

UNIVERSIDAD PONTIFICIA DE SALAMANCA
FACULTAD DE INFORMÁTICA



Programa de Doctorado en Ingeniería de Software

Bienio 2003-2005

Calidad en Ingeniería del Software

Trabajo de Investigación:

"Propuesta de un Modelo de Calidad del software aplicado a la Web"

Profesor: Dr. Vidal Alonso Secades

Alumno: César Parejas Llanovarced

Madrid, Julio 2004

Propuesta de un Modelo de Calidad del software aplicado a la Web

Un modelo de calidad tiene en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquisidores y usuarios finales (ISO, 2001); el cual es usado para construir mejores productos y asegurar su calidad.

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto software y la calidad del proceso de desarrollo de éste (calidad de diseño y fabricación). No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos a establecer de calidad del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos, de ésta. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

La importancia que las tecnologías Web han cobrado en los últimos años hace que se preste una atención especial a su calidad. Una forma de asegurar esta calidad es mediante la utilización de modelos específicamente definidas para el entorno Web que nos permitan controlar la calidad de los diseños realizados. Además, para que los modelos sean realmente útiles deben ser validadas mediante métricas tanto de manera formal como empírica.

En general, un modelo de calidad descompone la calidad jerárquicamente en una serie de características y subcaracterísticas que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad y usan generalmente como base la ISO9126

Aunque esas listas de propiedades proveen un punto de partida útil para entender y mejorar la calidad, la tónica general es la escasez de consenso en dichas propuestas, con una gran cantidad de esfuerzo dedicado a tareas ya desarrolladas por otros autores.

Nuestro objetivo, bajo estas consideraciones, es el de proponer un modelo de calidad orientado a aplicaciones Web tratando no de excluir el planteamiento

de otros modelos definidos, sino de aglutinar a los mismos adecuándolos para la consecución de un modelo fortalecido desde nuestro punto de vista.

El Modelo de Calidad Propuesto

Para la realización de nuestro modelo, tomamos como referencia el WQM (Web Quality Model) [1]. El modelo WQM es un modelo global de calidad de la Web que se basa en una perspectiva tridimensional desde los cuales el usuario puede contemplar la calidad de un producto Web. Estas dimensiones son las siguientes (**Fig. 1**):

- Características Web
- Características de calidad
- El proceso del ciclo de vida

