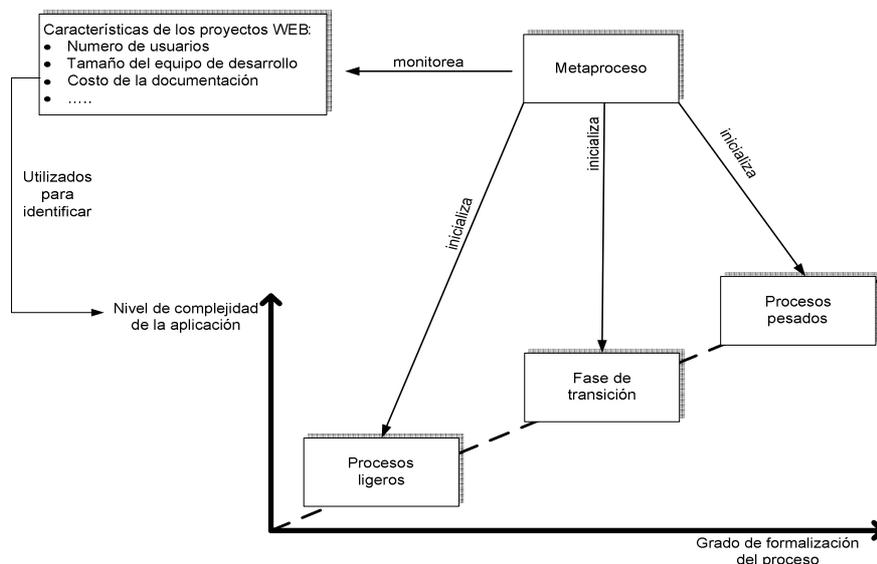


Figura 2-3: Metaproceso (adaptación de la figura 10-8 de [2])

Es importante notar que la palabra utilizada para distinguir los procesos “lightweight” (eXtrem Programing) o de bajo peso y los procesos “heavyweight” (RUP) o pesados es formalización. Por formalización se refieren a la cantidad y formato de elementos documentados así como a la rigidez de los procesos y procedimientos para la interacción entre los involucrados. Los procesos tradicionales o pesados (heavyweight) requieren documentación generalmente en forma de documentos de texto con diagramas. Suele especificar una serie de documentos y requisitos de contenidos, formato y validación para cada uno. Una gran parte de la documentación de un proceso “heavyweight” perdura después del fin del proyecto para el que fue creada. Estos procesos pesados promueven incluso que se guarden las diversas versiones de cada documento. Los procesos ligeros establecen formas de trabajo basadas en comunicación personal constante documentan estrictamente lo necesario. El medio para documentar incluye medio tradicionales como documentos pero también otros como pizarrones y notas adheribles. Gran parte de la información no se almacena más allá del proyecto y cambia conforme este evoluciona, sin que se respalden las versiones anteriores.

En esencia, casi todos los procesos modernos son similares. Tienen una fase de *arranque* en la que se establece formalmente el proyecto y el equipo de desarrollo. Una fase de *requerimientos* en la que se determina qué debe hacer el sistema. Una fase de *diseño* en la que se determina como se construirá el sistema. Una fase de *construcción* en la que se hace el sistema y en otra de pruebas en la que se constata que sistema cumpla sus objetivos y funcione correctamente. Estas fases fueron determinadas desde del hoy obsoleto proceso en cascada. Este establecía que estas fases se sucedían una a la otra secuencialmente. A diferencia de este los procesos modernos proponen una ejecución constante y/o simultanea de diversas fases a lo largo de todo el proyecto. También incluyen la posibilidad de trabajar en ciclos.

Es en el establecimiento de estas fases en las que hay convergencia entre MoProSoft y lo planteado por los métodos ágiles o lightweight, los formales y heavyweight y sus posibles adopciones para desarrollo de sistemas WEB. La guía se apaga a MoProSoft que establece en forma análoga a un método formal una serie de productos que se deben generar a lo largo del proceso de desarrollos de software. El objetivo de todos estos productos, exceptuando el sistema en si es documentar. Sin embargo MoProSoft no establece un formato o medio para dichos productos. Esto permite que MoProSoft se adapte a la propuesta del capítulo 3 de [3] que plantea una relación directamente proporcional entre la formalidad del proceso y la complejidad de los sistemas WEB a desarrollar. La guía es consistente con esta misma propuesta y proporciona descripciones de los productos y diversos ejemplos pero no formato específico para los mismos.

3.1 Técnicas para desarrollo WEB

Si bien el proceso general permanece prácticamente inalterado en cuanto a sus fases requerimientos, diseño, construcción y pruebas. En el área de técnicas y/o metodologías específicas para alguna de las etapas encontramos varias propuestas.

T1 Análisis de requerimientos enfocado a objetivos

El desarrollo WEB involucra información, navegabilidad, estética y personas de diversas disciplinas además de la funcionalidad fundamental en todo sistema de software. En [7] se propone el levantamiento de requerimientos enfocado a objetivos. Consiste en términos generales en listar a los diversos involucrados y sus objetivos. Posteriormente descomponer dichos objetivos en sub-objetivos que si son complementarios se marcan con –y- (AND) y si son alternativos se marcan con – ó - (OR). Después se marcan las relaciones entre los diversos árboles y/o ramas con líneas punteadas. Los objetivos contradictorios se marcan con una línea cruzada con una X. “Las raíces de los árboles son los objetivos mientras que las hojas son los requerimientos” [7, pag:6]. Las hojas o requerimientos se pueden catalogar después como:

- Content (labeled with C); -Contenido-
- Structure of Content (SC); -Estructura del contenido-
- Access Paths to Content (A); -Ruta a contenido-
- Navigation (N); -Navegación-
- Presentation (P); -Presentación-
- User Operation (U); -Operación del usuario-
- System Operation (O); -Operación del sistema-
- Interaction (I). -Interacción-

El árbol resultante del ejemplo utilizado en [7] se muestra en la figura 2-4.

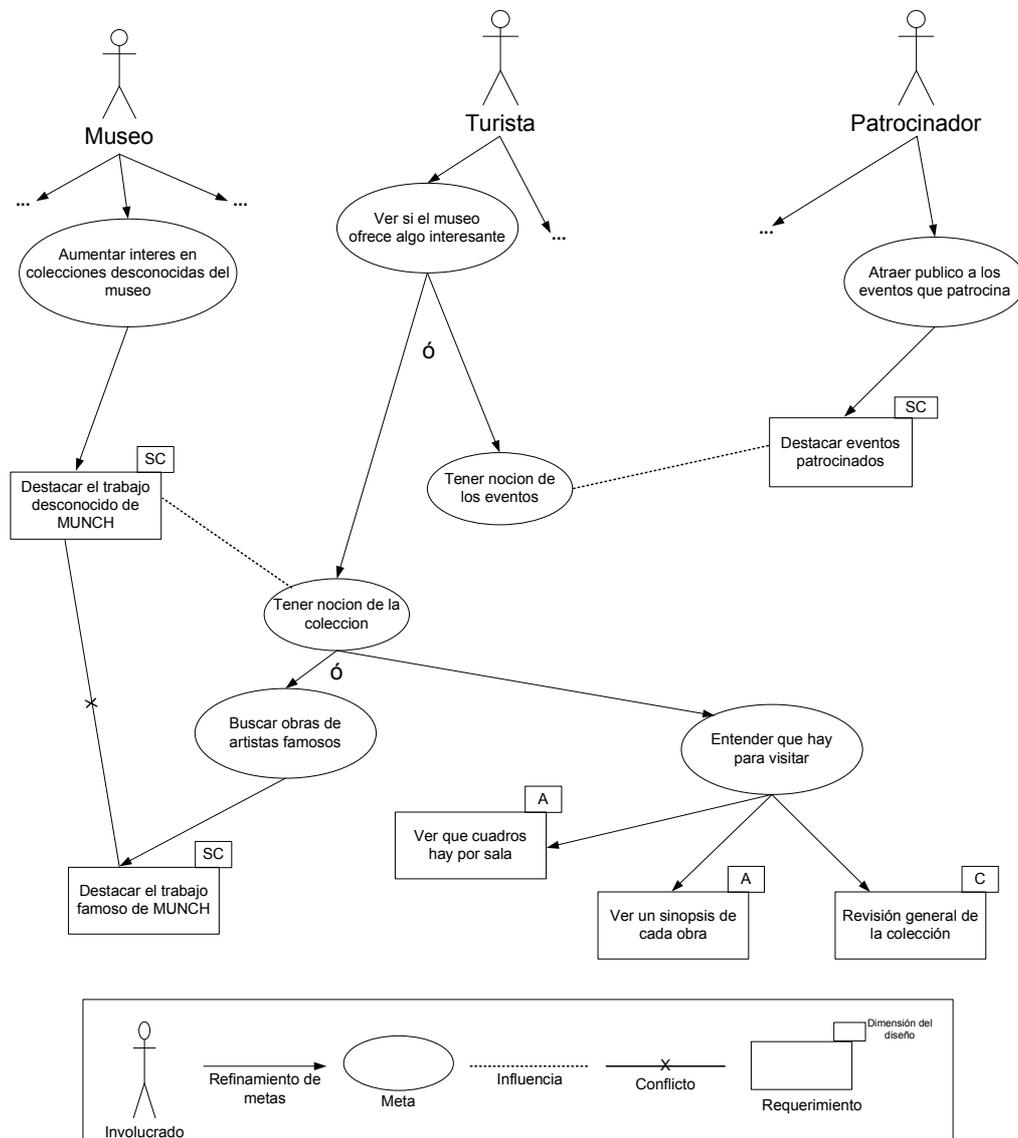


Figura 2-4: Ejemplo de árbol de requerimientos por objetivos, traducción de árbol en [7].

En este ejemplo se muestra que entre otros objetivos el museo desea aumentar el interés en colecciones poco conocidas de su acervo. El turista desea ver que hay para ver de su interés en el museo y los patrocinadores desean atraer público a los eventos que patrocinan. Después e muestra la descomposición de estos objetivos hasta llegar a los requerimientos puntuales como Ver que cuadros hay por sala, hacer una revisión general de la colección, destacar el trabajo de E. Munch.

Los requerimientos resultantes se pueden listar y priorizar para resolver los que sean posible de acuerdo al alcance del proyecto. Los conflictos se resuelven haciendo propuestas y discutiendo con los involucrados.

Este procedimiento es inicialmente poco específico. Esto es una ventaja, en los desarrollos WEB, pues no obliga a hacer decisiones de diseño cuando aún no se tienen todos los elementos necesarios. Por su propia definición y proceso, el levantamiento de requerimientos enfocado a objetivos también obliga a considerar a todos los involucrados en el sistema, por ejemplo: El que encarga el sistema, quienes lo usaran tanto del grupo (u organización) de clientes como fuera de este, quienes lo administraran, quienes generar la información inicial y quienes al actualizaran. Adicionalmente una vez terminado este tipo de análisis proporciona datos tanto de los requerimientos funcionales como de la información e inclusive varios requerimientos no funcionales. Es importante tener en cuenta que esta técnica permite resolver de mejor manera las dificultades que causan usuarios con objetivos difusos o muy generales. Millones de cibernautas acceden a sitios sin más objetivo que ver que encuentran. En los sistemas tradicionales este tipo de usuario es muy improbable pero en los sistemas WEB es más bien la norma.

T2 Prototipos Grises

La comunicación no es solo una de las principales funciones de un sistema WEB, es también uno de los principales problemas durante el su desarrollo. En [6] Eric Holter describe la metodología de trabajo usada en su empresa Newfangled WEB Factory. Bajo la premisa de evitar la ilusión de comunicación se plantea un esquema de trabajo cuyo eje conductor es la elaboración de prototipos denominados grises.

Los prototipos grises son páginas con ligas funcionales y contenidos pero que se enfocan únicamente en la información que cada página o sección debe contener. El calificativo gris, color que la mayoría de las personas no encuentra atractivo, es intencional y se da por la ausencia de todo diseño gráfico. Estos prototipos permiten que clientes y desarrolladores se enfoquen en los contenidos sin distractores. El trabajo y los avances se hacen tangibles y evaluables inmediatamente en los prototipos y evitan ambigüedad y la ilusión de comunicación que pueden dar los términos y descripciones abstractas utilizados cuando no se hacen prototipos. Son también instrumentos fáciles de hacer y que sirven de base para el producto final.

T3 Prototipos operativos

Los e-prototipos o prototipos operativos son un nuevo paradigma que aborda dos de los principales problemas de los desarrollos WEB, el contacto con el cliente y el cambio constante. Son descritos en [5] y básicamente consisten en elaborar prototipos funcionales que se pongan en operación para que los usuarios reales puedan probarlos y dar sus opiniones. En el software

tradicional también se llegan a usar algunas versiones beta que son distribuidas a usuarios selectos, con este fin. Pero la naturaleza técnica de estos sistemas hace difícil la implementación de este esquema como principal estrategia de desarrollo. En la WEB por el contrario es fácil y natural; se puede consultar desde la misma aplicación con los que serán los usuarios reales y los cambios se pueden implementar fácil y constantemente. La bidireccionalidad propia del medio también hace extremadamente sencilla la retroalimentación. Los prototipos generalmente pueden ser reaprovechados parcial o totalmente. Los usuarios no sugieren o expresan necesidades supuestas sino reales, fruto de su propia experiencia con el producto.

El desarrollo por e-prototipos se basa en un paradigma de evolución constante del software en ciclos cortos. Esto se ajusta a las demandas del mercado WEB de cambio constante. Además establece comunicación directa con los usuarios, que generalmente no se pueden consultar en otros modelos de desarrollo WEB. Los prototipos operativos son ya ampliamente utilizados por diversas compañías de gran éxito en Internet. Los que podemos identificar son aquellos sistemas con la leyenda BETA como en su momento fueron por ejemplo G-Mail o Yahoo-Groups. Pero puede haber muchos otros sistemas que omitan este calificativo y sin embargo operen bajo el esquema de recopilación de requerimientos directos de los usuarios finales por medio de correos, foros, etc.

T4 Técnicas para modelación

Aun no existe un estándar para modelar los sistemas WEB. Existen muchas propuestas como son OOWS[2], WebML, UWE, entre otros. Estos lenguajes no son idénticos pero en general todos siguen la misma estrategia y paradigmas. Toman como base UML y lo extienden para poder modelar también los contenidos y la navegación.

El uso de diagramas se ha vuelto indispensable al documentar los sistemas. La versatilidad, expresividad y simplicificación que implican, ha llevado incluso al planteamiento de una nueva forma de desarrollo de sistemas conocida como MDA o Model Driven Architecture por sus siglas en ingles. En el CDROM anexo que contiene una versión de la guía en HTML se pueden encontrar un sitio modelado en OOWS como un ejemplo anexo al producto *P2 Bocetos, Listas de Sitios Similares y Prototipos* puedes encontrar diagramas OOWS una herramienta de MDA con la que se modelo casi en su totalidad (excluyendo parte del el diseño grafico que se especifica en CSS) el sitio: <http://arqsostenible.upv.es/>. OOWS es uno de los nuevos metalenguajes de desarrollo basado en modelos aunque aun es muy limitado y está en etapa experimental pero que ya permite la generación de sitios a partir únicamente de modelos como se muestra en [2]. El desarrollo WEB 100% basado en modelos es aún muy limitado y carece de herramientas profesionales y estables. Sin embargo es un área con mucho potencial y es conveniente mantenerse atento a los desarrollos de la misma.

T5 Uso de patrones

En la sección de arquitecturas de [1] se puede ver que en los sistemas WEB se pueden usar la mayoría de los patrones del diseño de software tradicionales. Sin embargo si se pone atención a la búsqueda de patrones al navegar la propia Internet y se consultan sitios dedicados al tema como [19] y [20] veremos que el uso de patrones no se limita a éstos. Los usuarios de sistemas WEB están poco dispuestos a leer indicaciones y manuales. Por esto los sitios deben ser intuitivos y fáciles de usar para sus usuarios objetivo. Los patrones para sitios WEB (son diferentes de los diseño) son una gran herramienta para lograr la simplicidad y familiaridad de los usuarios con el sistema. El uso de elementos comunes en muchos sitios facilita a los usuarios el uso de los mismos. Así podemos encontrar patrones casi universales en los sitios WEB como son la barra de navegación, las tablas o los contenidos indexados. Hay también patrones para sistemas o componentes con funciones específicas, por ejemplo los buscadores básicamente constan de un campo de texto y un botón para iniciar la búsqueda; en el caso de los sistemas que venden productos utilizan alguna variante de los “carritos de compras”. Otro caso son patrones para mostrar cierto tipo de información como las ligas o secciones tituladas “Inicio”, “Acerca de” o “Contacto”. Otros patrones se encargan de la presentación de la información así podemos encontrar contenidos indexados, tablas, pestañas, paginación, mosaicos de imágenes, etc. La taxonomía y clasificación de los diversos patrones WEB aun no está unificada sin embargo esto no impide su uso. La observación de los sitios WEB similares resulta crítica en este sentido.

T6 Técnicas para el diseño de arquitecturas

En diversas aéreas como el diseño grafico y funcional las variaciones entre los patrones utilizados en los sistemas WEB y los tradicionales son notorias. Sin embargo en el área correspondiente a la arquitectura y la programación en sí, estas diferencias son más sutiles y en general en los sistemas WEB se utiliza un subconjunto de los muchos patrones de arquitectura que se pueden utilizar en los sistemas tradicionales. Esto se debe a que las arquitecturas WEB deben poder funcionar en el esquema cliente/servidor y es muy deseable que permitan separar las funciones de la información.

En la sección de arquitecturas de [3] se describen muchas posibles arquitecturas de los sistemas WEB como modelo de N capas, JSP Modelo-2, Struts o OOHDM-Java2 en su mayoría son adaptaciones de la arquitectura de los sistemas de software tradicional. Las bases de la mayoría son la arquitectura por capas y la de modelo, vista, controlador. La primera se plantea generalmente con una capa de interfaz con la que interactúa el usuario, una o varias capas de procesamiento de información una última capa persistente donde se almacenan los datos. Los datos y comandos en este modelo son propagados capa por capa y cada capa puede interactuar directamente sólo con las capas superior y/o inferior. La segunda arquitectura presenta un modulo de vista, uno de procesamiento de datos o control y uno de almacenamiento persistente. En este caso los tres módulos pueden interactuar indistintamente con los otros dos. Estas arquitecturas se utilizan porque tienden a aislar y facilitar el manejo de la información que como

se mencionó en las características *C1 Comunicación*, *C2 Interacción* son preponderantes en los sistemas WEB.

Aun cuando los paradigmas y estructura básica de estas arquitecturas no varían es importante tener en cuenta la influencia de la información en los sistemas WEB. Las arquitecturas de N capas y las basadas en modelo-vista-controlador permiten un manejo separado de los datos y la información del sistema y en este punto convergen también con las nuevas tecnologías de lenguajes de marcado como XML, por esto son de las más utilizadas en los sistemas WEB.

T7 Técnicas de Diseño gráfico

Como se explica en *C3 Estética y diseño gráfico preponderantes* son muy importantes y tienen un rol fundamental en retener a los navegantes en los primeros instantes en que visitan un sistema WEB. Sin embargo, resultan inservibles si el sistema no resulta útil al usuario. El diseño gráfico debe entonces procurar una presentación clara de la información en forma estética. En sistemas WEB el mejor lenguaje para manejar estos aspectos son las hojas de estilos CSS que, con un poco de práctica, permiten el manejo y manipulación de los siguientes elementos mencionado en [18] y [3 cap:11]:

Tipografía (tipo, tamaño, color y decorado de las letras): En este caso se recomienda utilizar tipografía más grande y clara que en medios impresos, pues leer en la pantalla es incomodo. El tamaño y la decoración se deben utilizar para destacar la información más importante.

Agrupación: La información y elementos visuales deben agruparse para facilitar su entendimiento al lector. Las categorías pueden ser por función, tipo de dato, formato de presentación, etc. Los diferentes grupos pueden hacerse evidentes por proximidad de sus elementos e incluso el uso de elementos agrupadores como márgenes, líneas, recuadros e imágenes. También se puede usar el color o la tipografía.

Orden: El orden de lectura de arriba abajo y de izquierda a derecha o el que venga al caso según la lengua de los usuarios objetivo. Debe considerarse al momento de ubicar los contenidos dentro de la pantalla pues determina en buena medida en que zona de la misma pondrá su atención en primera instancia el lector.

