



Introducción a la Gestión de Calidad de Software

Dr. Pablo Straub

1 al 16 de septiembre 2008

Presentaciones

- Yo me presento
- Tú te presentas
 - Nombre
 - Actividad/lugar de trabajo
 - Años de experiencia
 - En software o tecnologías de la información
 - Totales
 - Por qué estoy en el diplomado

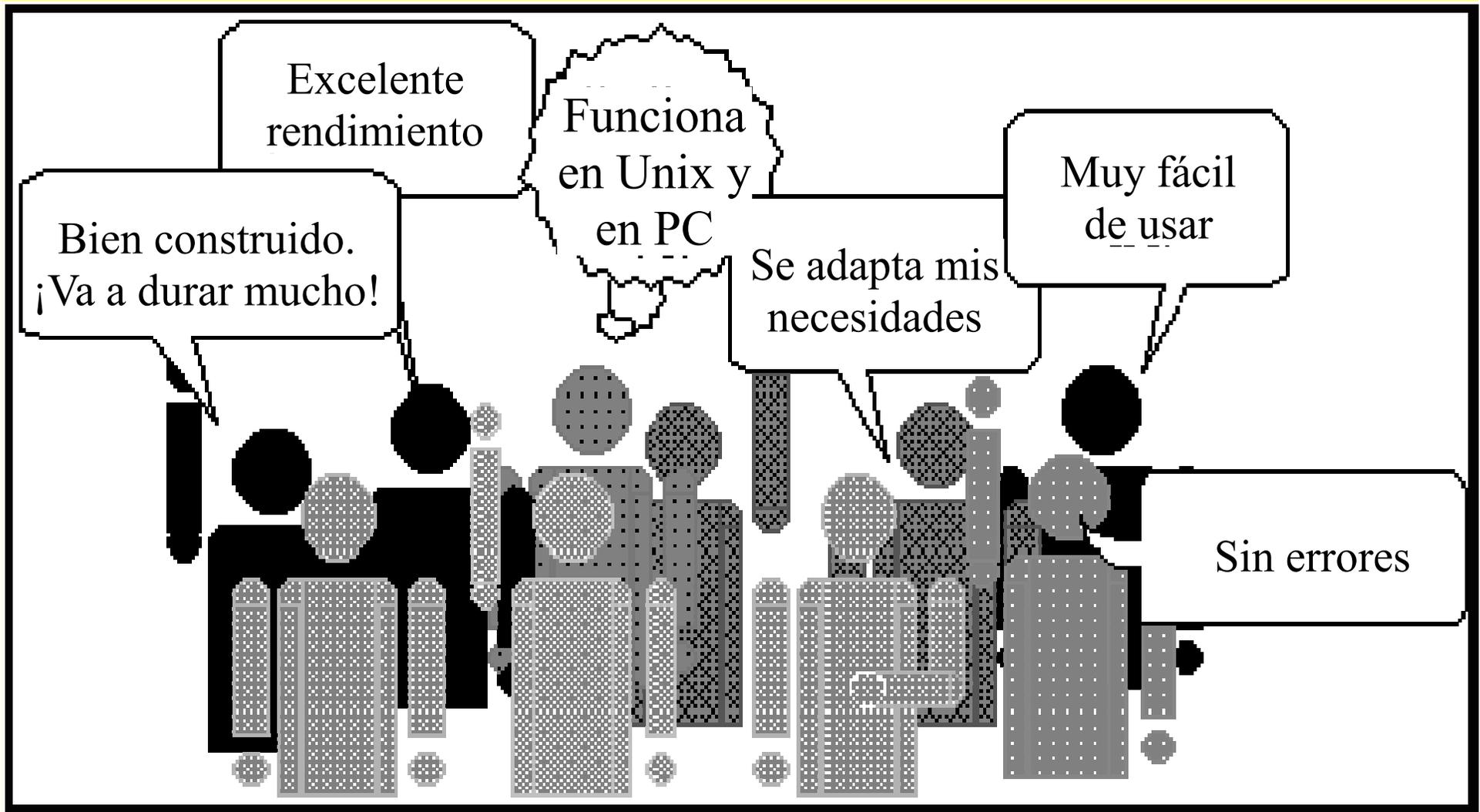


¿Por qué esto es importante?

- ❑ Expectativa “obvia”
 - ❑ Obtener educación continua de alta calidad
 - ❑ En parte escogimos a la Universidad de Chile por su prestigio
- ❑ Los profesores debemos entregarles un producto de calidad
 - ❑ Ustedes tienen expectativas distintas, intereses distintos
 - ❑ Ustedes se satisfacen de modo distinto
 - ❑ Cada uno de ustedes tiene su propia definición de calidad
 - ❑ Y cada profesor tiene su propia definición de calidad
- ❑ ¿Cómo nos ponemos de acuerdo?
- ❑ ¿Cómo aseguramos que entregamos un programa de calidad?