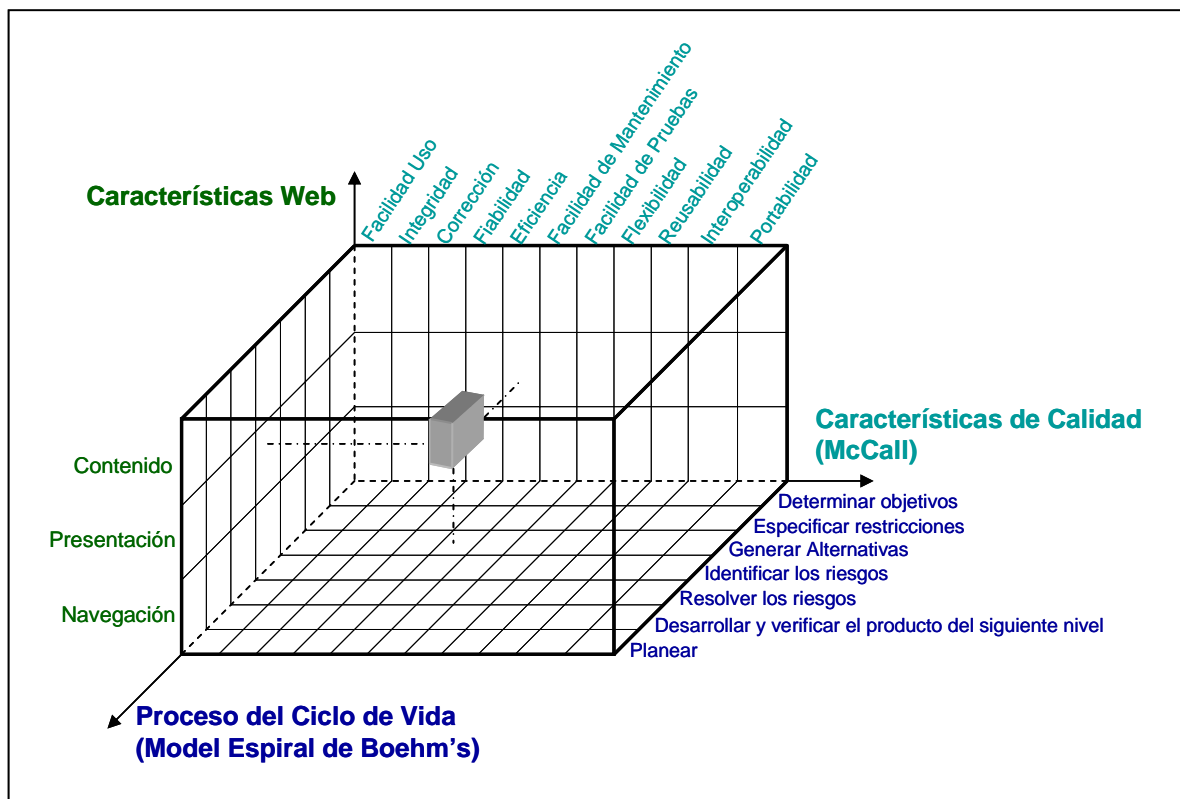


Fig. 1. El modelo de calidad Web Propuesto (Basado en WQM)

1. Dimensión de características Web

Utilizamos tres de los factores más difundidos en la literatura para caracterizar un sitio Web:

- Contenido

- Presentación
- Navegación

Aunque podríamos incluir influencias de docenas de campos, estas tres características son consideradas básicas para todo desarrollo de sitios Web. El objetivo principal del contenido es informar, o quizá persuadir a los usuarios. La navegación va unida a la tecnología que se utiliza para dotar de funcionalidad al sitio. Los temas de presentación proporcionan el aspecto del sitio. Por supuesto, la influencia de cada característica particular de los sitios Web varía según el tipo de sitio que se construye. Una intranet para una empresa de producción puede que no necesite las mismas consideraciones visuales que un sitio Web público para la promoción de una película, etc.

2. *Dimensión de características de calidad del Producto*

Como base propuesta para definir esta dimensión, empleamos el modelo de calidad de McCall por ser uno de los más difundidos y porque además ha servido de base para otros modelos (el modelo de Boehm, el modelo de Arthur, el modelo de Deutsch y el Software Quality Management –SQM- de Murine).

El modelo de McCall organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto, basándose en once factores de calidad organizados en torno a los tres ejes y a su vez cada factor se desglosa en otros criterios:

Puntos de Vista	Factor	Criterios	Descripción
Operación del Producto	Facilidad de Uso	Facilidad de operación	Atributos del software que determinan la facilidad de operación del software.
		Facilidad de comunicación	Atributos del software que proporcionan entradas y salidas fácilmente asimilables.
		Facilidad de aprendizaje	Atributos del software que facilitan la familiarización inicial del usuario con el software y la transición del modo actual de operación.
		Formación	El grado en que el software ayuda para permitir que nuevos usuarios apliquen el sistema.
	Integridad	Control de accesos.	Atributos del software que proporcionan control de acceso al software y los datos que maneja.
		Facilidad de auditoría	Atributos del software que facilitan la auditoría de los accesos al software.
		Seguridad	La disponibilidad de mecanismos que controlen o protejan los programas o los datos.
	Corrección	Compleitud	Atributos del software que proporcionan la implementación completa de todas las funciones requeridas.
		Consistencia	Atributos del software que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación.
		Trazabilidad o rastreabilidad	Atributos del software que proporcionan una traza desde los requisitos a la implementación con respecto a un entorno operativo concreto.
	Fiabilidad	Precisión	Atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados.
		Consistencia.	Atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados.
		Tolerancia a fallos	Atributos del software que proporcionan una estructura de módulos altamente independientes.
		Modularidad	Atributos del software que posibilitan la implementación de funciones de la forma más comprensible posible.
		Simplicidad	La precisión de los cálculos y del control.
		Exactitud	Atributos del software que minimizan el tiempo de procesamiento.
	Eficiencia	Eficiencia en ejecución	Atributos del software que minimizan el espacio de almacenamiento necesario.
Eficiencia en almacenamiento			