Imagen y color: El uso de colores e imágenes es una de las formas más sencillas de hacer visualmente atractivo un sistema WEB. El poder de estas herramientas es tal que muchas veces se cae en el abuso. El primer punto es evitar que la combinación de colores dificulte la lectura el color de las letras debe tener un alto contraste con el color que se use de fondo. Si se usan imágenes como fondo es muy recomendable utilizarlas como sello de agua con los colores muy atenuados. Las imágenes deben usarse como

elementos que sean a la vez informativos y decorativos. Iconos, delimitadores de conjuntos de elementos, botones, títulos o encabezados son excelentes opciones para utilizar imágenes que aporten tanto al fácil entendimiento del sistema, como a la estética del mismo. La selección cuidadosa del color las combinaciones de colores son un tema delicado no solo pueden hacer fácil o difícil la lectura alteran drásticamente la percepción de los usuarios. Una de las mejores herramientas para hacer dicha selección es el círculo de color. El círculo de color es un acomodo de los colores en una circunferencia que utilizando simples separaciones de diversos grados como 180 o 120 permite encontrar los colores que combinan con aquel que seleccionamos. En la propia Internet se pueden encontrar gran variedad de versiones de esta herramienta y lograr fácilmente buenas combinaciones de colores.

Es importante además tener constancia y congruencia en todo el sistema. Mantener procesos, criterios de clasificación y formas de presentación similares para el mismo tipo funciones, datos y/o elementos a lo largo de todo el sistema para evitar que los usuarios deban aprender e interpretar en cada pantalla.

T8 Técnicas para el diseño de navegación

Un factor único de los sistemas WEB es la navegación, la información y su presentación no pueden ser usados ni apreciados si no se encuentran. El acceso a la información debe también diseñarse. En [2] y [3] se señalan lineamientos generales que es recomendable seguir en la mayoría de los sistemas WEB y estos son:

1. **Siempre informar al usuario donde se encuentra:** La versatilidad de ir de un sitio WEB a otro y de una sección a otra con un solo clic, puede causar rápidamente desorientación en el usuario. Ya sea por ligas poco claras o distracciones ajenas al sistema el usuario puede perder noción de lo que está viendo. Por eso casi 100% de los sitios utiliza un encabezado presente en todas sus pantallas que indica de que sitio se trata. Además es recomendable indicar exactamente en qué pantalla del sistema está el usuario mostrando un título en cada una. En sistemas medianos y grandes también se debe poner la ruta o clasificación de la pantalla dentro del sistema.
2. **Tres clics de distancia:** Los contenidos deben ser accesibles. La regla de los tres clics sugiere hacer una estructura del sitio muy horizontal en donde no se necesiten más de tres ligas o pasos para llegar a cualquier contenido. Si se deben dar más pasos dificultan que se encuentren los contenidos en primera instancia y que los usuarios logren recordar cómo encontrarlos en visitas posteriores.
3. **Ligas externos:** Un sistema no está aislado, se deben generar ligas desde y hacia otros sitios relacionados. Esto facilitará que los usuarios encuentren la información y que se aprovechen desentenderte de tareas e información útil para los objetivos del

sistema pero que están más allá de su alcance. Al apoyarse en otros sistemas los contenidos y funciones pueden mantenerse simples y actualizados facilitando entre otras cosas cumplir con la regla de los tres clics.

T9 Técnicas para presentar la información

Como se constata en [21] la información en Internet puede ser video, imagen, audio o texto pero en todos los casos estos elementos deben adaptarse al medio. La *C7 Tecnología diversa*, los *C8 Usuarios simultáneos y diversos* y limitaciones como la velocidad de transmisión o el tamaño de las pantallas, influyen sobre la efectividad de estas formas de comunicación. En todos los casos se debe buscar un equilibrio de dos factores contrapuestos al presentar información clara y completa pero a la vez breve y concisa.

En el marco del *C6 Cambio continuo* que impone Internet la información es el aspecto que más cambia de todos. La generación de información clara y completa pero a la vez breve y concisa es muy costosa y tardada por lo tanto es muy importante distribuir esta tarea entre diversos miembros de la organización cliente e involucrar a los propios usuarios de sistema, lo más posible, en las tareas de generación y actualización de la información.

En el caso de los textos deben ser breves, generalmente es recomendable elaborar resúmenes de la información que manejan diariamente los clientes o usuarios en otros medios. En los casos en que los objetivos del sistema incluyan justamente el acceso a grandes volúmenes de información lo más recomendable es presentarla en formatos descargables para su consulta fuera de línea y/o diseñados para ser impresos. Los datos en texto ofrecen una gran variedad de patrones de presentación como son la tabulación, el paginado, el indexado, el listado, etc.

En el caso de las imágenes se deben adaptar la resolución y el formato para que se vean bien en pantalla pero ocupen el menor espacio posible.

El audio y el video deben también codificarse en formatos de alta compresión y a baja resolución para que puedan ser vistos y/o escuchados aun en enlaces que presenten baja velocidad. En el caso de video es recomendable generar video donde el fondo cambie poco y/o cambie lentamente.

T10 Técnicas para probar las aplicaciones WEB

En la sección de pruebas en [3] se puede apreciar que las pruebas de las aplicaciones WEB son similares a las de aplicaciones tradicionales pero presentan dos diferencias fundamentales. Primero debido a la *C7 Tecnología diversa* son necesarias muchas pruebas para probar la misma funcionalidad en diversas plataformas. Segundo, gracias a la facilidad con que se puede actualizar el sistema y a que se cuenta con *C8 Usuarios simultáneos y diversos* estos se pueden involucrar más fácilmente en las pruebas de la aplicación, como se sugiere en [5].

Dentro del espectro de pruebas que puede desarrollar un equipo pequeño se recomienda lo siguiente:

- Probar la combinación de las tecnologías a utilizar en varios navegadores antes de incorporarlas al desarrollo.
- Seguir un procedimiento de pruebas tradicional para probar la funcionalidad de la aplicación.
- Utilizar la retroalimentación de clientes y usuarios para mejorar la detección de errores y reducir costo y tiempo de las pruebas. Un extremo de este enfoque apoyándose en los propios usuarios es la técnica de *T3 Prototipos operativos*.

4 MoProSoft

4.1 Introducción a MoProSoft

MoProSoft es el Modelo de Procesos para la Industria de Software en México. Fomenta la estandarización de la operación de las empresas desarrolladoras de Software a través de la incorporación de las mejores prácticas en gestión e ingeniería de software. La adopción del modelo permitirá elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.

La industria de software en México, está conformada principalmente de empresas pequeñas y medianas. MoProSoft selecciona, adapta e integra las mejores prácticas internacionales a fin de que el modelo sea fácil de entender, fácil de aplicar y no costoso en su adopción para las PYMES. A su vez sirve como base para alcanzar evaluaciones exitosas con otros modelos o normas, tales como ISO 9000:2000 [12] o CMM V1.1[15].

El modelo de procesos MoProSoft está dirigido a las empresas o áreas internas dedicadas al desarrollo y/o mantenimiento de software. Las organizaciones, que no cuenten con procesos establecidos, pueden usar el modelo ajustándolo de acuerdo a sus necesidades. Mientras que las organizaciones, que ya tienen procesos establecidos, pueden usarlo como punto de referencia para identificar los elementos que les hace falta cubrir.

Para la elaboración del modelo de procesos MoProSoft, fueron aplicados los siguientes criterios:

1. Generar una estructura de los procesos que esté acorde con la estructura de las organizaciones de la industria de software (Alta Dirección, Gestión y Operación).
2. Destacar el papel de la Alta Dirección en la planificación estratégica, su revisión y mejora continua como el promotor del buen funcionamiento de la organización.
3. Considerar a la Gestión como proveedor de recursos, procesos y proyectos, así como responsable de vigilar el cumplimiento de los objetivos estratégicos de la organización.
4. Considerar a la Operación como ejecutor de los proyectos de desarrollo y mantenimiento de software.
5. Integrar de manera clara y consistente los elementos indispensables para la definición de procesos y relaciones entre ellos.
6. Integrar los elementos para la administración de proyectos en un sólo proceso.
7. Integrar los elementos para la ingeniería de productos de software en un solo marco que incluya los procesos de soporte (verificación, validación, documentación y control de configuración).

8. Destacar la importancia de la gestión de recursos, en particular los que componen la base de conocimiento de la organización tales como: productos generados por proyectos, datos de los proyectos, incluyendo las mediciones, documentación de procesos y los datos recaudados a partir de su uso y lecciones aprendidas.

9. Basar el modelo de procesos en ISO9000:2000[12] y nivel 2 y 3 de CMM V.1.1.[15] Usar como marco general ISO/IEC 15504[12] - Software Process Assessment e incorporar las mejores prácticas de otros modelos de referencia tales como PMBOK [4], SWEBOK [16] y otros más especializados.

El desarrollo y mantenimiento de software se lleva a cabo a través de una serie de actividades realizadas por equipos de trabajo. La Ingeniería de Software se ha dedicado a identificar las mejores prácticas para realizar estas actividades recopilando las experiencias exitosas de la industria de software a nivel mundial. Estas prácticas se han organizado por áreas de aplicación, y se han dado a conocer como áreas clave de procesos, en caso de CMM[15], o como procesos de software en ISO/IEC 15504[12].

*extracto de MoProSoft[1 pags: 5-6]

4.2 Categorías

MoProSoft está enfocado en procesos y considera los tres niveles básicos de la estructura de una organización que son: la Alta Dirección, Gestión y Operación. El modelo pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua.

La categoría de Alta Dirección aborda las prácticas de Alta Dirección relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos

La categoría de Gerencia aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información generada por éstos y comunica los resultados a la Categoría de Alta Dirección.

La categoría de Operación aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados.

*extracto de MoProSoft[1 pag: 12]

4.3 Procesos

Gestión de Negocio

El proceso de Gestión de Negocio establece la razón de ser de la organización, sus objetivos y las condiciones para lograrlos, para lo cual es necesario considerar las necesidades de los clientes, así como evaluar los resultados para poder proponer cambios que permitan la mejora continua.

Adicionalmente habilita a la organización para responder a un ambiente de cambio y a sus miembros para trabajar en función de los objetivos establecidos.

Gestión de Procesos

La Gestión de Procesos establece los procesos de la organización, en función de los requerimientos identificados en el plan estratégico. Así como definir, planificar, e implantar las actividades de mejora en los mismos.

Gestión de Proyectos

El propósito de la Gestión de Proyectos es asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.

Gestión de Recursos

El proceso de Gestión de Recursos consigue y dota a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores. También crea y mantiene la base de conocimiento de la organización con la finalidad de apoyar el cumplimiento de los objetivos del plan estratégico de la misma.

Recursos Humanos y Ambiente de Trabajo

El propósito del proceso de Recursos Humanos y Ambiente de Trabajo es proporcionar los recursos humanos adecuados para cumplir las responsabilidades asignadas a los roles dentro de la organización, así como la evaluación del ambiente de trabajo.

Bienes, Servicios e Infraestructura

El proceso de Bienes, Servicios e Infraestructura proporciona proveedores de bienes, servicios e infraestructura para atender los requisitos de adquisición de los procesos y proyectos.

Conocimiento de la Organización

El proceso de Conocimiento de la Organización mantiene disponible y administra la base de conocimiento que contiene la información y los productos generados por la organización.

Administración de Proyectos Específicos

El proceso de la Administración de Proyectos Específicos establece y lleva a cabo sistemáticamente las actividades que permiten cumplir con los objetivos de un proyecto en tiempo y costo esperados.

Desarrollo y Mantenimiento de Software

El proceso de Desarrollo y Mantenimiento de Software realiza sistemáticamente las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requerimientos especificados.

4.4 Niveles

MoProSoft divide todas sus actividades e incluso segmentos de estas en 5 niveles de capacidades. Estos son Realizado, Gestionado, Establecido, Predecible y Optimizado. El llegar a una aplicación completa de las mejores prácticas debe ser un proceso gradual que fomente hábitos de trabajo y no actividades que se “impongan” a la fuerza, pues los cambios que implica el seguir un modelo de procesos en el funcionamiento de la empresa y los equipos son muy drásticos. Los niveles de capacidades permiten la aplicación gradual de MoProSoft, permitiendo que las personas adopten los procesos, gracias a que experimentan los cambios y virtudes de un nivel y pueden visualizar la necesidad y oportunidades del siguiente.

4.5 MoProSoft y la guía de desarrollo WEB

MoProSoft está enfocado a la industria de software en general. Las empresas y grupos especializadas en sistemas WEB son parte de esta industria y pueden adoptar directamente los procesos sugeridos por este modelo. En la industria prácticamente no hay variación en los niveles de dirección y gerencia. Es en el nivel de operación, donde se generan los sistemas, en el que se presentan algunas particularidades. Para atender estas diferencias es conveniente hacer pequeños cambios y/o señalamientos. En la presente tesis se enfoca al subconjunto de la industria dedicado a aplicaciones WEB, permite cubrir con más detalle algunos aspectos de los procesos.

La guía observa y es compatible con el planteamiento en el capítulo 10 de [3]. Que plantea diversos grados de formalidad en los procesos para desarrollo WEB conforme la complejidad de estos productos aumente. Esto es a su vez compatible con MoProSoft que no especifica la forma o formalidad de los productos sino sus contenidos, objetivos e importancia.

5 Construcción de la guía

La guía se construyó con la intención de ayudar a la adopción de MoProSoft por parte de equipos, de dos a diez integrantes, que hagan desarrollos WEB. Los ejes conceptuales para el diseño de sus contenidos son integrar a MoProSoft los cambios y recomendaciones necesarias para que esté acorde con las técnicas, probadas y propuestas, del estado del arte en desarrollo WEB que son viables en equipos pequeños y facilitar la adopción de MoProSoft.

MoProSoft es modelo de procesos integral que abarca todos los procesos de una empresa. La guía se concentra en los cambios necesarios a este modelo para el desarrollo de sistemas WEB y en la primera etapa de su adopción. Si las hay, las diferencias entre una empresa de desarrollo de software tradicional y una de sistemas WEB en niveles de dirección o gerenciales son muy pocas y pequeñas, es en el nivel de operación donde se desarrollan los sistemas y se coordina este desarrollo donde existen las mayores variaciones. Las empresas y equipos de desarrollo son afectadas por malas prácticas en todos sus niveles, sin embargo los procesos que por su naturaleza, complejidad e importancia tienen más impacto en conjunto de empresas y equipos son justamente el proceso de desarrollo de software y el de la administración de dicho desarrollo. Por estos motivos la guía, no se enfoca a todo el modelo de procesos, sino exclusivamente al nivel 1 y algunas actividades del nivel 2 de capacidad de los procesos de *Desarrollo y Mantenimiento de Software* y *Administración de Proyectos Específicos*.

5.1 Proceso

En la primera etapa se estudió MoProSoft[1] y referencias sobre ingeniería WEB[2]y[3]. A raíz de esto se determinaron las convergencias y divergencias de los temas y se determinó que la guía podría enfocarse únicamente a los procesos de desarrollo y mantenimiento de software y administración de proyectos específicos. Conforme se establecían las relaciones entre MoProSoft y las diversas propuestas de ingeniería WEB también se inició una investigación bibliográfica para ubicar técnicas específicas para desarrollo WEB de cada una de las tareas y/o productos de MoProSoft.

La segunda etapa se desarrolló en España con el Dr. Oscar Pastor, coautor de [2], y su equipo de trabajo. También se consultó, por Internet, con las Ing. Liliana León Azpiroz directora de proyectos multimedia de Eje-Z diseño y la Ing. Diana Campos Chalot directora de logística y programación WEB de BTC Technology Company. En esta etapa se discutieron las diversas técnicas halladas en la etapa anterior. Se seleccionaron aquellas que a juicio de los expertos académicos y las líderes de equipos de desarrollo WEB eran más adecuadas para equipos de desarrollo WEB pequeños. Esta selección y discusión también hizo evidente aspectos e inquietudes que faltaba cubrir por lo que se inició una segunda etapa de investigación bibliográfica. Conforme se seleccionaban técnicas y ejemplos se plasmaban en un borrador que hacia el final de la etapa formó una primera versión de la guía. En esta etapa se observó que la

parte de la cotización de los proyectos quedaba fuera de los proceso de MoProSoft que abarca la guía y para dar atención a este punto dentro de la guía se extendieron algunas tareas y se cambió su secuencia, con respecto a MoProSoft. El trabajo con el equipo del Dr. Oscar Pastor, además del análisis de las técnicas para desarrollo WEB, permitió incluir en esta lo referente al desarrollo basado en modelos o MDA que si bien aun no es viable para su uso cotidiano será importante en el mediano y largo plazo.

La etapa final consistió en revisar y corregir con la Dr. Hanna Oktaba el primer borrador de guía para asegurar la convergencia y sincronía de la guía con MoProSoft y validar los cambios y adaptaciones del mismo que se sugieren para el desarrollo WEB. Se valido el cambio de orden de actividades y se hicieron los ajustes necesarios para hacerlo completamente congruente. Así mismo simultáneamente a los ajustes a la guía se redactó y revisó la presente tesis.

5.2 Estructura

La “Guía para el desarrollo de aplicaciones WEB. Basada en el modelo de procesos para la industria de software MoProSoft.”, está conformada de las siguientes secciones.

La 1.- *Introducción*, Explica brevemente las razones y el contexto en que se generó la guía. Contesta a las preguntas. ¿Es esta guía para mí? ¿Qué puedo aprender de esta guía?

En la sección 2.- *Estructura*, se informa sobre las diferentes secciones y contenidos. Es solamente un pequeño listado de las secciones:

Características de los sistemas WEB

Dificultades más frecuentes en los desarrollos de sistemas WEB

Los involucrados más frecuentes en los desarrollos de sistemas WEB

Proceso sugerido para el desarrollo de sistemas WEB

Técnicas útiles en el desarrollo WEB

Productos del proceso de desarrollo de sistemas WEB

Referencias y enlaces.

Ejemplos.

Estas secciones contestan diversas preguntas: ¿Cómo son los sistemas WEB? ¿Qué problemas se enfrentan al desarrollarlos? ¿Quiénes deben intervenir en el desarrollo de un sistema WEB? ¿Qué hacer para desarrollar bien un sistema WEB y cómo hacerlo? Más allá del listado, no se presentan más explicaciones, pues se procuró crear una sección sencilla que diese rápidamente una idea integral de los contenidos de la guía. Los títulos de las diferentes secciones son en sí mismos la explicación y el contenido, emulando un ejercicio que se debe

buscar continuamente en el desarrollo WEB que consiste en dar información completa pero breve.

Las secciones 3.- *Características de los sistemas WEB* y 4.- *Dificultades en los desarrollos de sistemas WEB*, describen los aspectos y problemas que más afectan los procesos y tareas que abarca la guía. Cualquier modelo de procesos o metodología de trabajo implica cambio, pues requiere la adopción de nuevas prácticas y formas de trabajo.

Toda organización o equipo tiene un cierto grado de resistencia al cambio que se ve incrementado si no se comprende la utilidad de algunas de las prácticas propuestas. Estas secciones son las bases sobre las que se argumentan, explican y/o justifican diversas acciones, técnicas y productos a lo largo de la guía.

En la sección 5.- *Involucrados en el desarrollo de un sistema WEB* se listan los diversos roles que se involucran en el desarrollo de un sistema WEB. Este listado ayuda a identificar a los diversos participantes y afectados en el desarrollo de un sistema WEB. Esto es el primer paso para disminuir los problemas que causa el gran número de involucrados de diversas disciplinas y a su vez es indispensable para designar las tareas y responsabilidades adecuadamente.

La sección 6.- *Proceso sugerido para el desarrollo de sistemas* es el centro de la guía y dice que se debe hacer para desarrollar un sistema WEB. Presenta acciones y tareas basadas en el nivel 1 de los procesos *Desarrollo y Mantenimiento de Software y Administración de Proyectos Específicos* de MoProSoft, pero con adaptaciones para responder a las características y dificultades del desarrollo de los sistemas WEB.

La sección 7.- *Técnicas útiles en el desarrollo WEB* explica brevemente algunos métodos o consejos para elaborar los productos más importantes.

La sección 8.- *Productos* explica el proceso general de elaboración de cualquier producto y detalla la descripción, la utilidad y observaciones para cada uno.

