

# Taller No. 1

JOHN ALEJANDRO GRIRALDO Y ADRIAN VARGAS

MARZO 07

## Índice

|                          |    |
|--------------------------|----|
| 1. MODELOS AD-HOC        | 3  |
| 2. MODELO FURPS          | 3  |
| 3. MODELO DE MCCALL      | 5  |
| 4. MODELO ISO/IEC 15504: | 5  |
| 5. MODELO DE BOEHM       | 6  |
| 6. PARTE No 2            | 9  |
| REFERENCIAS              | 10 |

## INTRODUCCIÓN

Los Modelos de Calidad son herramientas que guían a las Organizaciones a la Mejora Continua y la Competitividad dándoles especificaciones de qué tipo de requisitos debe de implementar para poder brindar productos y servicios de alto nivel. Conjunto de criterios agrupados en áreas o capítulos que sirven como referencia para estructurar un plan de calidad total en una empresa u organización, o de una de sus partes.

## 1. MODELOS AD-HOC

para monitorear la calidad de software, se pueden tomar dos caminos: Adoptar un modelo fijo se supone que todos los factores de calidad importantes son un subconjunto de los de un modelo publicado; se acepta el conjunto de criterios y métricas asociados al modelo Desarrollar un modelo propio de calidad se acepta que la calidad está compuesta por varios atributos, pero no se adopta lo impuesto por modelos existentes. En este último caso, se debe consensuar el modelo con los clientes antes de empezar el proyecto Se deciden cuáles atributos son importantes para el producto, y cuáles medidas específicas los componen. Mendoza et al. (2005)

### Ventajas

No se necesita tener una conexión a internet. Se puede acceder a los datos de las pc's conectadas a dicha red. La información viaja directamente entre el emisor y receptor sin pasar por ningún otro equipo. No necesitan administración central ni tampoco algún tipo de dispositivo de forma cableada Las configuraciones para crear dicha red son mínimas.

### Desventajas

Si las tarjetas de red no son compatibles no se puede realizar la conexión en modo Ad – Hoc Se debe configurar dicha conexión cada vez que se va a utilizar. Los equipos deben estar dentro del área de cobertura de los demás para comunicarse con ellos Los nodos compiten por acceder al medio compartido inalámbrico. A menudo esta competencia resulta en colisiones (interferencia), por estar transmitiendo dos o más nodos a la vez.

## 2. MODELO FURPS

Tomando como base el modelo de McCall, Hewlett-Packard lo desarrolla en 1987. Los factores de calidad que lo componen son los siguientes: funcionalidad (Functionality), usabilidad (Usability), confiabilidad (Reliability), desempeño (Performance) y capacidad de soporte (Supportability). Formando un acrónimo de donde proviene su nombre. Se divide en dos grupos:

Requerimientos Funcionales (F): Especifican que el software debe ser capaz de realizarse sin tomar en cuenta restricciones físicas, y se definen a través de las entradas y salidas esperadas.

### Funcionalidad

Los requisitos de funcionalidad deben incluir

- Conjunto de Características,
- Capacidades

- Seguridad.

## **Requerimientos no funcionales (URPS)**

Describen atributos del sistema o del ambiente.

### **Usabilidad**

Deben incluir subcategorías tales como:

- Factores humanos.
- Estéticos
- Consistencia en la Interfaz de Usuario.
- Ayuda en línea.
- Asistentes.
- Documentación del usuario.
- Material de capacitación.

### **Confiabilidad**

Se considera requisitos de confiabilidad:

- Frecuencia y severidad de fallas.
- Recuperación a fallos.
- Tiempo entre fallos.

### **Desempeño (Rendimiento)**

Un requisito de rendimiento impone condiciones a los requisitos funcionales. Por ejemplo: a una acción dada, se pueden especificar los siguientes parámetros de rendimiento:

- Velocidad.
- Eficiencia.
- Disponibilidad.
- Tiempo de Respuesta.
- Tiempo de Recuperación.
- Utilización de Recursos.

## Soporte

Los requisitos de soporte pueden incluir:

- Requisitos de instalación.
- Requisitos de Configuración.
- Requisitos de Adaptabilidad.
- Requisitos de Compatibilidad.

El desarrollo incluye además restricciones de diseño y requerimientos de implementación, físicos y de interfaz. No toma en cuenta la portabilidad de los productos. Los cuales conforman el FURPS +

## 3. MODELO DE MCCALL

El modelo de McCall fue el primero en ser presentado en 1977 y se originó motivado por Air Forcé y Dod Se focaliza en el producto final identificando atributos claves desde el punto de vista del usuario. Estos atributos se denominan factores de calidad y son normalmente atributos externos. pero también se incluyen algunos atributos posiblemente internos. los factores de calidad son demasiados abstractos para ser medidos directamente, por lo que por cada uno de ellos se introduce atributos de bajo nivel denominados criterios de calidad. algunos criterios de calidad son atributos internos según McCall que el atributo interno tiene un efecto directo en el atributo externo correspondiente, este modelo es usado en Colombia por la universidad de Pereira UTP.