Puntos de Vista	Factor	Criterios	Descripción
Revisión del Producto	Facilidad de mantenimiento	Modularidad.	
		Simplicidad.	
		Consistencia.	
		Concisión	Atributos del software que posibilitan la implementación de una función con la menor cantidad de códigos posible.
	Facilidad de prueba	Auto descripción	Atributos del software que proporcionan explicaciones sobre la implementación de las funciones.
		Modularidad.	
		Simplicidad.	
		Auto descripción.	
	Flexibilidad	Instrumentación	Atributos del software que posibilitan la observación del comportamiento del software durante su ejecución para facilitar las mediciones del uso o la identificación de errores.
		Auto descripción.	
		Capacidad de expansión	Atributos del software que posibilitan la expansión del software en cuanto a capacidades funcionales y datos.
		Generalidad	Atributos del software que proporcionan amplitud a las funciones implementadas.
Transición del Producto	Reusabilidad	Modularidad.	
		Generalidad.	
		Auto descripción.	
		Independencia entre sistema y software	Atributos del software que determinan su dependencia del entorno operativo.
	Interoperabilidad	Independencia del hardware	Atributos del software que determinan su dependencia del hardware.
		Modularidad.	
		Compatibilidad de comunicaciones	Atributos del software que posibilitan el uso de protocolos de comunicación e interfaces estándar.
		Compatibilidad de datos	Atributos del software que posibilitan el uso representaciones de datos estándar.
	Portabilidad	Estandarización en los datos	El uso de estructuras de datos y de tipos estándar a lo largo de todo el programa.
		Auto descripción.	
		Modularidad.	
		Independencia entre sistema y software.	
		Independencia del hardware.	

3. Dimensión del proceso del ciclo de vida

En el desarrollo tradicional se han utilizado estándares y modelos de evaluación y mejora de los procesos de software. Los estándares más utilizados son el ISO 9000-3, ISO/IEC 12207 y el ISO/IEC 15504 (SPICE). Los modelos de calidad están orientados a la satisfacción de los clientes y mejora de los procesos mediante la definición de principios y prácticas que conducen a mejores productos de software. Los principales modelos de calidad que han sido propuestos son el Ticket, CMM (Capability Maturity Model), Boehm, Bootstrap y Trillium. En el ámbito de la Web, el estudio de la calidad de procesos de desarrollo es muy reciente. La mayoría de las veces se emplean procesos "ad hoc" y todavía no existen estándares para este tipo de de entorno.

En nuestra propuesta, para definir esta dimensión, hemos utilizado el modelo de espiral de Boehm's [Boehm, 1987] debido a que consideramos se adapta mejor a los cambios frecuentes del desarrollo de software, un factor que caracteriza en especial a las aplicaciones Web.

El modelo espiral se enfoca en abordar los riesgos en forma incremental, de acuerdo a su prioridad (**Fig. 2**). Cada ronda está compuesta por cuatro fases. Durante la primera fase los desarrolladores exploran alternativas, definen restricciones e identifican los objetivos. Durante la segunda fase los desarrolladores manejan los riesgos asociados con las soluciones definidas durante la primera fase. Durante la tercera fase los desarrolladores realizan y validan un prototipo o la parte del sistema asociada con los riesgos que se tratan en esta ronda. La cuarta fase se enfoca en la planeación de la siguiente ronda con base en los resultados producidos en la ronda actual. La última fase de la ronda se realiza por lo general, como una revisión que involucra a los participantes en el proyecto, incluyendo a los desarrolladores, clientes y usuarios. Esta revisión abarca los productos desarrollados durante las rondas anteriores y actual y los planes para la siguiente ronda. El modelo espiral de Boehm distingue entre las siguientes rondas: Conceptos de operación, Requerimientos de software, Diseño de productos de software, Diseño detallado, Código, Prueba unitaria, Integración y pruebas, prueba de aceptación, Implementación.

Cada ronda sigue el modelo de cascada e incluye las siguientes actividades:

1. Determinar objetivos
2. Especificar restricciones
3. Generar alternativas
4. Identificar los riesgos
5. Resolver los riesgos
6. Desarrollar y verificar el producto del siguiente nivel
7. Planear

Las dos primeras actividades definen el problema que se trata en el ciclo actual. La tercera actividad, Generar alternativas, define el espacio de solución. Las actividades Identificar riesgos y Resolver riesgos sirven para identificar problemas futuros que pueden causar costos elevados o la cancelación del proyecto. La actividad Desarrollar y verificar el producto del siguiente nivel es la realización del ciclo. La actividad Planear es una actividad administrativa que prepara el siguiente ciclo.

Estas rondas pueden verse en un sistema de coordenadas polares, como se muestra en la siguiente figura. La primera ronda, Conceptos de operación, se inicia en el cuadrante superior izquierdo. Las rondas subsiguientes se representan como capas adicionales de la espiral. La notación permite que se determine con facilidad el estado del proyecto a lo largo del tiempo. La distancia con respecto al origen es el costo acumulado del proyecto. La coordenada angular indica el avance logrado en cada fase.