Las secciones 7 y 8, así como los ejemplos anexos que complementan a esta última, responden a una de las principales inquietudes al conocer por primera vez modelos de procesos como MoProSoft. El problema radica en saber cómo hacer lo que el modelo de procesos indica que se debe hacer. En el caso particular de MoProSoft presenta un gran número de tareas referentes a la elaboración, consulta y/o modificación de productos que quien inicia en la adopción de un modelo de procesos generalmente nunca ha generado, consultado o inclusive oído mencionar. El que no haya claridad en el cómo realizar estos productos se debe a varios factores, destaca que para responder a preguntas como: ¿Cómo elaboro una cotización, un plan de riesgos, un plan de proyecto, etc.?, la mejor respuesta es “depende de cada proyecto”. Además si cubriera estos detalles un modelo como MoProSoft se volvería excesivamente extenso y

complicado, desviándose de objetivo original. La guía al estar enfocada a solo dos procesos y proyectos particularmente de desarrollo WEB y de equipos pequeños tiene mayores posibilidades de responder a esta inquietud. Con la inclusión de las dos secciones y los ejemplos se proporcionó a los lectores información sobre los productos y su elaboración.

Finalmente las secciones las secciones 9.- *Referencias bibliográficas*, 10.- *Referencias a aplicaciones* y 11.- *Sitios de interés* guían al lector tanto a los textos originales como a aplicaciones y sitios WEB para ayudarse en generación de ciertos productos y expandir sus conocimientos.

5.3 Notación, cambios y formato

En diversas partes de la guía no es necesario modificar lo propuesto por MoProSoft, o en las otras fuentes consultadas. Sin embargo, para que los miembros del equipo de desarrollo pongan especial atención en algún aspecto o tengan mayor cuidado sobre la relevancia y motivos por los que un actor, tarea o producto son necesarios, la guía ofrece información adicional. Esta información se agregó en una columna de observaciones en las tablas de secciones: 5.1 *Roles del equipo de desarrollo*, 5.2 *Roles de clientes y usuarios*, 6.2 *Actividades del Proceso de Administración de Proyectos Específicos*, 6.3 *Actividades del Proceso de Desarrollo y Mantenimiento de software* y 8.- *Productos*.

Como los propios sistemas WEB, la guía es un documento altamente interrelacionado. Los contenidos de los capítulos 3.- *Características de los sistemas WEB*, 4.- *Dificultades en los desarrollos de sistemas WEB*, 7.- *Técnicas útiles en el desarrollo WEB* y 8.- *Productos* se utilizan constantemente como referencia en toda la guía. Por esta razón los elementos de estas secciones no siguen la numeración del resto de las secciones sino que presentan cada elemento con un número secuencial pero antecedido de una letra C(características), D(dificultades), T(técnicas) y P(productos) respectivamente de esta forma cuando el texto se refiera a alguno de estos elementos como en el siguiente fragmento de las observaciones de la actividad W1.1 de la sección 6.2 “Hacer *P2 Bocetos, listas de sitios de similares y prototipos* disminuye las *D3 Dificultades de comunicación*. El *P3 Glosario* ayuda al mismo fin y se debe iniciar en este momento.” el lector puede identificar fácilmente de qué tipo de elemento se trata y complementar la información que proporciona el identificador del mismo.

Las actividades y tareas son el eje central de los procesos y por lo tanto de MoProSoft y la guía. Las actividades propuestas en la guía se basan en las de MoProSoft pero presentan diferencias con estas. Estos cambios responden tanto a los factores del desarrollo WEB como a la necesidad de presentar aislados procesos que forman parte de un modelo más grande. Las adecuaciones de formato son:

- La enumeración de las tareas se hizo en forma análoga a MoProSof pero cambiando la letra A por una W y conservando la enumeración original de la

tarea en MoProSoft entre paréntesis. Las tareas sin esta referencia son aquellas fueron que agregadas específicamente para el desarrollo WEB.

- Se agregó la columna de observaciones.
- En cada tarea se subrayaron los productos que se generan o modifican en la misma y se destacaron con **negritas** aquellos que se deben consultar para poder realizarla.

Ejemplo: Actividad W5.2 de sección 6.3 Actividades del Proceso de Desarrollo y Mantenimiento de software.

PR RPU RE	<p>W5.2 (A5.2) Realizar integración y pruebas.</p> <ul style="list-style-type: none"> • Integrar los componentes en subsistemas o en el sistema de P19 Software y aplicar las pruebas siguiendo el P17 Plan de Pruebas • Corregir los defectos encontrados o programarlas para el siguiente ciclo. • Generar el <u>P20 Registro de errores</u> 	<p>Por la salvedad de que es recomendable probar con más de un navegador. Estas pruebas se pueden ejecutar prácticamente igual que en una aplicación tradicional. Es necesario utilizar el <i>P20 Registro de errores y cambios</i> y se recomienda usar herramientas específicas para este fin.</p> <p>Como se indica en <i>T10 Técnicas para probar las aplicaciones WEB</i> es válido apoyarse en los propios clientes y usuarios para lograr una mayor detección de fallos. Si el proyecto lo permite es....</p>
-----------------	---	--

En cuanto a los procesos se les hicieron algunos cambios para poder presentarlos independientemente.

- Se cambió, el nombre de la actividad inicial en el proceso *6.2 Administración de Proyectos Específicos*, Planificación por Aceptación. En esta actividad se integraron y cambiaron tareas que responden a la aceptación del proyecto que en MoProSoft se cubren en otros procesos.
- Se reubicaron tareas como “W1.13 (A1.7) Establecer los P12 Requerimientos de Adquisición...” que como su enumeración indica en el MoProSoft es la séptima y en la guía es la terciaba. La reubicación de esta y otras tareas es para atender a la necesidad de generar una cotización en una etapa temprana de la fase de aceptación, pues es uno de los principales requisitos de los clientes.
- Se agregaron tareas como la “W3.2 Probar y elegir las tecnologías a utilizar” que son necesarias en los desarrollos WEB y no están presentes en MoProSoft.

- Se agregaron dentro del proceso de *Administración de Proyectos Específicos* los señalamientos de cuando ejecutar las diversas actividades del proceso *Desarrollo y Mantenimiento de Software* para presentar una unidad y secuencia clara de actividades y tareas.

5.4 Validación

La guía es una conjunción de diversas propuestas y técnicas con un modelo de procesos. Para validar las propuestas plasmadas en la misma se trabajó con expertos de diversas aéreas.

La concordancia con MoProSoft fue avalada y revisada por la Dra. Hanna Jadwiga Oktaba, directora del proyecto MoProSoft.

La selección de técnicas y los contenidos referentes a la WEB fueron revisados por el Dr. Oscar Pastor coautor de [2] y distinguido investigador de Ingeniería de Software de la Universidad Politécnica de Valencia.

También se presentó toda la guía y se evaluó la viabilidad de aplicación de diversas técnicas de desarrollo WEB en equipos pequeños con dos directoras de áreas de desarrollo WEB: Ing. Liliana León Azpiroz directora de proyectos multimedia de Eje-Z diseño e Ing. Diana Campos Chalot directora de logística y programación WEB de BTC Technology Company. La selección se hizo discutiendo con las Ingenieras las diversas técnicas encontradas para las diversas etapas del desarrollo WEB y basándose en su experiencia y criterio para evaluar si les sería práctico y viable implementarlas. En particular con la Ing. Liliana León se ha continuado el trabajo y que se está poniendo en lo sugerido en la guía en proyectos reales.

6.- Conclusiones

Para el desarrollo de la presente tesis se estudiaron las mejores prácticas para desarrollo de software propuestas por MoProSoft. Así como mejores prácticas para administración de proyectos en general y proyectos WEB presentados en [2], [3] y [4]. Esto se complementó con investigaciones sobre técnicas aplicables en desarrollos WEB en tareas más específicas como el levantamiento de requerimientos, la modelación, el diseño de la navegación, la cotización, etc. Finalmente se verificó y complemento esta información con la asesoría directa de académicos expertos como la Dra. Hanna Oktaba y el Dr Oscar Pastor, así como de las Ing. Liliana León y Diana Campos, ejecutivas que enfrentan día a día al desarrollo de aplicaciones WEB. La investigación bibliográfica produjo una gran cantidad de información. Las asesorías sirvieron para seleccionar aquella aplicable a los objetivos y contexto de la guía. Con esta combinación de información y selección se integró una guía para desarrollo WEB basada en MoProSoft para equipos pequeños de desarrollo, que se estén formando o migrando su modelo de procesos a MoProSoft.

La conjunción de las diversas experiencias, opiniones y conocimientos de todos los involucrados en el desarrollo de esta tesis y la guía ha sido enriquecedor. Se generó “una guía, basada en los proceso de Administración de Proyectos Específicos y Desarrollo y Mantenimiento de software de MoProSoft[1], que integre técnicas especializadas en desarrollo de sistemas WEB, enfocada en apoyar a equipos pequeños, de desarrollo de aplicaciones WEB, en formación” como era el objetivo inicial. Durante este trabajo el autor experimento un proceso formativo que complementó los conocimientos adquiridos en los diversos cursos de la Maestría.

7.- Trabajo a futuro

El ámbito y los temas del desarrollo WEB y los modelos de procesos son mucho más amplios de lo abarcado en la guía. Esta ha sido desde su planteamiento concebida como un apoyo que pretende clarificar como seguir un modelo de procesos como MoProsof al desarrollar sistemas WEB con un equipo de trabajo pequeño. Así mismo el sustento de la guía es el juicio de expertos pero hace falta ponerla en práctica. Por lo anterior es posible ampliar el trabajo realizado en esta guía en diversas áreas como son:

- Documentar y evaluar los resultados de la aplicación de la guía en diversas empresas y equipos de trabajo, para comprobar su efectividad y mejorar sus deficiencias.
- Incrementar el número de niveles de MoProSoft que cubre la guía.
- Aumentar el público objetivo a empresas y equipos medianos, lo cual entre otras cosas permitiría la aplicación de técnicas más complejas como las pruebas de desempeño propuestas en [29], durante las diversas actividades del desarrollo de un sistema WEB.
- Hacer guías similares para el desarrollo de otro tipo de sistemas, como aplicaciones de tiempo real, de celulares, de escritorio, etc.
- Proponer nuevas técnicas de desarrollo WEB. La guía se concentró en técnicas y propuestas ya existentes, pero durante su desarrollo y en particular durante la colaboración sobre WEB sostenible con el Dr. Oscar Pastor y su equipo surgieron algunas propuestas interesantes e innovadoras.

Referencias:

- [1] **Modelo de Procesos para la Industria de Software MoProSoft Por Niveles de Capacidad de Procesos Versión 1.3.** Secretaría de economía Agosto 2005.
- [2] O. Pastor, D. Schwabe, G. Rossi, et al. **Web Engineering: Modelling and Implementing Web Applications** Springer London Ltd Octubre 2007.
- [3] G. Kappel, B. Pröl, et al. **WEB Engineering: The Discipline of Systematic Development of Web Applications**, John Wiley & Sons, Ltd Julio 2006.
- [4] **A Guide to the Project Management Body of Knowledge** third edition. Project Management Institute 2004.
- [5] Wolf-Gideon Bleek, Martti Jeenicke, Ralf Klischewski **Developing Web-based Applications through e-Prototyping** Proceedings of the 26 th Annual International Computer Software and Applications Conference IEEE 2002.
- [6] Eric Holter **Client vs. Developer Wars** Newfangled Web Factory 2006.
- [7] Davide Bolchini , John Mylopoulos **From Task-Oriented to Goal-Oriented Web Requirements Analysis** Proceedings of the Fourth International Conference on Web Information Systems Engineering IEEE 2003.
- [8] Victoria Torres, Javier Muñoz, Vicente Pelechano **A Model Driven Method for the Integration of Web Applications** Proceedings of the Third Latin American Web Congress IEEE 2005.
- [9] Rashid Ahmad2, Zhang Li, Farooque Azam **Web Engineering: A New Emerging Discipline** International Conference on Emerging Technologies IEEE 2005.
- [10] Lasse Vogelsang, Peter Carstensen **New Challenges for the Collaboration in Web-based Information Systems Development** IEEE 2001
- [11] <http://www.software.net.mx/desarrolladores/prosoft/>
- [12] <http://www.iso.org/>
- [13] <http://alarcos.inf-cr.uclm.es/Competisoft/>
- [14] <http://www.w3.org/>
- [15] <http://www.sei.cmu.edu/cmmi/>
- [16] **SWEBOK.** Software engineering Coordinating Committee, Computer Society, Software Engineering Institute. 2001.
- [17] **Stakeholder Analysis.** http://www.mindtools.com/pages/article/newPPM_07.htm
- [18] Kevin Mullet, Darrell Sano **Designing Visual Interfaces, communication oriented techniques** Sun Press, 1995
- [19] <http://webpatterns.org/>
- [20] <http://www.ui-designpatterns.org/>
- [21] http://www.w3schools.com/media/media_intro.asp
- [22] **Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification**
<http://www.w3.org/TR/CSS21/>
- [23] **XHTML™ 1.0 The Extensible HyperText Markup Language** <http://www.w3.org/TR/xhtml1/>
- [24] **RFC1123 Requirements for Internet Hosts -- Application and Support**
<http://tools.ietf.org/html/rfc1123>

- [25] **RFC 1122 Requirements for Internet Hosts -- Communication Layers**
<http://tools.ietf.org/html/rfc1122>
- [26] **W3schools Web Building tutorial** http://www.w3schools.com/site/site_intro.asp
- [27] HUBERT ZIMMERMANN **OS1 Reference Model-The ISO Model of Architecture for Open Systems Interconnection** IEEE 1980
- [28] Cisco systems **Internetworking Technology Handbook**
http://www.cisco.com/en/US/docs/internetworking/technology/handbook/ito_doc.html
- [29] Zao-Bin, Deng-Wen WEI **Evaluating the Performance and Scalability of Web Application systems** IEEE 2005

Anexo A - “Guía para el desarrollo de aplicaciones WEB. Basada en el modelo de procesos para la industria de software MoProSoft.”

Postgrado UNAM

Guía para el desarrollo de aplicaciones WEB.

**Basada en el modelo de procesos para la industria de
software MoProSoft.**

Rubén Schaffer Levine

Índice

1.- Introducción.....	5
2.- Estructura.....	5
3.- Características de los sistemas WEB.....	6
C1 Comunicación	6
C2 Interacción	6
C3 Estética y diseño gráfico preponderantes	6
C4 Intuitivos y autoexplicativos	6
C5 Vinculados	6
C6 Cambio continuo	6
C7 Tecnología diversa.....	7
C8 Usuarios simultáneos y diversos.....	7
C9 Múltiples involucrados con múltiples áreas de especialización	7
4.- Dificultades en los desarrollos de sistemas WEB.....	8
D1 Necesidades de usuarios frecuentemente relegadas.....	8
D2 Requisitos altamente cambiantes	8
D3 Dificultades de comunicación	8
D4 Tecnología diversa y cambiante	8
5.- Involucrados en el desarrollo de un sistema WEB.....	9
5.1 Roles del equipo de desarrollo:	9
5.2 Roles de clientes y usuarios:.....	11
6.- Proceso sugerido para el desarrollo de sistemas WEB	12
6.1 Notas sobre los procesos.....	12
6.2 Actividades del Proceso de Administración de Proyectos Específicos	13
6.3 Actividades del Proceso de Desarrollo y Mantenimiento de software	19

7.- Técnicas para desarrollo WEB.....	22
T1 Análisis de requerimientos enfocado a objetivos	22
T2 Prototipos Grises	24
T3 Prototipos operativos	24
T4 Técnicas para modelación	25
T5 Uso de patrones	25
T6 Técnicas para el diseño de arquitecturas	25
T7 Técnicas de Diseño gráfico.....	26
T8 Técnicas para el diseño de navegación	27
T9 Técnicas para presentar la información.....	27
T10 Técnicas para probar las aplicaciones WEB.....	28
8.- Productos.....	28
P1 Descripción del Proyecto	29
P2 Bocetos, Listas de Sitios Similares y Prototipos	30
P3 Glosario.....	31
P4 Mapa de Involucrados	32
P5 Descripción del Producto y Entregables.....	32
P6 Plan de Manejo de Riesgos	33
P7 Plan de Comunicación.....	33
P8 Cotización	34
P9 Repositorio del Proyecto.....	34
P10 Proceso General de Desarrollo.....	35
P11 WBS.....	35
P12 Requerimientos de Adquisiciones y Capacitación	35
P13 Calendario.....	36
P14 Plan de Proyecto	36
P15 Protocolo de Cambio	37
P16 Especificación de Requerimientos	37
P17 Plan de Pruebas	38
P18 Análisis y Diseño	39
P19 Software.....	39
P20 Registro de Errores y Cambios	39
P21 Manual de Mantenimiento.....	40
P22 Base de Conocimientos.....	40
P23 Directorio.....	40

Referencias bibliográficas:.....	41
Referencias a aplicaciones	42
Servidores:	42
Administración de proyectos:	42
Editores e IDEs:	42
Diseño gráfico:.....	42
Control de código:.....	42
Control de errores:.....	42
Sitios de interés.....	43
Ingeniería de software:	43
Patrones para web:.....	43
Tutoriales y referencias de lenguajes:	43
Ejemplos Anexos:.....	44

1.- Introducción

Todo profesionista que viva de desarrollar sistemas se enfrenta en algún momento a retos de interacción, organización y trabajo humanos radicalmente diferentes a la vez que inevitablemente vinculados a la tarea del desarrollo del software. Los desfases en los calendarios, los continuos cambios, las horas extra y sistemas con fallas o que no ahorran tanto trabajo como deberían, parecen ser constantes en la industria de software y el sector de sistemas WEB no es la excepción. Una de las muchas aportaciones para disminuir estos problemas es MoProSoft, el Modelo de Procesos para la Industria de Software en México. Es resultado de un esfuerzo conjunto de la industria de software y el gobierno mexicano para ayudar a las empresas del sector a operar de manera más eficiente y eficaz. Este modelo de procesos resume y adapta a la realidad mexicana las prácticas, normas y modelos más efectivos utilizados por la industria de software a nivel mundial.

Dentro del contexto de la industria de software el desarrollo de sistemas WEB ha cobrado vital importancia por sus efectos sociales, versatilidad y costos. A pesar de esta creciente importancia por la relativa novedad de las tecnologías WEB y su gran dinamismo la cantidad de empresas y equipos de trabajo adecuadamente capacitados para desarrollar eficientemente sistemas WEB es muy pequeña. Es en este contexto que se ha desarrollado esta que guía que es para pequeños equipos de desarrollo de aplicaciones WEB que deseen mejorar sus metodologías de trabajo. Se enfoca principalmente en el primer nivel de MoProSoft por lo tanto es ideal para aquellos equipos de desarrollo que estén migrando de procesos totalmente empíricos hacia MoProSoft o equipos que estén iniciando sus actividades. Da las bases y los primeros procesos para mejorar la producción de sistemas WEB y facilita la posterior adopción de MoProSoft en todos los niveles y tareas de la empresa o equipo.

2.- Estructura

Esta guía está formada de varias secciones que son complementarias. Estas son:

Características de los sistemas WEB

Dificultades más frecuentes en los desarrollos de sistemas WEB

Los involucrados más frecuentes en los desarrollos de sistemas WEB

Proceso sugerido para el desarrollo de sistemas WEB

Técnicas útiles en el desarrollo WEB

Productos del proceso de desarrollo de sistemas WEB

Referencias y enlaces.

Ejemplos.

Se sugiere que se lean todas las secciones al menos una vez pero están diseñadas para permitir su consulta directa en cualquier momento.

3.- Características de los sistemas WEB

Cada sistema es único sin embargo hay características que se repiten en casi todos los sistemas WEB. Estas características están directamente relacionadas con la naturaleza de la propia Internet, con las funciones y usos de los sistemas así como con las dificultades en el desarrollo de los mismos.

C1 Comunicación

La función intrínseca de todos los componentes de Internet es transmitir y recibir información, o en otros términos: comunicar. “La comunicación es obviamente el factor subyacente del éxito de la WEB a la vez que la mayor fuente de complejidad y retos para el desarrollo WEB”[2, pag:38]. Los sistemas de software tradicionales almacenan, procesan y organizan la información, pero en comparación con los sistemas WEB, sus posibilidades de transmitirla son limitadas. Los sistemas WEB pueden o no cumplir alguna o todas las funciones “tradicionales”, pero inevitablemente deben transmitir información. En este contexto la generación, actualización, confiabilidad y adecuación de la información de un sitio o sistema WEB es un factor de singular importancia.