Muchas definiciones de calidad



Objetivos del curso

- ❑ Conocer el concepto de calidad de software y los modelos de calidad
- ❑ Conocer cómo la calidad se integra a las actividades normales de un ciclo de vida
- ❑ Conocer los nuevos enfoques ágiles de desarrollo de software y contrastarlos con los enfoques rigurosos



Contenidos generales

1. Introducción
2. Modelos Rigurosos de Gestión de Calidad
3. Especificación de Requisitos de Calidad y Otros Requisitos
4. Calidad en el Ciclo de Vida y Actividades de Apoyo
5. Desde el Desarrollo Evolutivo a los Métodos Ágiles

Contenidos Detallados

1. Introducción

- Presentación
- Conceptos de Calidad de Software
- Taller: Procesos Críticos

2. Modelos Rigurosos de Gestión de Calidad

- Modelos de Calidad: ISO 9000 y CMMI
- Organización para Calidad
- Ejemplos de Procesos Rigurosos



Contenidos Detallados

3. Especificación de Requisitos de Calidad y Otros Requisitos
 - Requisitos de Calidad versus Otros Requisitos
 - Detalles de la Especificación de Atributos
 - Requisitos de Calidad Importantes
4. Calidad en el Ciclo de Vida y Actividades de Apoyo
 - Calidad en las Actividades del Ciclo vida
 - Calidad en las Actividades de Apoyo
5. Desde el Desarrollo Evolutivo a los Métodos Ágiles
 - Desarrollo Evolutivo
 - Métodos Ágiles
 - Taller: Creación de un Backlog
 - Método Híbrido: Fast V-Model



Universidad de Chile
Departamento de Ciencias de la Computación



Conceptos de Calidad de Software

Principios y definiciones

Por qué calidad

Tipos de problemas

Ideas relevantes sobre la calidad del software

- La calidad es la clave del éxito en el negocio del software
 - Para mejorar la productividad hay que mejorar la calidad
 - La calidad del soporte es tan importante como la calidad del producto
- Para mejorar la calidad del producto hay que mejorar la calidad del proceso
 - La mejora de los procesos debe contar con apoyo completo de la gerencia
 - La gestión es tanto o más importante que la tecnología
- Modelos de calidad como SEI CMM o ISO 9001 son buenos para guiar el proceso y definir una meta objetiva



Principios de calidad de software

- ❑ Prevenga defectos en vez de corregirlos
- ❑ Detecte y corrija los defectos lo antes posible
- ❑ Determine y elimine las causas de los defectos
- ❑ Audite el trabajo en cuanto al uso de procedimientos y estándares



Principios de gestión de software

- ❑ Defina roles, responsabilidades y procedimientos
- ❑ Planifique el trabajo con detalle, a partir de los requisitos de software
- ❑ Haga un seguimiento de la ejecución de los planes y tome medidas correctivas si es necesario
 - ❑ Es malo estar atrasado: es peor no saberlo
- ❑ Refine los planes mientras avanza



Ingeniería de Software

“Es un proceso definido paso a paso que facilita la especificación, el diseño, la realización y las pruebas de una solución de software para un conjunto de requisitos explícitos, de modo eficaz y eficiente.”

- Esto requiere que antes de empezar se tenga
 - objetivos claros
 - planes para lograr los objetivos
 - procedimientos que implementan los planes
 - un ambiente conducente al logro de los objetivos
- Los procesos son fundamentales para lograr calidad repetible



¿Qué es la calidad?

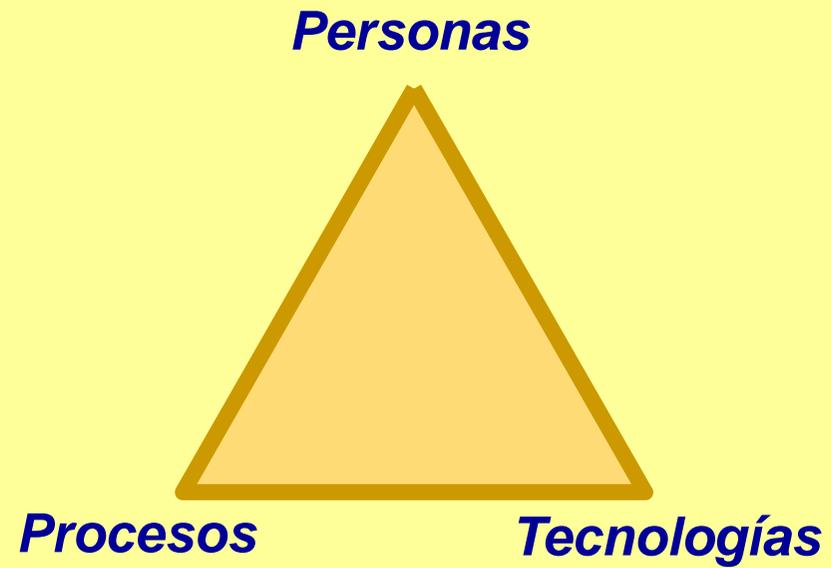
- La totalidad de características y atributos de un producto o servicio relacionados con satisfacer necesidades expresas o implícitas