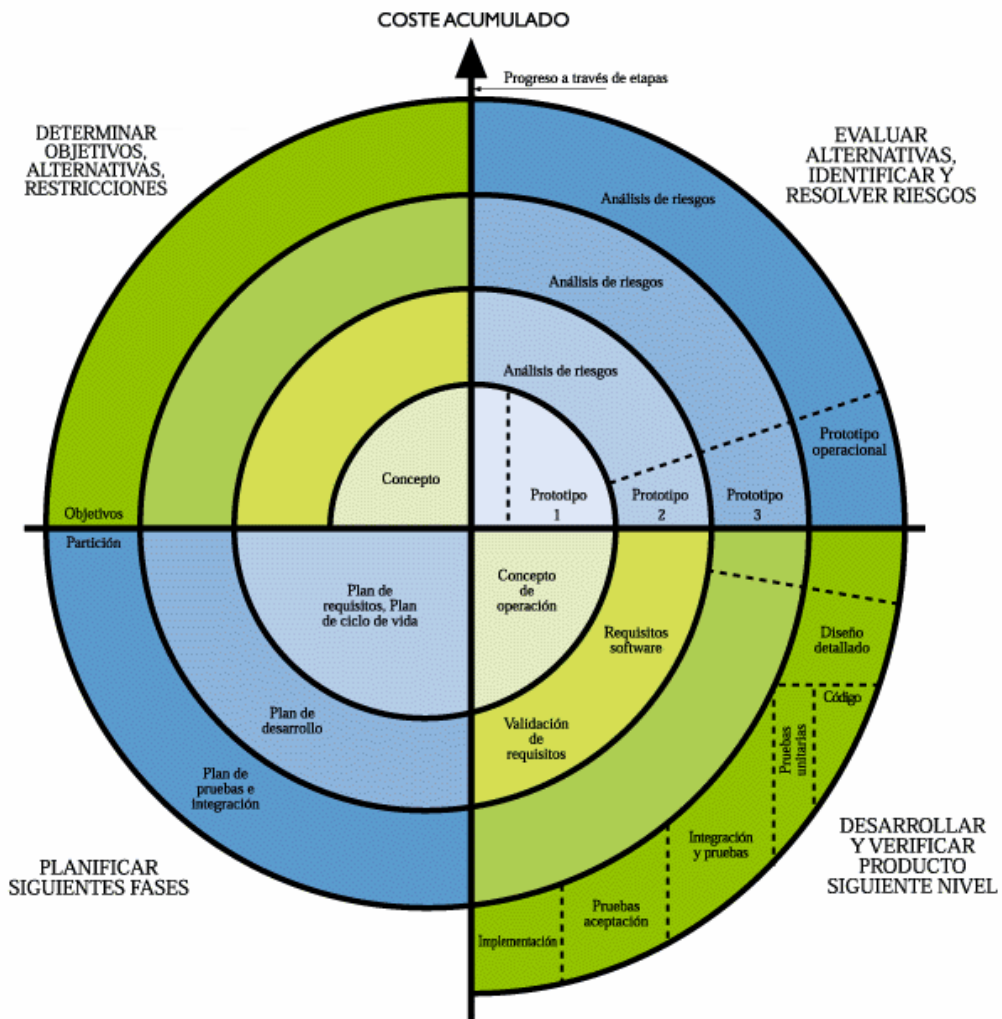


Fig. 2. El modelo Espiral de Boehm

Clasificación de las Métricas

En la estimación tradicional, especialmente para el desarrollo cliente/servidor, existe una aplicabilidad práctica en los proyectos Web. Sin embargo, existen a su vez diferencias cruciales que requiere que los modelos y métricas utilizadas actualmente sean adaptados a los entornos Web.

Cleary ha presentado la métrica Web-Points (Cleary, 2000) para estimar el tamaño y esfuerzo proporcionados por los sitios Web estáticos. El tamaño de los web-points se obtiene considerando la complejidad de las páginas HTML del sitio. Se clasifica la complejidad en los niveles bajo, mediano y alto. El número de web-points de cada página se obtiene basándose en el tamaño de

la página en palabras, el número de enlaces existentes y el número de elementos no textuales de la página Web.