C2 Interacción

Un factor adicional, que da aún más versatilidad y complejidad a la WEB, es la bidireccionalidad del flujo de la información. Los usuarios no solo reciben información, también pueden enviarla. Esto permite que los usuarios puedan asumir una amplia gama de roles al usar un sistema. Pueden ser desde simples lectores, como en el caso de un portal de noticias, hasta los responsables casi absolutos de los contenidos de los sistemas, como ocurre en Wikipedia. La tendencia general es involucrar a los usuarios en algún grado en todos los sistemas. Los sitios de noticias permiten a los lectores hacer comentarios sobre las mismas, los de ventas permiten evaluar los productos, etc, etc.

C3 Estética y diseño gráfico preponderantes

El esquema Cliente/Servidor implica que el cambio de un sistema a otro es simple e inmediato para los usuarios. Internet da a los sistemas instantáneamente millones de usuarios en potencia, pero a la vez miles de sistemas competidores. La utilidad, la facilidad de uso, el precio, la accesibilidad, la oportunidad y el atractivo visual son los factores más significativos para retener usuarios. Ante la gran oferta, los usuarios hacen juicios velozmente basándose a veces en un solo vistazo. Esto hace que la estética y el diseño gráfico sean más importantes que en los sistemas tradicionales, pues resultan claves para retener al usuario lo suficiente para que decida evaluar las otras características.

C4 Intuitivos y autoexplicativos

Los mismos factores que causan C3 también fuerzan que los sitios sean fáciles de usar e intuitivos. Los usuarios que decidan usar el sitio, lo abandonarán si no logran lo que pretenden fácilmente.

C5 Vinculados

La abundancia de sitios y la facilidad de cambiar de uno a otro no representa únicamente retos, también es una oportunidad. Los sistemas WEB pueden valerse de otros para atraer visitantes y cubrir funciones y contenidos si gastar en implementarlos o mantenerlos.

C6 Cambio continuo

La tecnología de Internet permite el cambio continuo y además el mercado lo exige. La información es un bien perecedero, y el cambio continuo una herramienta para atraer a más usuarios. Estos factores refuerzan la necesidad de cambios muy frecuentes a todos los niveles en la mayoría de los sistemas WEB.

C7 Tecnología diversa

La estandarización de los protocolos y la tecnología Cliente/Servidor permite la integración de una red muy heterogénea. El hecho de que la interoperabilidad sea posible fomenta la diversidad de los sistemas. La variedad de sistemas existe tanto en características relacionadas al hardware como al software. En cuanto al hardware existen una gran variedad de tipos de computadoras, tamaños y resoluciones de la pantalla así como diversos dispositivos de entrada como ratones, pantallas táctiles, teclados, lápices apuntadores, etc. En el terreno del software varía el navegador, su versión, y los plugins (complementos) que tenga para interpretar formatos adicionales de archivos.

Los estándares permiten que sistemas muy diversos puedan interactuar y ser reemplazados sin alterar a los sistemas restantes pero no cambia las diferencias entre un sistema y otro. Así aunque un celular o una PC puedan acceder a un sistema WEB específico el celular nunca podrá presentar en su pequeña pantalla la información del mismo modo en que se esta se presenta en la pantalla de PC. En el caso del software es posible que en dos sistemas idénticos en hardware y con el mismo navegador uno pueda desplegar animaciones multimedia y otro no. Esto puede ocurrir si el plugin para tal fin se instaló sólo en un navegador.

C8 Usuarios simultáneos y diversos

La diversidad de los usuarios es igual o más compleja que la de la tecnología. Hay que considerar factores variables en cada individuo como el idioma, la familiaridad con la información del sistema, el tiempo disponible, etc. También es necesario evaluar si la información es sensible al acceso simultáneo. En sitios dependientes de operaciones con fechas y horas se debe atender las confusiones y problemas que puede causar el acceso desde lugares con otro uso horario. La meta generalmente es satisfacer lo mejor posible a un amplio número de miembros de un segmento de mercado específico, pues satisfacer plenamente a todos es imposible.

C9 Múltiples involucrados con múltiples áreas de especialización

La transferencia de información es crítica en Internet y no es un elemento puramente tecnológico. Intervienen muchos otros factores relativos a la experiencia y cognición humana. Este fenómeno ocasiona la necesidad de roles e involucrados en los sistemas WEB que son inexistentes o muy poco frecuentes en los sistemas tradicionales. Debe haber quienes generen la información, quienes diseñen como organizarla para hacerla accesible, quienes la actualicen, quienes la validen y evidentemente quienes la consulten. Para cumplir estas tareas suelen ser necesarias personas de diversas especialidades.

4.- Dificultades en los desarrollos de sistemas WEB

Las características comunes de los sistemas WEB hacen que a su vez se presenten con frecuencia diversas dificultades en casi todos los desarrollos de sistemas WEB. Las más significativas de estas son:

D1 Necesidades de usuarios frecuentemente relegadas

Los criterios para diseñar los sistemas suelen considerar a las preferencias, necesidades y capacidades de los usuarios. El uso a distancia, que permite Internet, implica que no sea fácil o viable un contacto presencial con ellos. Por esta razón es común que los usuarios estén físicamente ausentes durante el desarrollo de los sistemas WEB. Esta ausencia presencial ocasiona muchas veces que se omitan algunos de los aspectos importantes. Por la misma razón muchos otros aspectos se diseñen con base en suposiciones, interpretaciones y preferencias, de otros involucrados, como el cliente patrocinador, con mayor injerencia directa en el proyecto. Generalmente dichos involucrados reflejan sus gustos personales y no las necesidades de quienes serán los verdaderos usuarios.

D2 Requisitos altamente cambiantes

Aunado al factor de cambio constante de Internet, los propios requisitos de los sistemas suelen cambiar antes de que estos estén terminados. El iniciar con objetivos muy generales, las nuevas ideas que surgen a partir del aprendizaje que causa la interacción de los diversos involucrados y el mismo número de involucrados, son solo algunos ejemplos de posibles causas. En cualquier caso el cambio de requisitos es muy frecuente y sus efectos son muy importantes en los desarrollos de sistemas WEB.

D3 Dificultades de comunicación

Las comunidades generan sus propios sub-lenguajes. Dan usos particulares a expresiones generales, tienen expresiones propias y utilizan términos que desconocen otros grupos de personas. En los desarrollos WEB, aun en el caso de sistemas pequeños, intervienen personas de diversas disciplinas. Esto ocasiona dificultades de comunicación que suelen ser muy costosas.

D4 Tecnología diversa y cambiante

La arquitectura de Internet y el uso de estándares permiten la heterogeneidad pero no eliminan todos los problemas que ésta causa. Estos incluyen desde diferencias en características físicas (como el tamaño de la pantalla) hasta pequeñas sutilezas en la forma y detalle con que cada navegador interpreta un estándar. Aun sobre las bases de los estándares más generales como HTML, los navegadores tienen algunas diferencias que pueden causar serias dificultades en algunos sistemas. Además hay un gran número de tecnologías complementarias que permiten desarrollar funciones complejas a bajo costo, pero que no están ampliamente difundidas en los sistemas clientes. La oferta de dispositivos, protocolos, formatos, navegadores y servidores cambian constantemente. En ocasiones el sistema que estamos desarrollando o manteniendo se puede ver afectado por un cambio en un componente que no tiene efecto en el 99.9% de los sistemas. La dificultad para programar ciertas funciones o acceder a otros sistemas de apoyo como bases de datos, varía entre los diversos lenguajes con que se puede programar el sistema en el servidor. Nuevas tecnologías pueden ser muy sencillas de implementar pero no ser utilizadas por la mayoría de los clientes objetivo. La diversidad de navegadores y servidores puede volver compleja y crítica la selección de los lenguajes y tecnologías utilizados.

5.- Involucrados en el desarrollo de un sistema WEB

Las siguientes tablas muestran los roles más comunes de los involucrados en los desarrollos WEB. Se consideran tanto los miembros del equipo de desarrollo como los clientes y los usuarios.

5.1 Roles del equipo de desarrollo:

Rol	Responsabilidades	Conocimientos	Herramientas y apoyos
RAPE Responsable de Administración de Proyectos Específicos	<ul style="list-style-type: none"> - Establecer con el cliente encargado los objetivos del sistema. - Coordinar a todos los miembros del equipo. - Comunicación y relación oficial con el cliente - Asegurar que el sistema cumpla sus objetivos - Procurar recursos necesarios 	<ul style="list-style-type: none"> - Administración de proyectos - Desarrollo WEB básico 	<ul style="list-style-type: none"> - Email - Agenda - Teléfono - Presentaciones - Diagramas - Pizarrón - Etc.
AN Analista	<ul style="list-style-type: none"> - Establecer los requerimientos en términos entendibles para clientes y desarrolladores - Determinar que requerimientos cambiarán más rápidamente - Identificar perfiles de los usuarios - Determinar las limitantes técnicas - Detallar qué información se usara en el sitio y quién la proveerá 	<ul style="list-style-type: none"> - Desarrollo WEB básico - Tendrá que aprender sobre temas relativos al sitio y las actividades de los clientes y usuarios 	<ul style="list-style-type: none"> - Sitios WEB relacionados o similares - Entrevistas con los clientes - Prototipos: grises, dumies, power point, etc.
DN Diseñador de Navegación	<ul style="list-style-type: none"> - Diseñar la navegación (pantallas con sus funciones y ligas) del sistema en función de los objetivos, requerimientos e información 	<ul style="list-style-type: none"> Deseables: - Conocimientos sobre cómo se percibe de la información 	<ul style="list-style-type: none"> - Sitios WEB relacionados o similares - Prototipos: grises, dumies, power point, etc.
DU Diseñador de Interfaz de Usuario	<ul style="list-style-type: none"> - Diseñar la interfaz del sistema en función de la navegación, los objetivos, requerimientos e información 	<ul style="list-style-type: none"> - Diseño grafico - Manejo de herramientas graficas Deseables: - XHTML - CSS 	<ul style="list-style-type: none"> - Sitios WEB relacionados o similares - Patrones de interfaz - Templates - Circulo del color - GIMP, PhotoShop, PowerPoint, CorelDraw, etc.
DI Diseñador	<ul style="list-style-type: none"> - Establecer la arquitectura y módulos del sistema 	<ul style="list-style-type: none"> - Noción de las habilidades y conocimientos de los programadores - Técnicas de 	<ul style="list-style-type: none"> - Sistemas y librerías previos - Sistemas y librerías de código abierto - Catálogos de

		<ul style="list-style-type: none"> programación - Patrones - Plataformas existentes - Lenguajes - Estándares - Limitaciones técnicas del sistema 	<ul style="list-style-type: none"> patrones - Diagramadores UML
PR Programador	- Construir el sistema.	<ul style="list-style-type: none"> - Técnicas de programación - Patrones - Plataformas existentes - Lenguajes - Estándares - Limitaciones técnicas del sistema 	<ul style="list-style-type: none"> - Eclipse - Dreamweaver - Visual Studio - NotePad++ - HTMLKit
RE Revisor	- Verificar que los componentes separados y el sistema integrado funcionen como se espera	- Conocimiento de técnicas de revisión y experiencia en el desarrollo y mantenimiento de software.	<ul style="list-style-type: none"> - Documentación - Sistema <p>Deseable:</p> <ul style="list-style-type: none"> - Herramientas especializadas en pruebas y/o estrés de servidores
RDM Responsable de Desarrollo y Mantenimiento de Software	- Coordinar la construcción del sistema	- Conocimiento y experiencia en el desarrollo y mantenimiento de software.	<ul style="list-style-type: none"> - IDEs - Subversion, SourceSafe
RM Responsable de Manuales	- Elaborar los manuales	- Conocimiento en las técnicas de redacción y experiencia en el desarrollo y mantenimiento de software.	
RPU Responsable de Pruebas	- Coordinar la elaboración de las pruebas	- Conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema.	- Bugzilla
ET Equipo de Trabajo		- Conocimiento y experiencia de acuerdo a su rol.	

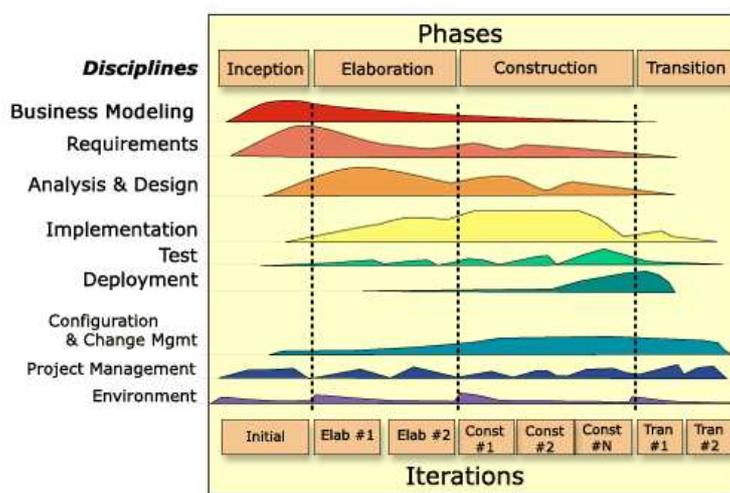
5.2 Roles de clientes y usuarios:

Rol	Descripción	Participación	Observaciones para miembros del equipo de desarrollo *de donde salen* de lo investigado y consultado-
CP Cliente Patrocinador	Es quien paga por el sistema	- Provee los recursos - Valida el sistema al final	Si el CP no es el encargado directo del proyecto, se debe documentar a detalle todas las interacciones con el encargado CAE para prever y solucionar malos entendidos.
CAE Cliente Administrador Encargado	El responsable del proyecto en la practica	- Coordina los diferentes recursos que deben proporcionar los clientes a los desarrolladores. - Valida el sistema durante todas sus etapas	- Cerciorarse de identificarlo formalmente junto con el patrocinador - No saltar su autoridad - Infórmale y acordar con él todos los cambios en los requerimientos, plazos y pagos - Informar de cualquier falla de los CEIs que afecte plazos o costos - No aceptar a priori todas sus observaciones o caprichos; discutir las y recordarle constantemente los objetivos del sistema y los posibles deseos y criterios de los usuarios y el patrocinador
CEI Clientes Encargados de Información	- Tienen la responsabilidad sobre la información que presentará o recibirá el sitio	- Proveer y mantener o recibir y procesar información del sistema.	- Identificarlos y aclarar con ellos y el cliente encargado que su participación es necesaria para que el sistema se pueda crear y mantener - Considerarlos especialmente en el diseño - Procurar que se vuelvan usuarios con privilegios y puedan manejar la información que les compete por si mismos - Prever que se tardarán en sus entregas de información - Solicitar su información frecuentemente - Condicionar plazos y costos a su participación y recordar dicho condicionamiento con antelación a los plazos
US Usuarios	Son los que utilizarán el sistema	Uso del sistema	- Identificarlos a todos; habrá usuarios dentro de los clientes pero además usuarios externos (cibernautas). No olvidarlos aunque sean anónimos. Definirlos por grupos ej: Médicos, profesores, pasantes, hombres con ingresos X de 24 a 30 años, etc. - NO OLVIDARLOS, parece absurdo pero es muy común anteponer los criterios de los clientes a los de usuarios. Que los clientes conozcan o entiendan la aplicación no implica que los usuarios podrán hacerlo.

6.- Proceso sugerido para el desarrollo de sistemas WEB

6.1 Notas sobre los procesos

Los últimos modelos de procesos de desarrollo de software no nos indican una secuencia de pasos sino una serie de temas o áreas a cubrir. Aunque, por ejemplo, es claro que es indispensable elaborar los requerimientos antes de iniciar el desarrollo, no se abandonan los primeros al iniciar lo segundo. Se trabaja simultáneamente en diversas áreas aunque con distinta intensidad, lo que mejor ilustra esta situación es el siguiente cuadro del RUP (Rational Unified Process)[22] en el que las áreas coloreadas en cada renglón señalan el esfuerzo dedicado a cada disciplina en las diferentes fases. Por ejemplo se puede ver que al inicio se trabaja mucho en los requerimientos pero aun durante la construcción se sigue trabajando en estos.



MoProSoft no señala áreas de trabajo señala diversas actividades y productos específicos a desarrollar. A pesar de estas diferencias de forma, la filosofía de trabajo simultáneo en diversas disciplinas se mantiene, en este caso aplicada a las actividades y productos. Esto implica que los productos evolucionan y se refinan a lo largo del proyecto. Dos de los productos, solicitudes de cambios y el control de la configuración, sirven para administrar y documentar las modificaciones.

Los productos son instancias de información de diversa naturaleza. Existen documentos como el plan de proyecto, plan de comunicación, descripción del proyecto, etc... que aportan información concerniente al proyecto, otros documentos contienen conocimientos sobre el sistema en sí, como son los requerimientos, la configuración del sistema, los reportes de verificación, etc. Finalmente hay productos como la solicitud de cambios y las lecciones aprendidas que contienen información que ayuda a controlar el proceso de desarrollo o a las organizaciones y equipos involucrados.

En particular en los sistemas WEB los propios requerimientos cambian mucho conforme los clientes aprenden sobre la WEB y organizan la información para el sistema y los desarrolladores conocen y entienden la actividad de los clientes. Por eso las primeras fases del desarrollo pueden enfocarse en el levantamiento de los requerimientos y en la elaboración de prototipos descartables para acelerar dicho aprendizaje y desencadenar los cambios en etapas más tempranas.

El desarrollo de un sistema WEB incluye información que solo los clientes pueden determinar y generar. Por lo tanto los clientes deben trabajar para el desarrollo del sistema independientemente de cuantos desarrolladores se involucren. El trabajo de los clientes es indispensable e insustituible y

además es adicional a las actividades que ya desempeñan. Esto suele causar retrasos y dificultades a los desarrolladores. Una forma sencilla de disminuir estos problemas es documentar claramente todo lo que se requiere del cliente y establecer claramente las formas de comunicación y en particular de solicitar y aprobar cambios.

A continuación se presenta una lista de actividades basadas en los procesos de Administración de Proyectos Específicos y Desarrollo y Mantenimiento de Software de MoProSoft. Hay que recordar que el orden señala una secuencia deseable de para el inicio de las actividades pero como se explicó anteriormente se debe trabajar simultáneamente en diversas áreas o productos a lo largo del proyecto. Cada tarea debe tener un responsable cuyo rol se especifica junto a ésta. A las tareas sugeridas para web se le numero con el prefijo W y en las tareas en que así aplica se puede leer entre paréntesis el número correspondiente a la tarea en MoProSoft. Adicionalmente en cada tarea se subrayaron los productos que se generan o modifican y se destacaron con **negritas** aquellos que se deben consultar para poder realizarla. Así mismo toda la información presentada en otra sección de la guía tiene un referente al elemento al que se refiere, por ejemplo P1 Descripción del Proyecto es un producto que se puede encontrar en el apartado P1 de la sección de productos y D3 Dificultades de comunicación. Una dificultad que está en el apartado D3 de la sección de dificultades.