## 4. MODELO ISO/IEC 15504:

Permite adaptar la evaluación para procesos en pequeñas y medianas empresas (pymes) y grupos de desarrollo pequeños, mediante la estructuración en seis niveles de madurez: Nivel 0- Organización inmadura, Nivel

- Organización básica,
- Organización gestionada,
- Organización establecida,
- Organización predecible y Nivel
- Organización optimizando. Su objetivo
- es llegar a que la organización logre ser madura, lo

- cual conlleva que la organización tenga procesos definidos, responsabilidades definidas, predicción de resultados, productos entregados con calidad, que las entregas se den en los tiempos pactados, incrementar la productividad, clientes satisfechos, y empleados felices (Córdoba,

2012).Garzás et al. (2009)

## 5. MODELO DE BOEHM

Presentado en el año de 1978, se denomina también modelo espiral ya que realiza un conjunto de iteraciones, estas no son fijas debido a que el equipo de trabajo es quien las determina; es un modelo que por naturaleza utiliza jerarquías, así mismo por cada vuelta que ejecuta tiene etapas que se describen de la siguiente forma:

Planeación: Se determinan los objetivos, alternativas y restricciones del proyecto.

Análisis de riesgos: Se realiza el análisis de las alternativas, se identifican y se resuelven los riesgos encontrados.

Ingeniería: Se procede al desarrollo del producto hasta .el siguiente nivel”.

Evaluación: El cliente realiza la valoración verificando los resultados obtenidos.

### Regiones de Tareas del Modelo

Este modelo en espiral (Boehm), posee un número de actividades de marco de trabajo, denominadas también regiones de tareas. El siguiente gráfico identifica dentro del modelo espiral cada una de sus regiones.

- Comunicación con el cliente: Tareas requeridas para establecer la comunicación existente entre el desarrollador y el cliente.
- Planificación: Se definen recursos, tiempo e información extra que se relacione con el proyecto.
- Análisis de riesgos: Se realiza un conjunto de tareas para evaluar riesgos de gestión y técnicos.
- Ingeniería: Tareas requeridas que permitan realizar una o varias representaciones de la aplicación.
- Construcción y acción: Tareas que nos ayuden con la construcción, pruebas, instalación y soporte para el usuario (documentación).
- “Evaluación del cliente: Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación. Cada una de las regiones está compuesta por un conjunto de

tareas del trabajo, que se adaptan a las características del proyecto que va a emprender se. Para proyectos pequeños, el número de tareas de trabajo y su formalidad es bajo, para proyectos mayores y más críticos cada región contiene tareas de trabajo que se definen para lograr un nivel más alto de formalidad. En todos los casos, se aplican las actividades de protecc

## **ventajas**

Tener una oportunidad para corregir los procesos de software que se hayan desajustado con el tiempo. Reducir los costos en todos los procesos. Cambiar la actitud del personal de la empresa. Realizar una mejora continua en la calidad de los procesos de software utilizados, servicios y productos de software. Lograr que la empresa de software sea más competitiva. Aumentar la productividad, efectividad y utilidad de la empresa. Asegurar la satisfacción de los clientes internos y externos. Tener productos de software y servicios con valor agregado. Tener permanentemente mejores procesos, productos de software

## **Desventajas**

los cambios pueden causar confusión cuando el equipo del proyecto comienza. Los proyectos reales raras veces siguen el modelo secuencial que propone el modelo. Es difícil afrontar cambios propuestos por el cliente en fases diferentes de la fase de análisis y definición de requisitos y entre más adelantado esté el proyecto más costoso son los cambios o modificaciones. A menudo es difícil que el cliente exponga explícitamente todos los requisitos. Un grave error puede ser desastroso si no se detecta hasta que se revisa el programa. El cliente debe tener paciencia.



## 6. PARTE No 2

La empresa mundo software cuenta con más de 30 sucursales dispersa en todo el país Su principal función es la creación de software empresariales la empresa no cuenta con un buen modelos de calidad de software por lo que se le hace la recomendación que adopten el modelo McCall ya que este se enfoca especificamente en que el producto final sea de la mejor calidad posible y satisfaciendo la necesidad del cliente

- ventaja: al ser un software empresarial solo se
- crea con el fin de cubrir las necesidades específicas cada cliente

desventaja: ser una empresa que crea software empresariales solo su mantenimiento servicios técnicos y actualizaciones pueden ser generados por nuestra empresa

## Referencias

- Garzás, J., Fernández, C. M., and Piattini, M. (2009). Una aplicación de la norma iso/iec 15504 para la evaluación por niveles de madurez de pymes y pequeños equipos de desarrollo. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 5(2).
- Mendoza, L. E., Pérez, M. A., and Grimán, A. C. (2005). Prototipo de modelo sistémico de calidad (mosca) del software. *Computación y sistemas*, 8(3):196–217.