Actualmente, se han publicado una serie de guías de estilos y criterios (W3C, 1999) que ayudan a mejorar el diseño y autoría de las aplicaciones Web con relación a aspectos de usabilidad, navegabilidad, accesibilidad, entre otros. Más específicamente el aspecto usabilidad ha sido objeto de estudio y se ha publicado interesantes guías como las propuestas por Nielsen (2002). Estas guías describen metas a seguir en un censo común, pero las indicaciones están a un nivel muy alto y se basan en evidencias observacionales. En este sentido, un análisis cuantitativo de una colección de sitios Web (Ivory et al., 2001) revela que el uso de métricas de páginas Web puede predecir si un sitio sigue las buenas prácticas descritas por estas guías.

Este análisis también proporciona la evidencia empírica de que métricas de composición (tales como número de páginas, número de archivos multimedia, número de enlaces internos/externos de una página y gráficos), de formato de página (como complejidad de la página) u otras características de página que difieren entre los distintos dominios Web (educativos, comunitarios o financieros).

Para estimar y probar el desarrollo de nuestro modelo propuesto, empleamos un subconjunto de las métricas de Ivory, que lo analizamos y establecemos una clasificación para cada dimensión en función de las características que controla. Se presenta el resumen en el siguiente cuadro:

Métrica	Descripción	Validación Empírica	Clasificación Propuesta		
			Características de Calidad	Ciclo de Vida	Características Web
Número de Páginas	Número de ficheros HTML o SHTML	Sí	Usabilidad	Requerimientos de software	Contenido
			Mantenibilidad	Diseño de Productos del Sistema	Presentación
			Mantenibilidad	Implementación	Presentación
				Prueba de aceptación	
Número de archivos multimedia	Número de ficheros multimedia (gráficos, audio, video, animación, imágenes)	Sí	Usabilidad	Diseño de Productos del Sistema	Contenido
			Eficiencia	Validación del diseño	Presentación
			Portabilidad	Implementación	
				Prueba de aceptación	
Número de programas	Número de scripts CGI, ficheros Java Script, applets de Java de una aplicación web.	Sí	Funcionalidad	Todas las etapas del ciclo de vida	Contenido
			Fiabilidad		Presentación
			Usabilidad		
			Portabilidad		
			Mantenibilidad		
Complejidad de la página	Número de tipos de ficheros multimedia diferentes usados en una página	Sí	Usabilidad	Prototipo	Presentación
				Diseño de Productos del Sistema	
				Validación del diseño	
Grado de concentración de los nodos de un hiperdocumento	Muestra como está conectada la estructura de un hiperdocumento. Cálculo mediante sus nodos y enlaces. (Valor 1 si todos los nodos están)	Experimental	Usabilidad	Diseño de Productos del Sistema	Navegación
			Mantenibilidad	Validación del diseño	
Número de enlaces internos de una página	Número de enlaces que apuntan a páginas del mismo sitio web (no páginas recargadas)	Sí	Usabilidad	Diseño de Productos del Sistema	Navegación
			Eficiencia	Validación del diseño	
			Mantenibilidad		
Número de enlaces externos de una página	Número de enlaces que apuntan a páginas de fuera del hiperdocumento	Sí	Usabilidad	Diseño de Productos del Sistema	Navegación
			Mantenibilidad	Validación del diseño	
Número de enlaces que son imágenes	Número de enlaces que vienen representados mediante una imagen en una página (comprensión vs sobrecarga)	Sí	Usabilidad	Diseño de Productos del Sistema	Presentación
			Mantenibilidad	Validación del diseño	
Superficies de imágenes	Total de superficie de la página usada por imágenes que representan enlaces de la misma	Sí	Usabilidad	Diseño de Productos del Sistema	Presentación
			Eficiencia	Validación del diseño	

Conclusiones

- En el ámbito de la Web existen diferentes propuestas de metodologías, marcos de calidad, modelos de estimación, guías de estilos, criterios y métricas específicas tanto para el desarrollo, como para evaluar la calidad del producto en su conjunto y la calidad en uso de las aplicaciones Web. Sin embargo, no existen aun estándares ni puestas en común de todas estas iniciativas. Bajo este marco, hemos planteado un modelo que pretende unificar algunos modelos desde perspectivas diferentes para una mejor visión de la calidad del producto por parte del usuario.