6.2 Actividades del Proceso de Administración de Proyectos Específicos

Rol	Descripción	Observaciones
A1. Aprobación		
CP CAE RAPE	<p>W1.1 (A1.1) Generar con el Cliente Patrocinador y el Cliente Administrador Encargado la <u>P1 Descripción del Proyecto</u>.</p> <ul style="list-style-type: none"> • Generar <u>P2 Bocetos, listas de sitios de similares y prototipos</u>. • Generar <u>P3 Glosario</u>. • Analizar, discutir y establecer el alcance y la magnitud del proyecto • Generar <u>P1 Descripción del Proyecto</u>. 	<p>Los proyectos WEB se caracterizan por sus <i>D2 Requisitos altamente cambiantes</i> definir claramente todos los elementos de la descripción del proyecto: Descripción del Producto, Objetivos, Entregables y Alcance. Ayuda a disminuir el número y el impacto de los cambios. Además es indispensable para poder negociar clara y abiertamente cuando estos ocurran.</p> <p>Hacer <i>P2 Bocetos, listas de sitios de similares y prototipos</i> disminuye las <i>D3 Dificultades de comunicación</i>. El <i>P3 Glosario</i> ayuda al mismo fin y se debe iniciar en este momento.</p>
RAPE (otros)*	<p>W1.2 A partir de la P1 Descripción del Proyecto identificar a todos involucrados, y sus expectativas, influencia, y responsabilidades generando el <u>P4 Mapa de Involucrados</u>.</p> <ul style="list-style-type: none"> • Generar el <u>P23 Directorio</u>. <p>* El CAE y/o el RDM pueden ayudar a elaborar un primer listado, que debe ser complementado con entrevistas con todos involucrados con quienes esto sea factible.</p>	<p>Los sistemas Web tienen <i>C9 Múltiples involucrados con múltiples aéreas de especialización</i> que tienen diversos grados de interés e influencia sobre el sistema. Identificar y considerar estas personas y factores durante la elaboración de los diversos planes disminuye los <i>D2 Requisitos altamente cambiantes</i> así como diversos riesgos.</p>
CP CAE RAPE	<p>W1.3 Revisar y ajustar la <u>P1 Descripción del Proyecto</u> considerando lo plasmado en el P4 Mapa de Involucrados.</p>	<p>Es común que entre los involucrados con mucha influencia existan discrepancias entre sus expectativas sobre el sistema. Es necesario tratar estos temas con el CAE y/o el CP para apoyarse</p>

Rol	Descripción	Observaciones
		en su influencia y autoridad para modificar la <i>P1 Descripción del Proyecto</i> y/o las expectativas e influencia de algunos involucrados.
RAPE CAE	W1.4 (A1.3) Definir conjuntamente con el Cliente el <i>P5 Protocolo de Entrega</i> de cada uno de los entregables especificados en la <i>P1 Descripción del Proyecto</i> .	
CAE RAPE RDM	<p>W1.5 (A1.11) Generar el <i>P6 Plan de Manejo de Riesgos</i>.</p> <ul style="list-style-type: none"> • Identificar, describir y evaluar los riesgos que pueden afectar el proyecto, contemplar riesgos relacionados con: <ul style="list-style-type: none"> ○ Los involucrados ○ La tecnología y metodología ○ La organización del proyecto ○ Agentes externos al proyecto. • Identificar la probabilidad e impacto de cada riesgo estimando sus implicaciones en los objetivos del proyecto (análisis cuantitativo). • Priorizar los efectos de los riesgos sobre los objetivos del proyecto (análisis cualitativo). • Desarrollar procedimientos para reducir la posibilidad de ocurrencia de los riesgos. • Desarrollar procedimientos para reducir el impacto de los riesgos. 	Los riesgos pueden afectar notoriamente el tiempo y costo de un desarrollo WEB. El control y manejo de riesgo es una disciplina muy amplia y compleja. La tabla planteada como <i>P6 Plan de Manejo de Riesgos</i> es muy sencilla pero suficiente y adecuada para la mayoría de los proyectos.
RAPE	W1.6 Elaborar <i>P7 Plan de Comunicación para el proyecto</i> a partir de los hábitos efectivos de comunicación del equipo de trabajo, la <i>P1 Descripción del Proyecto</i> , el <i>P4 Mapa de Involucrados</i> y el <i>P6 Plan de Manejo de Riesgos</i> . Durante el proyecto analizar la efectividad del plan y ajustarlo en caso necesario.	Debido a las características de los desarrollos Web como el <i>C6 Cambio continuo</i> , los <i>C9 Múltiples involucrados con múltiples áreas de especialización</i> y la <i>C7 Tecnología diversa</i> es necesario elaborar un <i>P7 Plan de Comunicación</i> particular para el proyecto. Un plan de comunicación adecuado considera no solo los canales de comunicación sino además las diversas áreas de especialización, necesidades y antecedentes de los involucrados. El seguimiento y ajuste del plan disminuye drásticamente las <i>D3 Dificultades de comunicación</i> .

Rol	Descripción	Observaciones
RAPE RDM ET*	<p data-bbox="292 427 850 544">W1.7 (A1.10) Generar la <i>P8 Cotización</i> a partir de la <i>P1 Descripción del Proyecto</i>, el <i>P6 Plan de Manejo de Riesgos</i> y <i>P5 Protocolo de Entrega</i>.</p> <ul data-bbox="339 600 850 1099" style="list-style-type: none"> • Revisar la <i>P22 Base de Conocimientos</i> • Establecer supuestos para las variables desconocidas. • Establecer los plazos de entrega condicionados a las fechas de inicio, pagos y entregas de información. • Estimar el costo para generar cada entregable. • Definir plazos de entregas • Estimar el costo de los riegos. • Estimar la ganancia esperada. • Establecer el precio sumando los costos de los entregables, costos de riegos y la ganancia. <p data-bbox="339 1193 850 1368">* En este momento aun no se ha formalmente establecido el equipo de trabajo pero es conveniente evaluar costos y tiempos con quienes se prevé que lo formarán o al menos con personas que tengan habilidades similares.</p>	<p data-bbox="882 427 1479 544">Una de las primeras necesidades del cliente es conocer el tiempo de desarrollo y el precio. La aprobación del proyecto depende en gran medida de estos factores.</p> <p data-bbox="882 600 1479 925">Para cotizar con precisión requieren mayor conocimiento sobre el proyecto del que se puede tener en una etapa tan temprana del proyecto. La solución es cotizar haciendo supuestos sobre todas las variables desconocidas por ejemplo número y complejidad de pantallas, robustez, grado de optimización de descarga, requisitos del servidor, colaboración de los CEI y otros involucrados, etc. Dichas suposiciones deben ser comunicadas y especificadas como <u>condicionantes</u> de la cotización presentada.</p> <p data-bbox="882 981 1479 1126">Una buena opción es generar dos o tres cotizaciones con diversos supuestos y alcances. Es muy importante definir claramente las dimensiones de cada propuesta (aunque sea solo una) y no dejar puntos ambiguos.</p> <p data-bbox="882 1182 1479 1294">Esto cubre la exigencia de una cotización y hace al cliente consciente de los muchos factores que influyen en los costos. Así se facilita lograr acuerdos ante los cambios en los requerimientos.</p> <p data-bbox="882 1350 1479 1585">Para negociar los cambios se debe considerar la relación precio, alcance, tiempo. Los cambios en uno afectan los otros. Si alguno es invariable como el precio las alteraciones en los plazos afectarán el alcance. Si se desea aumentar el alcance deben variar el precio y/o el tiempo y si alguno de estos es invariable el cambio en el otro es mayor.</p> <p data-bbox="882 1641 1479 1933">Hay muchas técnicas para la estimación de costos pero para equipos y proyectos pequeños la mayoría resulta más compleja y costosa que el proyecto en sí. La técnica más viable y recurrida es el juicio de expertos, generalmente los propios desarrolladores. Esta técnica puede ser muy imprecisa, pero se puede mejorar utilizando datos históricos (de la <i>P22 Base de Conocimientos</i>), el análisis de riesgos y contabilizando claramente las pantallas con que contaría cada propuesta.</p>

Rol	Descripción	Observaciones
CAE RAPE	W1.8 Comunicar, negociar, aclarar y ajustar la <u>P8 Cotización</u> apoyándose en la P1 Descripción del Proyecto los P2 Bocetos y <i>listas de sitios de similares</i> y el P3 Glosario , hasta conseguir la aprobación del proyecto.	<p>Aclarar oralmente lo escrito en la cotización y el contrato y en su defecto corregirlos.</p> <p>Es muy importante aclarar todos los términos apoyándose en el <i>P3 Glosario</i> y los <i>P2 Bocetos, Listas de Sitios de Similares</i> y prototipos pues ayudan a disminuir la dificultad para describir las funciones y características de los sistemas WEB y la ambigüedad de conceptos y términos.</p> <p>La forma de aprobación varía según el medio y la relación entre clientes y desarrolladores como por ejemplo contrato convenio VoBo, pero es imprescindible que quede registrada y sea clara para el RAPE, el CAE y el CP.</p>
CAE RAPE ET	<p>W1.9 Dar inicio formal al proyecto.</p> <ul style="list-style-type: none"> • (A1.8) Conformar el <i>Equipo de Trabajo</i>, asignando roles y responsabilidades basándose en la <i>Descripción del Proyecto</i>. • Notificar a clientes y otros involucrados. • Iniciar el <u>P9 Repositorio del Proyecto</u> e incluir los productos generados. 	<p>El <i>P9 Repositorio del Proyecto</i> es crucial para que el equipo trabaje organizadamente. Crear y usar el repositorio del proyecto desde el inicio formal del proyecto y el equipo ayuda a su máximo aprovechamiento y adopción.</p>
RAPE	<p>W1.10 (A1.4) Identificar el número de ciclos y las actividades específicas que deben llevarse a cabo para producir los entregables y sus componentes.</p> <ul style="list-style-type: none"> • Revisar los entregables documentados en la P1 Descripción del Proyecto. • Establecer el <u>P10 Proceso General de Desarrollo</u>. • Generar una primera versión general del <u>P11 WBS</u> para cumplir con el P5 Protocolo de Entrega. • Formar el <u>P14 Plan de Proyecto</u>, integrando el resultado como <i>Ciclos y Actividades</i> con el P6 Plan de Manejo de Riesgos, el P5 Protocolo de Entrega, el <i>Costo Estimado</i> obtenido al hacer la P8 Cotización y el <i>Equipo de trabajo</i>. 	<p>Es importante recordar que este es una primera versión del <i>P11 WBS</i> y los <i>Ciclos y Actividades</i> del <i>P14 Plan de Proyecto</i>. El objetivo es tener un panorama general de las diversas fases y actividades más generales necesarias para desarrollar el sistema. No es viable ni útil tratar de establecer a detalle un plan con todas las tareas del proyecto. El <i>P11 WBS</i> y el <i>P14 Plan de Proyecto</i> son productos que evolucionan a lo largo de todo proyecto.</p> <p>Los primeros ciclos pueden dedicarse elaborar prototipos como los sugeridos en las técnicas <i>T2 Prototipos Grises</i> y <i>T3 Prototipos Operativos</i>.</p>
	A1. Realización de la fase de Inicio de Desarrollo y Mantenimiento de Software.	(Ver sección 6.3)

Rol	Descripción	Observaciones
RAPE RDM	W1.11 (A1.5) A partir del P11 WBS identificar y documentar la relación y dependencia de cada una de las actividades.	Esta actividad resaltara la dependencia del proyecto de algunas actividades que deben realizar los clientes como entregas de información o validaciones de la navegación, diseño gráfico o funcionalidad.
RAPE RDM	W1.12 (A1.6) Establecer el Tiempo Estimado para desarrollar cada actividad.	
RAPE	W1.13 (A1.7) Establecer los <u>P12 Requerimientos de Adquisiciones y Capacitación</u> definiendo las características y el calendario en cuanto a recursos humanos, materiales, equipo y herramientas, incluyendo la capacitación requerida para que el equipo de trabajo pueda desempeñar el proyecto.	Existen una gran cantidad de elementos preprogramados que se pueden encontrar en Internet gratis o por unos pocos dólares ahorrando muchas horas de trabajo. Estos módulos así como la capacitación y otras adquisiciones se deben registrar en el <u>P12 Requerimientos de Adquisiciones y Capacitación</u> .
RAPE	W1.14 (A1.9) Asignar fechas de inicio y fin a cada una de las actividades para generar el <u>P13 Calendario</u> de trabajo tomando en cuenta los recursos asignados, la secuencia y dependencia de las actividades.	
RAPE	W1.15 (A1.12) Actualizar el <u>P14 Plan de Proyecto</u> antes de iniciar un nuevo ciclo.	
RAPE CAE	W1.16 Establecer el <u>P15 Protocolo de Cambio</u> .	El <u>P15 Protocolo de Cambio</u> debe planearse con el cliente. Sin embargo es importante que posteriormente el RAPE establezca que harán los miembros del equipo de trabajo cuando los clientes generen solicitudes de cambio que no sigan el protocolo pactado.
RAPE	W1.17 Capacitar al equipo respecto al P15 Protocolo de Cambio y su reacción para canalizar a otros involucrados cuando no lo sigan.	
RAPE RDM ET	W1.18 (A1.18) Dar inicio formal a un nuevo ciclo.	
A2. Realización Inicio de ciclo		Los primeros ciclos deben utilizarse para afinar aun más el mutuo entendimiento de las necesidades y posibilidades del proyecto por parte de clientes y desarrolladores. Los <u>T2 Prototipos Grises</u> y/o los <u>T3 Prototipos Operativos</u> deben desarrollarse durante estos primeros ciclos de forma que se avance en el desarrollo del proyecto y a su vez se perfeccione la definición de requerimientos.

Rol	Descripción	Observaciones
RAPE RDM	W2.1 (A2.1) Acordar con el Responsable de Desarrollo y Mantenimiento del proyecto la asignación de tareas al Equipo de Trabajo.	
RAPE RDM	W2.2 (A2.2) Acordar la distribución de la información necesaria al equipo de trabajo con base en el P7 Plan de Comunicación .	
	A2. Realización de la fase de Requerimientos de Desarrollo y Mantenimiento de Software.	(Ver sección 6.3)
	A3. Realización de la fase de Análisis y Diseño de Desarrollo y Mantenimiento de Software.	(Ver sección 6.3)
	A4. Realización de la fase de Construcción de Desarrollo y Mantenimiento de Software.	(Ver sección 6.3)
A3. Evaluación y Control		
RAPE RDM	W3.1 (A3.1) Evaluar el cumplimiento del P14 Plan del Proyecto , con respecto al alcance, costo, calendario, equipo de trabajo, proceso y actualizarlo incluyendo Acciones Correctivas.	
RAPE	W3.2 (A3.2) Dar seguimiento y controlar el P6 Plan de Manejo de Riesgos . Identificar nuevos riesgos y actualizar el plan.	
A4. Cierre Fin de ciclo		
	A5. Realización de la fase de Integración y Pruebas	(Ver sección 6.3)
	A6. Realización de la fase de Cierre	(Ver sección 6.3)
RAPE CAE	W4.1 (A4.1) Formalizar la terminación del ciclo o del proyecto de acuerdo a la P5 Descripción del Producto y Entregables y Protocolo de Entrega establecido en el P1 Plan del Proyecto .	
RAPE	W4.2 (A4.4) Integrar productos del proyecto a la P22 Base de Conocimientos .	Al terminar el proyecto además de la entrega y el comunicado apropiado para todos los involucrados. Es importante agregar los diversos productos del proyecto a la P22 Base de Conocimientos del equipo de trabajo o la organización de la que forma parte.

6.3 Actividades del Proceso de Desarrollo y Mantenimiento de software

Rol	Descripción	Observaciones
A1. Realización de la fase de Inicio		
ET	W1.1 (A1.1) Revisar con los miembros del Equipo de Trabajo el P14 Plan de Proyecto actual para lograr un entendimiento común y obtener su compromiso con el proyecto.	
A2. Realización de la fase de Requerimientos		
RAPE RDM ET	W2.1 (A2.1) Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al P14 Plan de Proyecto actual. <ul style="list-style-type: none"> Entregar un P23 Directorio adecuado a sus tareas y lo especificado en el P7 Plan de Comunicación a los miembros del ET. 	Es importante destacar las actividades que dependen de los CEI y/o de miembros de diferentes especialidades dentro del Equipo de Trabajo pues muy factible que estas actividades se retrasen si no se monitorean correctamente. Los CEIs generalmente tienen otras ocupaciones y suelen retrasar las entregas. Se debe dar a los miembros del Equipo de Trabajo responsables de estas tareas un P23 Directorio adecuado; pues deben estar en contacto constante con los CEIs para ayudarlos y motivarlos a hacer sus entregas a tiempo.
AN CAE CEI US DU DN	W2.2 (A2.2) Documentar o modificar la P16 Especificación de Requerimientos . <ul style="list-style-type: none"> Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos. Analizar los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto. Elaborar o modificar el prototipo de la interfaz con el usuario. Generar o actualizar la Especificación de Requerimientos. 	Debido a que en los sistemas web hay D2 Requisitos altamente cambiantes es necesario revisar los requerimientos en cada ciclo. En la mayoría de los proyectos es más conveniente usar la técnica T1 Análisis de requerimientos enfocado a metas que otros métodos de levantamiento de requerimientos. Este método es más eficiente cuando existen muchos involucrados y objetivos muy difusos o generales que son muy frecuentes en los desarrollos WEB.
CAE CEI US	W2.3 (A2.5) Validar la P16 Especificación de Requerimientos . Utilizar P2 Bocetos, listas de sitios de similares y prototipo para mejorar el proceso de comunicación.	Debido a la gran cantidad de cambios es importante validar siempre con el cliente. Esto ayuda a detectar los malentendidos a tiempo, disminuyendo los cambios por este concepto y ayuda a renegociar los plazos, el precio y el alcance, pues el cliente es más consciente de los cambios y sus implicaciones. Utilizar P2 Bocetos, listas de sitios de similares y prototipos es indispensable para disminuir las D3 Dificultades de comunicación .

Rol	Descripción	Observaciones
AN DU DN	W2.4 (A2.6) Corregir los defectos encontrados en la <u>P16 Especificación de Requerimientos</u> con base en el <i>Reporte de Validación</i> y obtener la aprobación de las correcciones.	
RPU AN	W2.5 (A2.7) Elaborar o modificar <u>P17 Plan de Pruebas</u> .	Por la característica <i>C7 Tecnología diversa</i> al generar el <i>P17 Plan de Pruebas</i> se debe poner especial atención en probar el sistema en navegadores que se espera que tendrán los usuarios objetivo. Generalmente es recomendable probar la funcionalidad en al menos los dos navegadores más populares. Pero esto puede variar según el grupo de usuarios objetivo.
A3. Realización de la fase de Análisis y Diseño		
RAPE RDM	W3.1 (A3.1) Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al <u>P14 Plan de Proyecto</u> actual.	
ET	W3.2 Probar y elegir las tecnologías a utilizar.	Es necesario probar que la combinación de lenguajes a utilizar funciona bien en conjunto. Probar especialmente los lenguajes y técnicas a utilizar en al menos los dos navegadores más populares (y/o aquellos que el proyecto requiera) pues es muy común que estos tengan comportamientos diferentes ante un mismo código.
AN DI DU DN	W3.3 (A3.2) Documentar o modificar el <u>P18 Análisis y Diseño</u> : <ul style="list-style-type: none"> • Analizar la <u>P16 Especificación de Requerimientos</u> para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y éstos a su vez en componentes, definiendo las interfaces entre ellos. • Describir el detalle de la apariencia y el comportamiento de la interfaz con base en la <u>P16 Especificación de Requerimientos</u> de forma que se puedan prever los recursos para su implementación. • Observar los requisitos de información del sistema de forma que se pueda diseñar una estructura de navegación simple y efectiva y planear la generación y manutención de la información del sistema • Describir el detalle de los componentes que permita su construcción de manera evidente. • Generar o actualizar el <u>P18 Análisis y Diseño</u>. 	<p>Un sistema WEB se debe diseñar en tres aspectos la funcionalidad del software, que es prácticamente idéntico a los sistema tradicionales, la navegación que es propia de los sistemas WEB y la interfaz que aunque existe en los sistemas tradicionales en los sistemas WEB resulta más compleja e importante como se explica en la característica <i>C3 Estética y diseño grafico preponderantes</i>.</p> <p>En estos tres aspectos es indispensable aplicar la técnica <i>T5 Uso de patrones</i>. A partir de lo observado al generar <i>P2 Bocetos, Listas de Sitios Similares y Prototipos</i> y la propia <i>P16 Especificación de Requerimientos</i> se deben seleccionar, mejorar e integrar los patrones para el sistema guiándose en los objetivos del sistema y lo señalado en las <i>T6 Técnicas para el diseño de arquitecturas</i>, las <i>T7 Técnicas de Diseño gráfico</i>, las <i>T8 Técnicas para el diseño de navegación</i> y las <i>T9 Técnicas para presentar la información</i>.</p>

Rol	Descripción	Observaciones
A4. Realización de la fase de Construcción		
RDM	W4.1 (A4.1) Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al P14 Plan de Proyecto actual.	
PR	W4.2 (A4.2) Construir o modificar el(los) <i>Componente(s)</i> de P19 Software : <ul style="list-style-type: none"> Implementar o modificar <i>Componente(s)</i> con base a la parte detallada del P18 Análisis y Diseño. Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada del <i>Análisis y Diseño</i>. 	La actividad de programación en sí de los sistemas Web no varía respecto a los sistemas tradicionales.
A5. Realización de la fase de Integración y Pruebas		
RDM	W5.1 (A5.1) Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al P14 Plan de Proyecto actual.	
PR RPU RE	W5.2 (A5.2) Realizar integración y pruebas. <ul style="list-style-type: none"> Integrar los componentes en subsistemas o en el sistema de P19 Software y aplicar las pruebas siguiendo el P17 Plan de Pruebas Corregir los defectos encontrados o programarlas para el siguiente ciclo. Generar el P20 Registro de errores 	<p>Por la salvedad de que es recomendable probar con más de un navegador. Estas pruebas se pueden ejecutar prácticamente igual que en una aplicación tradicional. Es necesario utilizar el P20 Registro de Errores y Cambios y se recomienda usar herramientas específicas para este fin.</p> <p>Como se indica en T10 Técnicas para probar las aplicaciones WEB es válido apoyarse en los propios clientes y usuarios para lograr una mayor detección de fallos. Si el proyecto lo permite es aconsejable pactar un periodo de pruebas en el que el equipo de desarrollo corrija los errores que encuentren los propios usuarios.</p>
A6. Realización de la fase de Cierre		
RM	W6.1 (A6.1) Documentar el P21 Manual de Mantenimiento .	

7.- Técnicas para desarrollo WEB

Si bien el proceso general permanece prácticamente inalterado en cuanto a sus fases requerimientos, diseño, construcción y pruebas. En el área de técnicas y/o metodologías específicas para alguna de las etapas encontramos varias propuestas.

T1 Análisis de requerimientos enfocado a objetivos

El desarrollo WEB involucra información, navegabilidad, estética y personas de diversas disciplinas además de la funcionalidad fundamental en todo sistema de software. En [7] se propone el levantamiento de requerimientos enfocado a objetivos. Consiste en términos generales en listar a los diversos involucrados y sus objetivos. Posteriormente descomponer dichos objetivos en sub-objetivos que si son complementarios se marcan con –y- (AND) y si son alternativos se marcan con – ó - (OR). Después se marcan las relaciones entre los diversos árboles y/o ramas con líneas punteadas. Los objetivos contradictorios se marcan con una línea cruzada con una X. “Las raíces de los árboles son los objetivos mientras que las hojas son los requerimientos” [7, pag:6]. Las hojas o requerimientos se pueden catalogar después como:

- Content (labeled with C); -Contenido-
- Structure of Content (SC); -Estructura del contenido-
- Access Paths to Content (A); -Ruta a contenido-
- Navigation (N); -Navegación-
- Presentation (P); -Presentación-
- User Operation (U); -Operación del usuario-
- System Operation (O); -Operación del sistema-
- Interaction (I). -Interacción-

El árbol resultante del ejemplo utilizado en [7] se muestra en la figura 2-4.

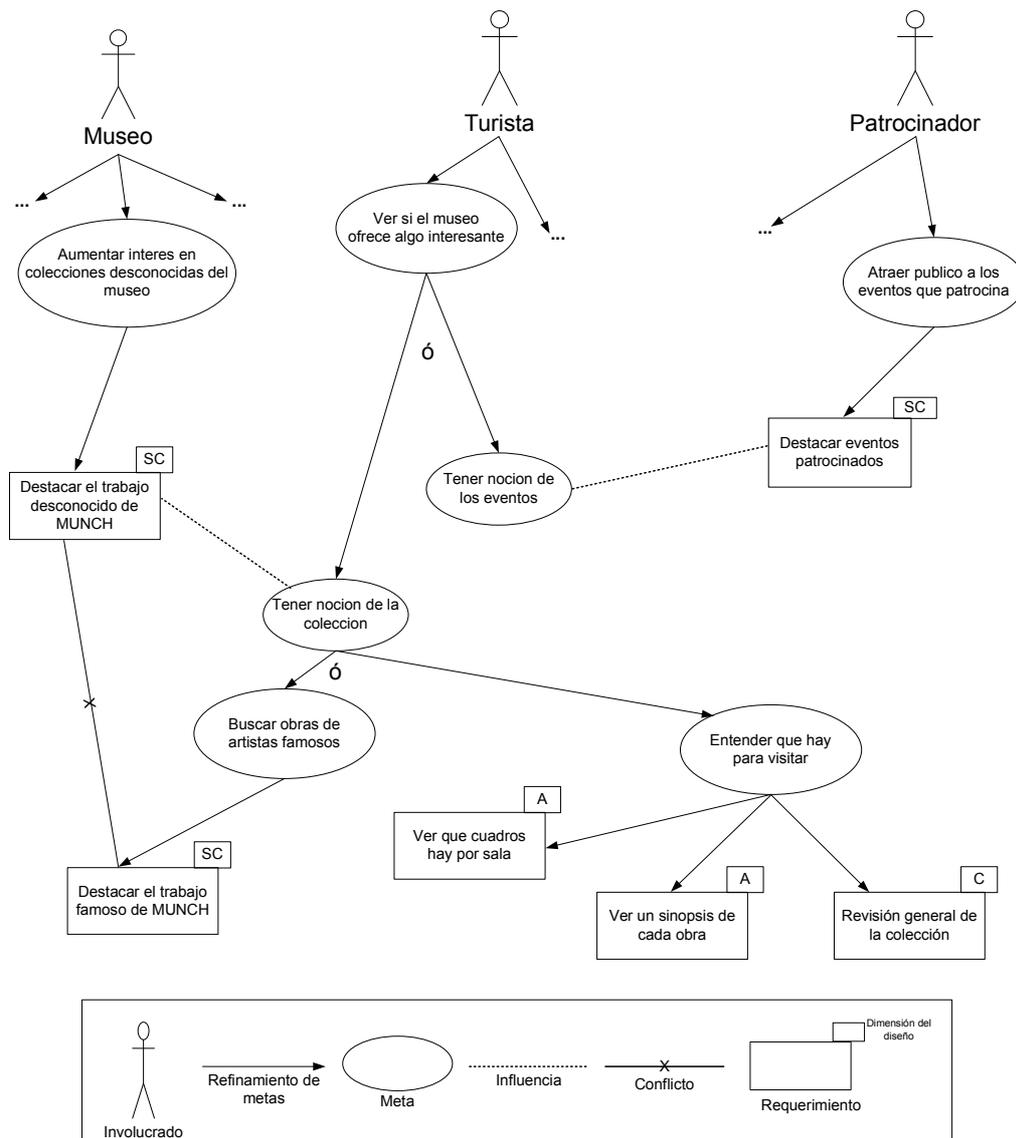


Figura 2-4: Ejemplo de árbol de requerimientos por objetivos, traducción de árbol en [7].

En este ejemplo se muestra que entre otros objetivos el museo desea aumentar el interés en colecciones poco conocidas de su acervo. El turista desea ver que hay para ver de su interés en el museo y los patrocinadores desean atraer público a los eventos que patrocina. Después se muestra la descomposición de estos objetivos hasta llegar a los requerimientos puntuales como Ver que cuadros hay por sala, hacer una revisión general de la colección, destacar el trabajo de E. Munch.

Los requerimientos resultantes se pueden listar y priorizar para resolver los que sean posible de acuerdo al alcance del proyecto. Los conflictos se resuelven haciendo propuestas y discutiendo con los involucrados.

Este procedimiento es inicialmente poco específico. Esto es una ventaja, en los desarrollos WEB, pues no obliga a hacer decisiones de diseño cuando aún no se tienen todos los elementos necesarios. Por su

propia definición y proceso, el levantamiento de requerimientos enfocado a objetivos también obliga a considerar a todos los involucrados en el sistema, por ejemplo: El que encarga el sistema, quienes lo usaran tanto del grupo (u organización) de clientes como fuera de este, quienes lo administraran, quienes generar la información inicial y quienes al actualizaran. Adicionalmente una vez terminado este tipo de análisis proporciona datos tanto de los requerimientos funcionales como de la información e inclusive varios requerimientos no funcionales. Es importante tener en cuenta que esta técnica permite resolver de mejor manera las dificultades que causan usuarios con objetivos difusos o muy generales. Millones de cibernautas acceden a sitios sin más objetivo que ver que encuentran. En los sistemas tradicionales este tipo de usuario es muy improbable pero en los sistemas WEB es más bien la norma.

T2 Prototipos Grises

La comunicación no es solo una de las principales funciones de un sistema WEB, es también uno de los principales problemas durante el su desarrollo. En [6] Eric Holter describe la metodología de trabajo usada en su empresa Newfangled WEB Factory. Bajo la premisa de evitar la ilusión de comunicación se plantea un esquema de trabajo cuyo eje conductor es la elaboración de prototipos denominados grises.

Los prototipos grises son páginas con ligas funcionales y contenidos pero que se enfocan únicamente en la información que cada página o sección debe contener. El calificativo gris, color que la mayoría de las personas no encuentra atractivo, es intencional y se da por la ausencia de todo diseño gráfico. Estos prototipos permiten que clientes y desarrolladores se enfoquen en los contenidos sin distractores. El trabajo y los avances se hacen tangibles y evaluables inmediatamente en los prototipos y evitan ambigüedad y la ilusión de comunicación que pueden dar los términos y descripciones abstractas utilizados cuando no se hacen prototipos. Son también instrumentos fáciles de hacer y que sirven de base para el producto final.

T3 Prototipos operativos

Los e-prototipos o prototipos operativos son un nuevo paradigma que aborda dos de los principales problemas de los desarrollos WEB, el contacto con el cliente y el cambio constante. Son descritos en [5] y básicamente consisten en elaborar prototipos funcionales que se pongan en operación para que los usuarios reales puedan probarlos y dar sus opiniones. En el software tradicional también se llegan a usar algunas versiones beta que son distribuidas a usuarios selectos, con este fin. Pero la naturaleza técnica de estos sistemas hace difícil la implementación de este esquema como principal estrategia de desarrollo. En la WEB por el contrario es fácil y natural; se puede consultar desde la misma aplicación con los que serán los usuarios reales y los cambios se pueden implementar fácil y constantemente. La bidireccionalidad propia del medio también hace extremadamente sencilla la retroalimentación. Los prototipos generalmente pueden ser reaprovechados parcial o totalmente. Los usuarios no sugieren o expresan necesidades supuestas sino reales, fruto de su propia experiencia con el producto.

El desarrollo por e-prototipos se basa en un paradigma de evolución constante del software en ciclos cortos. Esto se ajusta a las demandas del mercado WEB de cambio constante. Además establece comunicación directa con los usuarios, que generalmente no se pueden consultar en otros modelos de desarrollo WEB. Los prototipos operativos son ya ampliamente utilizados por diversas compañías de gran éxito en Internet. Los que podemos identificar son aquellos sistemas con la leyenda BETA como en su momento fueron por ejemplo G-Mail o Yahoo-Groups. Pero puede haber muchos otros sistemas que omitan este calificativo y sin embargo operen bajo el esquema de recopilación de requerimientos directos de los usuarios finales por medio de correos, foros, etc.

T4 Técnicas para modelación

Aun no existe un estándar para modelar los sistemas WEB. Existen muchas propuestas como son OOWS[2], WebML, UWE, entre otros. Estos lenguajes no son idénticos pero en general todos siguen la misma estrategia y paradigmas. Toman como base UML y lo extienden para poder modelar también los contenidos y la navegación.

El uso de diagramas se ha vuelto indispensable al documentar los sistemas. La versatilidad, expresividad y simplicificación que implican, ha llevado incluso al planteamiento de una nueva forma de desarrollo de sistemas conocida como MDA o Model Driven Architecture por sus siglas en inglés. En el CDROM anexo que contiene una versión de la guía en HTML se pueden encontrar un sitio modelado en OOWS como un ejemplo anexo al producto *P2 Bocetos, Listas de Sitios Similares y Prototipos* puedes encontrar diagramas OOWS una herramienta de MDA con la que se modelo casi en su totalidad (excluyendo parte del el diseño gráfico que se especifica en CSS) el sitio: <http://arqsostenible.upv.es/>. OOWS es uno de los nuevos metalenguajes de desarrollo basado en modelos aunque aun es muy limitado y está en etapa experimental pero que ya permite la generación de sitios a partir únicamente de modelos como se muestra en [2]. El desarrollo WEB 100% basado en modelos es aún muy limitado y carece de herramientas profesionales y estables. Sin embargo es un área con mucho potencial y es conveniente mantenerse atento a los desarrollos de la misma.

T5 Uso de patrones

En la sección de arquitecturas de [1] se puede ver que en los sistemas WEB se pueden usar la mayoría de los patrones del diseño de software tradicionales. Sin embargo si se pone atención a la búsqueda de patrones al navegar la propia Internet y se consultan sitios dedicados al tema como [19] y [20] veremos que el uso de patrones no se limita a éstos. Los usuarios de sistemas WEB están poco dispuestos a leer indicaciones y manuales. Por esto los sitios deben ser intuitivos y fáciles de usar para sus usuarios objetivo. Los patrones para sitios WEB (son diferentes de los diseño) son una gran herramienta para lograr la simplicidad y familiaridad de los usuarios con el sistema. El uso de elementos comunes en muchos sitios facilita a los usuarios el uso de los mismos. Así podemos encontrar patrones casi universales en los sitios WEB como son la barra de navegación, las tablas o los contenidos indexados. Hay también patrones para sistemas o componentes con funciones específicas, por ejemplo los buscadores básicamente constan de un campo de texto y un botón para iniciar la búsqueda; en el caso de los sistemas que venden productos utilizan alguna variante de los “carritos de compras”. Otro caso son patrones para mostrar cierto tipo de información como las ligas o secciones tituladas “Inicio”, “Acerca de” o “Contacto”. Otros patrones se encargan de la presentación de la información así podemos encontrar contenidos indexados, tablas, pestañas, paginación, mosaicos de imágenes, etc. La taxonomía y clasificación de los diversos patrones WEB aun no está unificada sin embargo esto no impide su uso. La observación de los sitios WEB similares resulta crítica en este sentido.

T6 Técnicas para el diseño de arquitecturas

En diversas aéreas como el diseño gráfico y funcional las variaciones entre los patrones utilizados en los sistemas WEB y los tradicionales son notorias. Sin embargo en el área correspondiente a la arquitectura y la programación en sí, estas diferencias son más sutiles y en general en los sistemas WEB se utiliza un subconjunto de los muchos patrones de arquitectura que se pueden utilizar en los sistemas tradicionales. Esto se debe a que las arquitecturas WEB deben poder funcionar en el esquema cliente/servidor y es muy deseable que permitan separar las funciones de la información.

En la sección de arquitecturas de [3] se describen muchas posibles arquitecturas de los sistemas WEB como modelo de N capas, JSP Modelo-2, Struts o OOHDM-Java2 en su mayoría son adaptaciones de la arquitectura de los sistemas de software tradicional. Las bases de la mayoría son la arquitectura

por capas y la de modelo, vista, controlador. La primera se plantea generalmente con una capa de interfaz con la que interactúa el usuario, una o varias capas de procesamiento de información una última capa persistente donde se almacenan los datos. Los datos y comandos en este modelo son propagados capa por capa y cada capa puede interactuar directamente sólo con las capas superior y/o inferior. La segunda arquitectura presenta un módulo de vista, uno de procesamiento de datos o control y uno de almacenamiento persistente. En este caso los tres módulos pueden interactuar indistintamente con los otros dos. Estas arquitecturas se utilizan porque tienden a aislar y facilitar el manejo de la información que como se mencionó en las características *C1 Comunicación*, *C2 Interacción* son preponderantes en los sistemas WEB.

Aun cuando los paradigmas y estructura básica de estas arquitecturas no varían es importante tener en cuenta la influencia de la información en los sistemas WEB. Las arquitecturas de N capas y las basadas en modelo-vista-controlador permiten un manejo separado de los datos y la información del sistema y en este punto convergen también con las nuevas tecnologías de lenguajes de marcado como XML, por esto son de las más utilizadas en los sistemas WEB.

T7 Técnicas de Diseño gráfico

Como se explica en *C3 Estética y diseño gráfico preponderantes* son muy importantes y tienen un rol fundamental en retener a los navegantes en los primeros instantes en que visitan un sistema WEB. Sin embargo, resultan inservibles si el sistema no resulta útil al usuario. El diseño gráfico debe entonces procurar una presentación clara de la información en forma estética. En sistemas WEB el mejor lenguaje para manejar estos aspectos son las hojas de estilos CSS que, con un poco de práctica, permiten el manejo y manipulación de los siguientes elementos mencionado en [18] y [3 cap:11]:

Tipografía (tipo, tamaño, color y decorado de las letras): En este caso se recomienda utilizar tipografía más grande y clara que en medios impresos, pues leer en la pantalla es incomodo. El tamaño y la decoración se deben utilizar para destacar la información más importante.

Agrupación: La información y elementos visuales deben agruparse para facilitar su entendimiento al lector. Las categorías pueden ser por función, tipo de dato, formato de presentación, etc. Los diferentes grupos pueden hacerse evidentes por proximidad de sus elementos e incluso el uso de elementos agrupadores como márgenes, líneas, recuadros e imágenes. También se puede usar el color o la tipografía.

Orden: El orden de lectura de arriba abajo y de izquierda a derecha o el que venga al caso según la lengua de los usuarios objetivo. Debe considerarse al momento de ubicar los contenidos dentro de la pantalla pues determina en buena medida en que zona de la misma pondrá su atención en primera instancia el lector.

Imagen y color: El uso de colores e imágenes es una de las formas más sencillas de hacer visualmente atractivo un sistema WEB. El poder de estas herramientas es tal que muchas veces se cae en el abuso. El primer punto es evitar que la combinación de colores dificulte la lectura el color de las letras debe tener un alto contraste con el color que se use de fondo. Si se usan imágenes como fondo es muy recomendable utilizarlas como sello de agua con los colores muy atenuados. Las imágenes deben usarse como elementos que sean a la vez informativos y decorativos. Iconos, delimitadores de conjuntos de elementos, botones, títulos o encabezados son excelentes opciones para utilizar imágenes que aporten tanto al fácil entendimiento del sistema, como a la estética del mismo. La selección cuidadosa del color las combinaciones de colores son un tema delicado no solo pueden hacer fácil o difícil la lectura alteran drásticamente la percepción de los usuarios. Una de las mejores herramientas para hacer dicha selección es el

círculo de color. El círculo de color es un acomodo de los colores en una circunferencia que utilizando simples separaciones de diversos grados como 180 o 120 permite encontrar los colores que combinan con aquel que seleccionamos. En la propia Internet se pueden encontrar gran variedad de versiones de esta herramienta y lograr fácilmente buenas combinaciones de colores.

Es importante además tener constancia y congruencia en todo el sistema. Mantener procesos, criterios de clasificación y formas de presentación similares para el mismo tipo funciones, datos y/o elementos a lo largo de todo el sistema para evitar que los usuarios deban aprender e interpretar en cada pantalla.

T8 Técnicas para el diseño de navegación

Un factor único de los sistemas WEB es la navegación, la información y su presentación no pueden ser usados ni apreciados si no se encuentran. El acceso a la información debe también diseñarse. En [2] y [3] se señalan lineamientos generales que es recomendable seguir en la mayoría de los sistemas WEB y estos son:

1. **Siempre informar al usuario donde se encuentra:** La versatilidad de ir de un sitio WEB a otro y de una sección a otra con un solo clic, puede causar rápidamente desorientación en el usuario. Ya sea por ligas poco claras o distracciones ajenas al sistema el usuario puede perder noción de lo que está viendo. Por eso casi 100% de los sitios utiliza un encabezado presente en todas sus pantallas que indica de que sitio se trata. Además es recomendable indicar exactamente en qué pantalla del sistema está el usuario mostrando un título en cada una. En sistemas medianos y grandes también se debe poner la ruta o clasificación de la pantalla dentro del sistema.
2. **Tres clics de distancia:** Los contenidos deben ser accesibles. La regla de los tres clics sugiere hacer una estructura del sitio muy horizontal en donde no se necesiten más de tres ligas o pasos para llegar a cualquier contenido. Si se deben dar más pasos dificultan que se encuentren los contenidos en primera instancia y que los usuarios logren recordar cómo encontrarlos en visitas posteriores.
3. **Ligas externos:** Un sistema no está aislado, se deben generar ligas desde y hacia otros sitios relacionados. Esto facilitará que los usuarios encuentren la información y que se aprovechen desentenderte de tareas e información útil para los objetivos del sistema pero que están más allá de su alcance. Al apoyarse en otros sistemas los contenidos y funciones pueden mantenerse simples y actualizados facilitando entre otras cosas cumplir con la regla de los tres clics.

T9 Técnicas para presentar la información

Como se constata en [21] la información en Internet puede ser video, imagen, audio o texto pero en todos los casos estos elementos deben adaptarse al medio. La *C7 Tecnología diversa*, los *C8 Usuarios simultáneos y diversos* y limitaciones como la velocidad de transmisión o el tamaño de las pantallas, influyen sobre la efectividad de estas formas de comunicación. En todos los casos se debe buscar un equilibrio de dos factores contrapuestos al presentar información clara y completa pero a la vez breve y concisa.

En el marco del *C6 Cambio continuo* que impone Internet la información es el aspecto que más cambia de todos. La generación de información clara y completa pero a la vez breve y concisa es muy costosa y tardada por lo tanto es muy importante distribuir esta tarea entre diversos miembros de la

organización cliente e involucrar a los propios usuarios de sistema, lo más posible, en las tareas de generación y actualización de la información.