- La terna (Facilidad de Uso, Diseño de productos del sistema, Navegación) sobresale como un valor de clasificación constante. (Fig. 3)

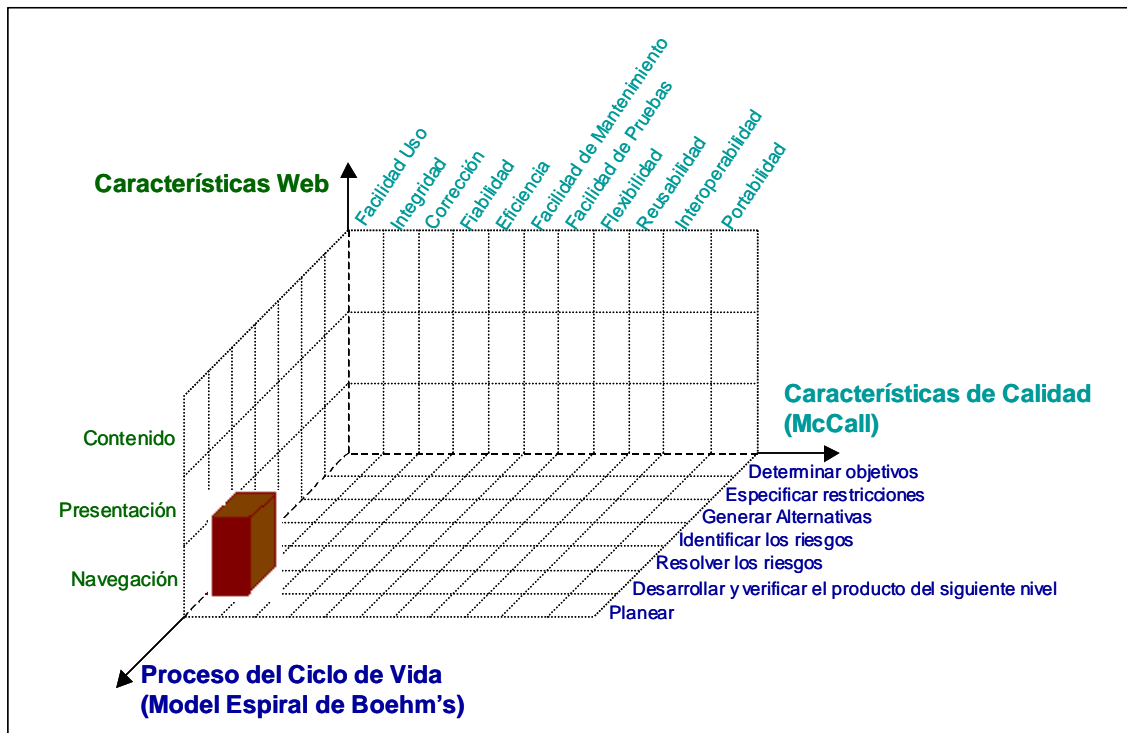


Fig. 3. Terna de clasificación que sobresale en el uso de las métricas.

- La complejidad inherente a las aplicaciones Web se debe a sus características propias, más la integración con aplicaciones ya existentes en ambientes diversos. Mientras que en el desarrollo tradicional el objetivo principal es la construcción de productos software de calidad con el mínimo coste, en el desarrollo Web, el objetivo es la puesta de productos de calidad en el mercado lo más breve posible. El tamaño típico de un proyecto tradicional va del mediano a grande y en él participa un equipo de decenas de miembros. Un proyecto Web suele ser un proyecto pequeño donde participa un equipo de 3 a 5 miembros. El tiempo de entrega de un proyecto tradicional es de aproximadamente 10 a 18 meses, mientras que el de un proyecto Web es de 3 a 6 meses.
- El análisis de la calidad de un sitio Web como un todo unificado es una difícil tarea debido a la necesaria mezcla de disciplinas que dependen de la naturaleza del sitio Web.

Bibliografía

Boehm, B.W., Brown, J.R., Lipow, M.; Quantitative Evaluation of Software Quality, Proceedings 2nd International Conference on Software Engineering, pp. 592-605, 1983.

Cavano, J.P., McCall, J.A., A Framework for the Measurement of Software Quality, Proc. of the ACM Software Quality Assurance Workshop, Nov. 1982.

Dra. Coral Calero, Grupo ALARCOS, Marco para la Calidad Web. Ene. 2004.

Ivory, M. Y., Rinha, R. R. y Hearst, M. A. (2001). Empirically Validated Web.

Kan S., Metrics and Models in Software Quality Engineering, Addison-Wesley Publishing Company, 1995.

Piattini M., García F., Calidad en el Desarrollo y Mantenimiento del Software, Ra-ma, Paracuellos de Jarama, 2002.

Thomas A. Powell, Diseño de sitios Web, 2001

Bernd Bruegge, Allen H. Dutoit, Ingeniería de Software Orientado a Objetos, 2002