En el caso de los textos deben ser breves, generalmente es recomendable elaborar resúmenes de la información que manejan diariamente los clientes o usuarios en otros medios. En los casos en que los objetivos del sistema incluyan justamente el acceso a grandes volúmenes de información lo más recomendable es presentarla en formatos descargables para su consulta fuera de línea y/o diseñados para ser impresos. Los datos en texto ofrecen una gran variedad de patrones de presentación como son la tabulación, el paginado, el indexado, el listado, etc.

En el caso de las imágenes se deben adaptar la resolución y el formato para que se vean bien en pantalla pero ocupen el menor espacio posible.

El audio y el video deben también codificarse en formatos de alta compresión y a baja resolución para que puedan ser vistos y/o escuchados aun en enlaces que presenten baja velocidad. En el caso de video es recomendable generar video donde el fondo cambie poco y/o cambie lentamente.

T10 Técnicas para probar las aplicaciones WEB

En la sección de pruebas en [3] se puede apreciar que las pruebas de las aplicaciones WEB son similares a las de aplicaciones tradicionales pero presentan dos diferencias fundamentales. Primero debido a la *C7 Tecnología diversa* son necesarias muchas pruebas para probar la misma funcionalidad en diversas plataformas. Segundo, gracias a la facilidad con que se puede actualizar el sistema y a que se cuenta con *C8 Usuarios simultáneos y diversos* estos se pueden involucrar más fácilmente en las pruebas de la aplicación, como se sugiere en [5]. Dentro del espectro de pruebas que puede desarrollar un equipo pequeño se recomienda lo siguiente:

- Probar la combinación de las tecnologías a utilizar en varios navegadores antes de incorporarlas al desarrollo.
- Seguir un procedimiento de pruebas tradicional para probar la funcionalidad de la aplicación.
- Utilizar la retroalimentación de clientes y usuarios para mejorar la detección de errores y reducir costo y tiempo de las pruebas. Un extremo de este enfoque apoyándose en los propios usuarios es la técnica de *T3 Prototipos operativos*.

8.- Productos

Es importante destacar que se habla de productos y no de documentos porque no se generan forzosamente documentos de texto convencional. El tipo y profundidad de cada producto debe adaptarse a cada equipo de trabajo y proyecto. Documentar y organizar es sin duda útil pero también costoso; no tiene sentido realizar un proyecto donde la documentación en si cueste más que el valor que aportará el proyecto. Normalmente entre más involucrados, tiempo y esfuerzo requiera el proyecto más formal debe de ser la documentación pero intervienen muchos otros factores.

En un trabajo muy sencillo, para un amigo cuya necesidad se conoce bien, puede ser suficiente (pero poco recomendable) simplemente platicar y establecer oralmente qué se requiere, escribir un breve listado, desarrollar el sistema y entregarlo. El mismo trabajo para un desconocido probablemente requiera no solo mas dialogo sino esquemas y una lista completa de las secciones y prestaciones del sistema. Algo muy similar para una empresa requiere, además de las secciones y prestaciones del sistema, especificar tráfico esperado, tiempo de garantía, etc.

Lo mismo ocurre con otros productos si hay mas involucrados o el proyecto es más complejo. Si el equipo de trabajo prefiere XP (eXtreme Programming) o RUP (Rational Unified Process) el plan de desarrollo y otros productos son, en su forma, muy diferentes pero en su fondo y objetivos similares.

Lo importante es tener en mente que los productos son útiles, deben adaptarse según las circunstancias para optimizar el trabajo presente y futuro del equipo así como la calidad del sistema. Los productos son unidades de información que se deben generar y/o conseguir y organizar. Se pueden usar diversos medios y técnicas, siempre que la información se pueda plasmar, consultar y comunicar cuando sea necesario. Los productos tampoco implican que se requiere un documento o elemento por cada uno, solo que es necesario tener esa información. En el propio MoProSoft se puede ver que hay productos conformados por otros productos. Por ejemplo es posible que en un proyecto el contrato especifique los entregables y las fechas de entrega mientras que en otro puede no haber un contrato formal y la aceptación y los entregables estén en documentos separados.

La principal función de los productos es registrar y comunicar. Los productos deben ayudar al trabajo aportando claridad, testimonio y una referencia común entre los diferentes involucrados y los miembros del equipo de desarrollo. Hay que tener en cuenta que el alcance de los productos debe ir más allá del proyecto en cuestión y servir como base para facilitar futuros proyectos. Es necesario encontrar un punto medio, demasiada documentación y detalle volverá su consulta más compleja que el volver a enfrentar los problemas en sí. Por otra parte productos breves e incompletos generan problemas durante el proyecto y resultaran inútiles en el futuro.

Los productos evolucionan con el proyecto, no hay que pretender hacer productos completos y perfectos en cuanto son mencionados en las actividades. Hay ciclos de desarrollo y además se debe trabajar simultánea y continuamente en diversos productos que dependen unos de otros. A continuación listamos los productos en el orden en que cronológicamente se inician.

P1 Descripción del Proyecto

Descripción: Este documento explica qué se necesita, para qué se necesita y qué grado de automatización, perfeccionamiento y calidad se requiere. Está formado por las siguientes secciones:

- Descripción del Producto, que explica en términos generales cómo será el sistema.
- Objetivos, aclara la finalidad del sistema.
- Entregables, es el listado de todos los productos y subproductos del proyecto que serán entregados al cliente.
- Alcance, especifica el nivel de detalle y funcionalidad de cada entregable y del sistema en su conjunto.

Utilidad: Este producto es el eje conductor del proyecto; indica que se hará y para qué. Es indispensable para cotizar, planear y diseñar correctamente; simplifica la comunicación y ayuda a unir al equipo y todos los involucrados.

Observaciones:

Los componentes de la *Descripción del Proyecto* suelen estar dispersos en diversos documentos y medios y por lo tanto es fácil que la descripción esté incompleta. Generalmente los clientes proveen

uno o más componentes al solicitar el proyecto pero no todos. Se debe cerciorar su recopilación, documentación y validación con los clientes y/o el responsable de gestión de proyectos. Todo el equipo de trabajo debe conocer al detalle la *Descripción del Proyecto* pues solo así podrán trabajar, evaluar y decidir correctamente. En general en toda interacción con cualquier involucrado es recomendable recordar lo plasmado en el Descripción del Proyecto, pues de lo contrario cada involucrado genera expectativas propias sobre el sistema. Por ejemplo: Se está desarrollando un sistema de administración para Empresa ejemplo S.A.de C.V. donde los objetivos y el alcance no están claramente especificados. Es muy probable que los clientes del área de contabilidad pidan un sistema que reciba las facturas escaneadas, las reconozca y las procese automáticamente, los del almacén pidan que el sistema haga pedidos automáticamente a los proveedores, y los programadores deseen probar un complejo sistema de encriptación para evitar que se levanten pedidos o facturas falsos. En el mismo escenario, pero con el objetivo específico de mostrar al dueño en todo momento el estado de la empresa y definiendo que el alcance de una primera etapa es únicamente reportar el estado de la empresa sin intervenir directamente en ninguna área, ninguno de los involucrados hubiese generado esas expectativas ni todas las horas hombre de discusión, estimación y validación que implican.

En sistemas WEB es posible que incluso algunos componentes de la descripción del proyecto cambien, conforme clientes y desarrolladores conocen más del proyecto. El impacto de dichos cambios suele ser más complejo y costoso de lo que los clientes imaginan y si no se documentan y validan todos estos es probable que los cambios no sean percibidos como tales y menos aun que los clientes entiendan los retrasos y costos extra que ocasionan.

El primer punto para desarrollar la descripción del proyecto es revisar la información que se tenga hasta el momento. Si existe un contrato es muy probable que este ya contenga todo o casi todos los apartados. Si en su organización hay un responsable de gestión de proyectos éste debería haber generado una *Descripción del Proyecto* adecuada. Los correos electrónicos y/o las conversaciones con el cliente patrocinador (CP) y/o administrador encargado (CAE) pueden proveer toda la información faltante. Una vez que se tenga toda la información es indispensable documentarla y sobre todo corroborarla y comunicarla con el CAE aun cuando se haya basado en información que él mismo haya proporcionado. Suponer que todos ya conocen la *Descripción del Proyecto* es uno de los errores más costoso y comunes. Es imprescindible que tanto el CAE como el responsable de la administración de proyecto específico RAPE tengan constancia documental de todos los componentes de la descripción del proyecto. Es importante además verificar que cada componente se entiende y el CAE y el RAPE están de acuerdo en su contenido independientemente de en qué documentos se encuentren. Así mismo el equipo de desarrollo y casi todos los otros involucrados deben ser informados de los objetivos y alcances del sistema.

P2 Bocetos, Listas de Sitios Similares y Prototipos

Descripción: Referencia a sitios similares o con características similares a las deseadas en el sistema. Dibujos sencillos que describen alguna idea o patrón y sistemas con funcionalidad parcial o enfocados solo a algún aspecto del sistema.

Utilidad: Ayudan a expresar o describir conceptos, funciones y características difíciles de describir con palabras y/o diagramas UML. Aumenta notoriamente la efectividad de la comunicación casi sin esfuerzo adicional.

Observaciones:

Muchas de las solicitudes de cambios en los desarrollos WEB ocurren hasta que los clientes ven el sistema o algún módulo operando. Los prototipos, modelos y ejemplos ayudan a prevenir este

problema detectando los cambios necesarios en etapas más tempranas. Pues permiten visualizar diversos aspectos del sistema antes de construirlo completamente. Los prototipos y diagramas pueden abarcar todos los aspectos del sistema pero son mucho más efectivos al enfocarlos en solo uno de estos, como la navegación, la interfaz, el contenido, la funcionalidad, etc. Por ejemplo para determinar qué pantallas habrá y qué información se presentará en cada una puede construirse un prototipo gris, que es un sitio sin ningún elemento de diseño gráfico pero con la información que llevara cada pantalla. Esto ayuda a que el cliente y los diversos evaluadores se enfoquen en los contenidos sin distraerse por el diseño. De la misma forma se pueden desarrollar diversas alternativas gráficas en programas de creación de imágenes para determinar el estilo visual del sitio sin que los contenidos específicos de una pantalla o su funcionalidad afecten el juicio de los elementos estéticos. La navegación es generalmente más fácil de entender modelándola con diagramas como los que para este propósito ofrecen prácticamente todos los métodos de modelado WEB como OOWS, WebML, WebRatio, OOHDM, etc. Finalmente también es recomendable extender la lista de sitios similares o relacionados con referencias más explícitas de qué funcionalidades se desean emular y con qué cambios e incluso qué se desea evitar. No solo no es necesario inventar el hilo negro tampoco es práctico describirlo.

El trabajo con prototipos funcionales o versiones simples y reducidas del sistema puede resultar útil también como parte del desarrollo en sí. Esta aproximación está siendo ampliamente utilizada en Internet, es por esto que a menudo nos encontraremos con servicios denominados BETA. Es especialmente útil cuando se desea contemplar la opinión de los usuarios finales que en muchos casos no se conocen ni se pueden contactar personalmente hasta que el sistema está operando. En este esquema se ponen en operación versiones incompletas del sistema pero con varias de sus funcionalidades ya operativas y los propios usuarios aportan retroalimentación tanto sobre los requerimientos y características del sistema como acerca de sus errores. Este mecanismo puede resultar mucho más efectivo rápido y económico que las pruebas tradicionales en cierto tipo de sistemas.

P3 Glosario

Descripción: Definición en el contexto del proyecto de términos clave y que puedan resultar ambiguos, desconocidos y/o tengan significados específicos dentro proyecto.

Utilidad: Evita la confusión, mejora y agiliza la comunicación.

Observaciones:

El glosario explicara en términos del propio proyecto palabras que se usen normalmente en este. Es una herramienta muy sencilla y útil que es suele ser muy poco apreciada. Tener claridad en los términos puede ahorrar muchas horas de discusión y trabajo. Los sistemas WEB involucran a personas con una gran variedad de especialidades y su vocabulario puede resultar ambiguo. Por ejemplo en un sistema para una escuela para el personal de la escuela la palabra clase se refiere a un grupo de alumnos con un profesor en un salón cubriendo los materiales de un curso X en un horario Y. Para el diseñador de la interfaz se trata de una definición de estilos CSS. Los programadores pueden interpretar dicha palabra tanto como un registro de la tabla clases como o como un elemento de conceptual de la programación orientada a objetos. Esto puede ocurrir con cualquier palabra y en ocasiones su efecto se circunscribe únicamente a un grupo de personas específico. Por ejemplo en una oficina de recursos humanos un puesto y una plaza pueden ser usados indistintamente y mientras que en otra (aun dentro de una misma organización) una plaza puede ser un puesto ya ocupado por empleado. Por esto es indispensable aclarar y definir los términos en cada proyecto y en ocasiones con cada uno de los involucrados por parte del cliente, pues incluso dentro de una misma organización pueden dar un uso diferente a algunos términos.

El glosario es una herramienta para mejorar la comunicación y esta se da desde el inicio del proyecto al establecerse los requerimientos iniciales. El RAPE y el CAE y/o el CP deben clarificar diversos términos para poder elaborar los requerimientos de forma efectiva y sin tener conflictos posteriores. Es desde este momento que el glosario se debe utilizar.

El glosario no debe hacerse con la expectativa de que será disciplinadamente leído por quienes no conocen algún término. El glosario no es un diccionario y sus términos no son exclusivamente palabras que algunos involucrados desconocen. Son además palabras que resultan ambiguas dentro del proyecto por la gran variedad de disciplinas, áreas e involucrados que abarca un sistema WEB. La gran mayoría de los involucrados asumirá que estas palabras significan para todos lo mismo que para él y no sentirá la necesidad de consultar el glosario.

El glosario sirve para recordar a quien ya es consciente de esas ambigüedades y usos particulares qué sentido tiene en particular dentro del proyecto. En todo proceso de comunicación quienes transfieren la información deben detenerse al usar dichos términos y verificar con su interlocutor que entienden lo mismo al usarlos apoyándose en el glosario. En otras palabras el glosario sirve cuando se usa activamente en las conversaciones.

P4 Mapa de Involucrados

Descripción: Plano que ubica en uno de cuatro cuadrantes a todos y cada uno de los involucrados, documentando su actitud, interés e influencia ante el proyecto.

Utilidad: Documentar la actitud, interés e influencia ante el proyecto de cada involucrado.

Observaciones:

El mapa de involucrado ubica en dos ejes, poder e interés, a los involucrados en un sistema esto permite catalogarlos en cuatro categorías que dan pautas de que tiempo de acciones tomar con cada uno. Las categorías son:

- **Alto poder, alto interés:** Esta categoría de personas son las se deben atender con más atención y hacer el mayor esfuerzo por satisfacer. Los comunicados deben ser frecuentes y claros y pueden ser amplios.
- **Alto poder, poco interés:** Es necesario esforzarse para satisfacer a estos involucrados pero es necesario ser breve y claro al comunicarse con ellos pues de lo contrario pueden aburrirse.
- **Poco poder, alto interés:** Es conveniente mantener suficientemente informados a estos involucrados, es muy conveniente hablar personalmente con ellos para recibir puntos de vista diferentes y detectar problemas tempranamente.
- **Poco poder, poco interés:** Se deben monitorear a estos involucrados pero no es necesario poner demasiado esfuerzo o atención en ellos.

Adicionalmente es recomendable utilizar colores para establecer su actitud hacia el proyecto, rojo = negativa, naranja = neutra y verde = positiva.

P5 Descripción del Producto y Entregables

Descripción: Documento que especifica que se entregará.

Utilidad: Documenta la meta acordada dando claridad y certidumbre a clientes y desarrolladores.

Observaciones:

Es indispensable hacerlo al inicio del proyecto y que lo discutan el RAPE y el CAE y el RAPE y el ET también debe actualizarse con cada cambio. Es importante especificar además a quien, donde y bajo qué condiciones se entregará, es decir se debe acordar un protocolo de entrega. Tanto clientes como desarrolladores deben tener constancia de lo pactado. Es muy común que esta información esté incluida en el contrato pero no siempre es así. Si es el caso se debe plantear y acordar que como cuando y donde se debe entregar. Es importante recordar también que puede haber pagos y/o entregas por parte de los clientes y generalmente se vinculan unas con otras.

Este es producto importante y seguramente muchos involucrados harán varias consultas a lo largo del proyecto. Importante que sea un producto fácil de identificar y encontrar, si por ejemplo se por utilizar el correo electrónico para comunicarlo y simultáneamente almacenarlo es aconsejable dedicar un correo únicamente a este producto y no mezclarlo con otros. Igualmente si está incluido en el contrato es recomendable plasmar la información por separado en otro medio de forma tal que se pueda difundir sin distribuir el contrato que puede tener información no apta para difundirse.

P6 Plan de Manejo de Riesgos

Descripción: El repositorio del proyecto es el almacén de todos los productos del proyecto.

Utilidad: Mantiene la información del proyecto accesible y organizada.

Observaciones:

Debe Los riesgos son inherentes a todo proyecto. Prevenirlos o reaccionar adecuadamente ante su ocurrencia es indispensable. Un Plan de Manejo de Riesgos viable y sencillo es una tabla que relaciona los riesgos que ya hayas identificado que pueden ocurrir con la probabilidad de que ocurran el grado en que afectarían al proyecto y diversas medidas de contención y/o reacción para reducir el riesgo de incidencia y/o los efectos negativos. En caso de que tu grupo o empresa ya cuente con una base de conocimientos debes consultar los planes de riesgo de proyectos similares. Es importen notar que en proyectos WEB puedes considerar (salvo en muy contadas excepciones) que el cambio en los requisitos ocurrirá y por lo tanto no es un riesgo si no una característica. Uno de los riesgos que más frecuentemente ocurren es que los Clientes encargados de información se retrasen en sus entregas. Pero hay muchos otros como que el servidor de trabajo y el desarrollo tengan diferencias de configuración que vuelvan algún modulo del sistema inoperante, que se domine el lenguaje de desarrollo, que no se tengan fondos para hacer las adquisiciones más importantes, etc.

P7 Plan de Comunicación

Descripción: Comunicación especifica las formas y medios de comunicación entre los diversos involucrados en el proyecto.

Utilidad: Reduce los problemas de comunicación.

Observaciones:

El plan de El glosario, la agenda y los protocolos de cambio son componentes indispensables pero no suficientes. Es necesario establecer diversas normas o guías como pueden ser: los criterios para hacer reuniones, formatos de los correos, que tipo de información le es útil a cada involucrado, etc.

Depende del equipo de desarrollo los clientes y proyecto el detalle con que esto se documenta. Es importante no suponer nada, hay clientes que tienen requerimientos específicos de comunicación, suele haber involucrados que no conocen expresiones, términos ni/o agendas que pueden compartir otros involucrados que se conozcan previamente o compartan la misma formación.

P8 Cotización

Descripción: Documento que especifica formalmente el costo del sistema para el cliente. Además de los costos para el cliente las cotizaciones deben incluir condiciones de entrega tanto del sistema y sus entregables así como de los recursos que proporcione el cliente patrocinador al equipo de trabajo.

Utilidad: Establece los recursos que el cliente patrocinador debe proporcionar al equipo de desarrollo para que se construya el sistema.

Observaciones:

Se trate de un proyecto interno o de la venta de un sistema a medida, una de las principales demandas de los clientes es saber tiempo y costo. Hay muchas técnicas para calcular estos datos como puntos de función, regla de los tres puntos, datos históricos, juicio de expertos, etc. La elección de una técnica depende del equipo y el proyecto. Una práctica muy común es usar el juicio de expertos respaldada con información de proyectos anteriores. La estimación de costos en sí es toda una disciplina más allá del alcance de esta guía. Independientemente de la técnica que se utilice en la mayoría de los proyectos web se tendrá que hacer una cotización con requerimientos muy incompletos y sujeta a cambios.

La solución es cotizar haciendo supuestos sobre todas las variables desconocidas. Dichas suposiciones deben ser comunicadas y especificadas como condicionantes de la cotización presentada. Esto cubre la exigencia inicial y hace consciente al cliente de los muchos factores que influyen en los costos. Permite mejores acuerdos ante los cambios en los requerimientos.

Conforme se trabaje más en el proyecto, se resuelvan dudas y aparezcan cambios la cotización inicial se transformará en una cotización definitiva. Los supuestos pasarán a ser condiciones y según el proyecto y las prioridades del cliente se establecerán nuevos requerimientos, precios, plazos y/o niveles de calidad.

Es importante no dar fechas, precios o funcionalidades específicas sin condicionarlas. Hay que vincular estos factores a las acciones del cliente de las que dependen, como la aprobación del proyecto, los pagos y la oportuna participación y entrega de información. Por ejemplo: La primera fase del sistema que consta de... se entregará al mes de haber recibido los formularios de... Esto protege a los desarrolladores de la incertidumbre inicial y hace consciente al cliente de su corresponsabilidad en el proyecto. Si el proyecto exige la entrega en una fecha específica entonces se debe condicionar el precio y/o la calidad y/o se debe establecer una fecha límite para iniciar el proyecto.

P9 Repositorio del Proyecto

Descripción: Almacén de todos los productos del proyecto

Utilidad: Mantiene en forma accesible y organizada toda la información referente al proyecto.

Observaciones:

Debes tener normas para mantener organizado el repositorio del proyecto. El repositorio debe responder a la naturaleza de tus productos y puede incluso constar de una combinación de medios y

elementos. Puede ser un directorio en el sistema de archivos de tu computadora, un grupo de yahoo, msn, etc, un sitio operando con sistema especializado como sourceforge, un pizarrón con post-it's como en scrum y/o un archivero con los documentos impresos, etc. Lo importante es que cada miembro del equipo de trabajo sepa que lo puede hacer e incluso modificar lo que sea su responsabilidad así como cuándo y cómo hacerlo. En todos los medios que lo permiten debes tener un sistema de nombrado de los productos que permita almacenar y distinguir las diferentes versiones de cada documento. Igualmente es importante hacer un respaldo periódico del repositorio, aun en el caso de medio físico como el pizarrón o los documentos impresos esto es posible con fotografías y/o fotocopias.

P10 Proceso General de Desarrollo

Descripción: Descripción de las fases y metodología de trabajo que se adoptara en el proyecto.

Utilidad: Dar un punto de partida a la planificación del proyecto.

Observaciones:

El Proceso General de Desarrollo es el punto de partida para determinar las actividades. Está muy ligado con la metodología de trabajo del equipo. Las diversas metodologías dividen el desarrollo de software en diferentes fases. Por ejemplo en RUP se establecen Inicio, elaboración, construcción y transición y Scrum divide el desarrollo en ciclos que llamados Sprints que a su vez subdivide en sprints diarios y juntas de planeación. A partir de estos marcos de referencia que impone la metodología de trabajo debes establecer las características propias del proyecto cuantos ciclos o sprints, las fechas de entregas ya pactadas, etc. Esto da una lista de fases a hitos para el desarrollo de un sistema específico. A partir de estas grandes fases se establecen paulatinamente las actividades necesarias para general el trabajo necesario en cada una.

P11 WBS

Descripción: Lista de las tareas y subtareas necesarias para completar el sistema

Utilidad: Descompone tareas complejas en tareas pequeñas, comprensibles y ejecutables por los involucrados.

Observaciones:

WBS son las iniciales de Work Break Down Structure o estructura de descomposición del trabajo. Se toma la tarea principal, por ejemplo hacer la Especificación de requerimientos y se divide en sub tareas necesarias para lograrlo, ver al cliente, buscar alternativas de solución, presentar opciones al cliente, etc. Luego se toma cada una de estas y se vuelven a dividir en subtareas y así sucesivamente, hasta llegar al detalle deseado. Que debe contener actividades puntuales que los diferentes involucrados puedan ejecutar. Una forma muy efectiva de representarlo es utilizando un árbol o listado indexado en el que se listan todas las tareas y subtareas necesarias para desarrollar el sistema. Muchos elementos dependen de la definición del proceso de desarrollo sin embargo estos son a su vez necesarios para detallar correctamente todas las actividades y/o productos y tener así un WBS completo. Para romper este círculo de dependencias se elabora inicialmente un proceso de desarrollo general y conforme el tiempo y el proyecto avanzan se van detallando las secciones que sean pertinentes del mismo. Ve la sección de ejemplos para mayor detalle.

P12 Requerimientos de Adquisiciones y Capacitación

Descripción: Lista de elementos y conocimientos que se tienen que adquirir para poder realizar el proyecto.

Utilidad: Ayuda a la planeación y oportuna ejecución de las adquisiciones y capacitaciones.

Observaciones:

Muchos sistemas tienen requerimientos que implican el uso de lenguajes y herramientas de programación desconocidos por los programadores. En otros es más conveniente comprar algún módulo ya sea como un producto comercial o solicitado a medida a otro grupo de desarrolladores. Lo mismo puede ocurrir con equipo de cómputo, boletos para viajar, etc. Tener registro de todo esto es indispensable para adquirirlo oportunamente y no retrasar o alterar las actividades planeadas ni la calidad del producto.

P13 Calendario

Descripción: Especificación de las fechas planeadas de inicio y fin de las diversas actividades.

Utilidad: Clarifica que se debe hacer cada día y fase para desarrollar el sistema.

Observaciones:

Ubica en el tiempo las actividades establecidas en el WBS y por lo tanto evoluciona junto con este. Para establecer el orden cronológico de las tareas necesitas determinar dos factores las dependencias que existen entre una tarea y las otras pues por ejemplo no se puede empezar a programar si haber hecho los requerimientos y quien ejecutara cada tarea. En proyectos grandes y complejos la distribución de las tareas en el tiempo y la asignación de los recursos humanos y materiales para que se puedan llevar a cabo se vuelve tan complicado como importante. Hay muchas herramientas para hacer esta asignación desde programas especializados como Microsoft Project y OpenWorkBench hasta rótulos y cartulinas calendarizadas donde se marcan las tareas y los responsables de cada una. Uno de los modelos más populares para representar la calendarización de actividades es el diagrama de Gantt.

P14 Plan de Proyecto

Descripción: Documento formal usado como guía para la ejecución y control del proyecto.

Utilidad: Contiene toda la información para administrar y monitorear la ejecución del proyecto.

Observaciones: Este producto se forma al integrar los siguientes productos:

- Ciclos y Actividades
- Tiempo Estimado
- Plan de Adquisiciones y Capacitación.
- Equipo de Trabajo
- Costo Estimado

- Calendario
- Plan de Manejo de Riesgos
- Protocolo de Entrega

P15 Protocolo de Cambio

Descripción: Descripción clara y paso a paso del procedimiento para solicitar y aceptar los cambios así como de cómo reaccionar ante solicitudes de cambios de los clientes que no sigan el protocolo.

Utilidad: Establece las pautas para atender y procesar los cambios disminuyendo el desorden y el retrabajo.

Observaciones:

La Los cambios son muy frecuentes en todos los proyectos y los desarrollos WEB que destacan de entre otros sistemas justamente por la gran cantidad estos. El descontrol en los cambios puede llevar fácilmente al fracaso del proyecto y en cualquier caso generara trabajo innecesario y tensión entre desarrolladores y clientes. Lo más recomendable es establecer un procedimiento junto con los clientes para llevar acabo dichos cambios. Pero es muy poco frecuente que el cliente siga dicho procedimiento en primera instancia, los clientes expresan sus deseos y expectativas en cuanto les surgen y generalmente en forma oral y ante cualquier miembro del equipo de desarrollo. Las propias dimensiones del proyecto y la estrecha relación que puede tener una empresa pequeña o un equipo de desarrollo con los clientes fomentan que se pacten de informalmente muchos cambios en el proyecto. Es necesario establecer en el equipo de trabajo un protocolo para canalizar y tratar adecuadamente las solicitudes informales de cambios. El manejo de los cambios debe incluir una forma para solicitarlos, evaluarlos y aprobarlos o rechazarlos una forma para documentarlos y comunicarlos. Cuando un miembro del equipo de desarrollo reciba una solicitud de cambio informal debe evitar aceptarla o rechazarla y por el contrario guiar al solicitante a que siga el procedimiento pactado. También pueden existir cambios que no involucren a los clientes como si se decide cambiar una clase o método es importante que estos cambios se traten con el mismo cuidado que los solicitados por los clientes pues las consecuencias de no comunicarlos ni evaluar sus impactos son prácticamente idénticas.

En cuanto a la documentación y la comunicación esta debe ser congruente con los otros productos y modelo de trabajo elegido. En muchos proyectos se recomienda que incluso la solicitud de cambio sea un documento formal. Sin embargo se pueden establecer otros mecanismos como la solicitud y discusión personal con el RAPE y el CAE. En todo caso un cambio aprobado se debe documentar y comunicar incluyendo que productos afecta, porque se hizo y quienes evaluaron y autorizaron el cambio. Así por ejemplo si se están manejando casos de uso en UML con un repositorio accesible desde Internet es congruente actualizar los archivos cambiando el número de versión y enviar un email a todos los desarrolladores. Si en cambio se está usando un método un pinzaron como sugiere Scrum se deben plasmar los cambios en este hacer mención de los mismos durante la reunión inicia los sprints.

P16 Especificación de Requerimientos

Descripción: Describe a detalle qué funcionalidad y datos es necesario que tenga el producto final para cumplir lo planteado en la descripción del proyecto. También puede incluir otros requisitos referentes al equipo de trabajo, los plazos, la calidad y/o características del sistema.

Utilidad: Completa la información de la descripción del proyecto. Clarifica qué información debe contener y qué debe hacer el sistema para cumplir con los objetivos. Especifica características necesarias para que el sistema y el proyecto operen y/o se desarrollen en su contexto y circunstancias específicas.

Observaciones:

Los requerimientos determinan el diseño y la construcción del sistema y cambian constantemente en los desarrollos WEB. Es necesario no solo documentarlos sino documentar sus cambios y aclarar las consecuencias de dichos cambios con el equipo de trabajo y los clientes.

Éste es uno de los documentos más variables en los desarrollos WEB y se debe ajustar durante todas las etapas. Su elaboración y perfeccionamiento es paulatina y suele darse gracias a constantes revisiones con los clientes.

Los requerimientos suelen catalogarse en dos tipos, los funcionales que describen qué debe hacer el sistema, por ejemplo listar facturas o imprimir recibos, y los no funcionales que especifican las otras características del sistema como pueden ser el uso de un lenguaje específico, capacidad de operar en sistema operativo determinado, el formato en que se deben guardar los datos, que colores, logos o letras se deben usar en la interfaz gráfica, etc.

Para documentar los requerimientos no funcionales suele ser suficiente listarlos. Generalmente basta hablar con el CAE y/o el SCT. En el caso de los funcionales existen algunas posibilidades como son las listas, los casos de uso de UML e incluso hay quienes han utilizado los diagramas de BPMN (modelado de procesos de negocios por sus siglas en inglés). Lo importante es que la técnica seleccionada cumpla lo mejor posible dos condiciones: que sea suficientemente expresiva y precisa así como comprensible por clientes y desarrolladores. La documentación de los requerimientos requiere del intercambio de información entre desarrolladores y clientes. Se recomiendan entrevistas personales y telefónicas combinadas con el intercambio y revisión de los documentos en sí.

En algunas ocasiones los requerimientos funcionales se manejan en forma de lista con el cliente y sin embargo para fines del desarrollo se generan y utilizan los diagramas de casos de uso.

P17 Plan de Pruebas

Descripción: Descripción de las evaluaciones que se deben hacer al sistema, calendarización y responsables de las mismas. Las evaluaciones incluyen que se componente se está evaluando datos a introducir y que se espera obtener como resultado correcto.

Utilidad: Disminuye los errores en el sistema y el costo de encontrarlos

Observaciones:

Las pruebas aseguran un nivel mínimo de operatividad y funcionalidad del sistema. Durante la construcción del mismo se debe establecer que se probara y a qué grado. Generalmente se prueban los diversos componentes por separado y posteriormente todos juntos. Hay una gran variedad de metodologías de pruebas desde revisiones pro el mismo desarrollador o sus similares al momento de desarrollar hasta sistemas especializados que simulan miles de usuarios simultáneos. Mientras sea posible lo más recomendable es que las pruebas no las realice el desarrollador. La mejor metodología depende del proyecto. Además de las metodologías aplicadas en el desarrollo de software en general en sistemas WEB es posible involucrar directamente a los usuarios en la pruebas al usar el desarrollo basado en prototipos funcionales que ya se mencionó en la sección de prototipos.

P18 Análisis y Diseño

Descripción: Este documento contiene la descripción textual y grafica de la estructura de los componentes de software.

Utilidad: Describe el sistema, su descomposición en partes y el funcionamiento de las mismas.

Observaciones:

Consta de las siguientes partes:

Estructura de navegación: Contiene la estructura el sistema web en términos de pantallas (información) y ligas. Describe que información contendrá cada pantalla y como se accederá a esta.

Especificaciones y modelos de diseño: Especifica el diseño grafico que será utilizado a lo largo del sistema.

Descripción arquitectónica: Contiene la estructura interna del sistema, es decir la descomposición del sistema en subsistemas. Así como la identificación de los componentes que integran los subsistemas y las relaciones de interacción entre ellos.

Detalle de componentes: Contiene el detalle de los componentes que permita de manera evidente su construcción y prueba en el ambiente de programación.

P19 Software

Descripción: Sistema de software objetivo del proyecto.

Utilidad: Varía en cada proyecto.

Observaciones:

A pesar de su gran variedad los sistemas WEB suelen usar un grupo pequeño de patrones y por lo tanto el reutilizar diversos componentes es posible si se diseñan pensando en esta posibilidad.

P20 Registro de Errores y Cambios

Descripción: Repositorio con los reportes de errores y cambios.

Utilidad: Mantiene documentados y organizados los errores reportados y los cambios solicitados

Observaciones:

La forma de comunicar los errores y cambios debe ser establecida en el plan de comunicación. Puede tener componentes orales y registro en medios no específicos para esto como pizarrón de trabajo pero en todo caso debe documentarse de forma ordenada y consensada cada error y cambio. Cuando se usa el correo electrónico este puede ser simultáneamente el testimonio y el medio de comunicación.

También se pueden usar sistemas especializados en rastreo de errores como bugzilla o reportes detallados que incluyan revisor fecha pruebas ejecutadas etc. Como todos los otros productos el grado de detalle y formalismo así como el medio dependen del proyecto pero todos los casos es indispensable tener una forma homologada para todo el equipo de trabajo de registro de los errores encontrados y cambios requeridos.

P21 Manual de Mantenimiento

Descripción: Documento que especifica a los clientes como instalar y/o mantener operando el sistema.

Utilidad: Facilitar el mantenimiento del sistema.

Observaciones:

Un sistema WEB no debe tener manual de usuario pues debe ser auto explicativo o los navegantes simplemente cambiaran de sitio. Sin embargo los sistemas WEB deben cambiar y mantenerse continuamente. Por eso es importante entregar la documentación del desarrollo de mismo e incluso un manual con información sobre la instalación y las rutinas de mantenimiento que requiera el sistema.

P22 Base de Conocimientos

Descripción: Repositorio de los proyectos e información referente a los mismos que ha desarrollado la empresa o equipo.

Utilidad: Sirve para aprovechar el trabajo invertido en otros proyectos. Se pueden reutilizar tanto componentes de los sistemas como la información de la administración y el aprendizaje que se generaron durante los proyectos anteriores.

Observaciones:

Para el nivel de competencia que cubre la guía se pueden utilizar para revisar y reutilizar los componentes, las cotizaciones, el plan de riesgos y en general los todos los otros productos si resultan útiles para un proyecto en particular. En niveles de competencia más avanzados se genera un producto específico que se llama lecciones aprendidas que complementa los productos generados durante el proyecto con notas y observaciones específicas para guardarse en la base de conocimientos y ser utilizadas en futuros proyectos.

P23 Directorio

Descripción: Directorios adecuados a cada grupo de involucrados con los involucrados su rol y sus datos de contacto.

Utilidad: Facilitar la comunicación entre los involucrados, que deben interactuar entre sí.

Observaciones:

La comunicación es la principal herramienta del administrador del proyecto. El primer paso para que esta sea posible es tener un directorio con todos los involucrados y diversos medios para contactarlos. Es muy importante procurar más de una forma de contacto por cada involucrado pues ninguno es 100% fiable. Ya que no todos los involucrados deben comunicarse entre sí, es importante generar directorios específicos para cada involucrado o grupo de involucrados, que contengan únicamente los datos de los involucrado con quienes debe interactuar.

Referencias bibliográficas:

- [1] **Modelo de Procesos para la Industria de Software MoProSoft Por Niveles de Capacidad de Procesos Versión 1.3.** Secretaría de economía Agosto 2005.
- [2] O. Pastor, D. Schwabe, G. Rossi, et al. **Web Engineering: Modelling and Implementing Web Applications** Springer London Ltd Octubre 2007.
- [3] G. Kappel, B. Pröl, et al. **WEB Engineering: The Discipline of Systematic Development of Web Applications**, John Wiley & Sons, Ltd Julio 2006.
- [4] **A Guide to the Project Management Body of Knowledge** third edition. Project Management Institute 2004.
- [5] Wolf-Gideon Bleek, Martti Jeenicke, Ralf Klischewski **Developing Web-based Applications through e-Prototyping** Proceedings of the 26 th Annual International Computer Software and Applications Conference IEEE 2002.
- [6] Eric Holter **Client vs. Developer Wars** Newfangled Web Factory 2006.
- [7] Davide Bolchini , John Mylopoulos **From Task-Oriented to Goal-Oriented Web Requirements Analysis** Proceedings of the Fourth International Conference on Web Information Systems Engineering IEEE 2003.
- [8] Victoria Torres, Javier Muñoz, Vicente Pelechano **A Model Driven Method for the Integration of Web Applications** Proceedings of the Third Latin American Web Congress IEEE 2005.
- [9] Rashid Ahmad2, Zhang Li, Farooque Azam **Web Engineering: A New Emerging Discipline** International Conference on Emerging Technologies IEEE 2005.
- [10] Lasse Vogelsang, Peter Carstensen **New Challenges for the Collaboration in Web-based Information Systems Development** IEEE 2001
- [11] <http://www.software.net.mx/desarrolladores/prosoft/>
- [12] <http://www.iso.org/>
- [13] <http://alarcos.inf-cr.uclm.es/Competisoft/>
- [14] <http://www.w3.org/>
- [15] <http://www.sei.cmu.edu/cmml/>
- [16] **SWEBOK, Trial Version.** Software engineering Coordinating Committee, Computer Society, Software Engineering Institute. 2001.
- [17] **Stakeholder Analysis.** http://www.mindtools.com/pages/article/newPPM_07.htm
- [18] Kevin Mullet, Darrell Sano **Designing Visual Interfaces, communication oriented techniques** Sun Press, 1995
- [19] <http://webpatterns.org/>
- [20] <http://www.ui-designpatterns.org/>
- [21] http://www.w3schools.com/media/media_intro.asp
- [22] Rational Software **Rational Unified Process Best Practices for Software Development Teams.** 1998

Referencias a aplicaciones

Servidores:

<u>Server2go</u>	http://www.server2go-web.de/
XAMPP	http://www.apachefriends.org/en/xampp.html
Tomcat	http://sourceforge.net/projects/easytomcat

Administración de proyectos:

MS-Project	http://office.microsoft.com/en-us/project/default.aspx
Open WorkBench	http://www.openworkbench.org/

Editores e IDEs:

<u>DreamWeaver</u>	http://www.adobe.com/products/dreamweaver/
HTMLKit	http://www.chami.com/html-kit/
Eclipse for LAMP	http://www.easyeclipse.org/site/distributions/lamp.html

Diseño gráfico:

PhotoShop	http://www.adobe.com/products/photoshop/index.html
Gimp	http://www.gimp.org/
Corel Draw	http://www.corel.com/
Ink Space	http://www.inkscape.org/

Control de código:

Subversion	http://subversion.tigris.org/
Source Safe	http://msdn.microsoft.com/en-us/vstudio/aa700907.aspx

Control de errores:

Bugzilla	http://www.bugzilla.org/
----------	---

Sitios de interés

Ingeniería de software:

Comunidad Moprosoft

<http://www.comunidadmoprosoft.org.mx/>

Project Management Institute

<http://www.pmi.org/>

Software Engineering Institute

<http://www.sei.cmu.edu/>

Patrones para web:

Web Patterns Project

<http://www.ui-designpatterns.org/index.html>

WebPatterns

<http://webpatterns.org/>

Tutoriales y referencias de lenguajes:

W3Schools

<http://www.w3schools.com/>

JavaScript Reference

<http://www.javascriptkit.com/jsref/>

PHP

<http://www.php.net/>

Ejemplos Anexos:

Consultar el CD-ROM anexo.