Calidad = Satisfacción

- La Calidad depende de las expectativas



¿De qué depende la calidad?



¿Por qué calidad?

1. Es un asunto de competitividad
2. Es esencial para sobrevivir
3. Es esencial para exportar
4. Baja los costos
5. Retiene clientes y aumenta utilidades



Calidad, Productividad y Costos

- ❑ En una cocina de restaurante, la calidad suele ser inversamente proporcional a la cantidad
- ❑ En una organización de software, la calidad es proporcional a la productividad
 - ❑ ¿Por qué?
 - ❑ ¿Qué implicancias tiene esto para el negocio?
- ❑ Desglose de costos
 - ❑ Costos básicos de hacer el trabajo
 - ❑ Costos debidos a la calidad
 - Prevención (capacitar, buscar causas de errores)
 - Evaluación (probar, revisar)
 - Fallas (corregir errores, problemas de los clientes)



Las tres fuentes de baja calidad

- ❑ Requisitos erróneos o incompletos del software y/o del soporte
- ❑ Defectos en el software
- ❑ Defectos en el soporte

- ❑ Hay dos tipos de controles de calidad
 - ❑ Verificación
 - ❑ Convalidación

- ❑ Una misma prueba puede contribuir a ambos tipos de controles de calidad



Verificación

- ❑ Consiste en asegurarse que el software corresponde a los requisitos
- ❑ Responde a la pregunta:
 - ❑ ¿Estamos resolviendo el correctamente el problema?
 - ❑ Dice relación con posibles problemas en el proceso de desarrollo
- ❑ Se realiza contrastando documentos entre si
 - ❑ Ejemplo: Se verifica si un diseño cumple los requisitos con una matriz de requisitos versus módulos
- ❑ A veces se puede asegurar la corrección de un programa (¿en qué condiciones?)

Convalidación

- ❑ Consiste en asegurarse que los requisitos corresponden al problema
- ❑ Responde a la pregunta:
 - ❑ ¿Estamos resolviendo el problema correcto?
 - ❑ Dice relación con posibles problemas en la especificación de requisitos
- ❑ Se realiza contrastando un documento con la realidad
 - ❑ Ejemplo: Un requisito se convalida mostrándole la salida que daría el sistema frente a una entrada dada.
- ❑ Es imposible asegurar la corrección de una especificación (¿por qué?)



Definiciones: defecto, error, falla

- ❑ No nos basta hablar de bugs: tenemos que hacer distinciones
- ❑ Definiciones
 - ❑ Los defectos son desviaciones presentes en documentos
 - ❑ Los errores son defectos no detectados
 - ❑ Las fallas son desviaciones presentes durante la ejecución
- ❑ Prevención
 - ❑ Usualmente sólo las fallas son percibidas por los usuarios
 - ❑ Para eliminar fallas hay que eliminar errores
 - ❑ Para eliminar errores hay que
 - prevenir defectos
 - mejorar los controles



Criterios de clasificación de defectos

- ❑ Conviene clasificar los defectos para poder entender dónde nos equivocamos más y así tener un esquema de mejora continua
 - ❑ Fuente - fase del ciclo de vida en que se produce
 - ❑ Detección - fase del ciclo de vida en que se detecta
 - ❑ Gravedad - medida de las consecuencias del defecto
 - Cosmético, menor, funcionalidad limitada, sistema no disponible
 - ❑ Técnico - clasificación de problemas de codificación
 - Inicialización, definición de datos, referencia nula, interfaz, control de flujo, condición, sincronización, etc.
 - ❑ Carácter - acción y/o omisión



El problema de la ingeniería de software: Falta de claridad en los objetivos

- ❑ Si queremos tener éxito debemos definir qué es éxito:
 - ❑ costo
 - ❑ plazo
 - ❑ recursos
 - ❑ nivel de satisfacción de requisitos
 - ❑ funcionalidad implementada
 - ❑ etc.
- ❑ Si los objetivos no son claros no los lograremos claramente



Requisitos vagos, objetivos ausentes

“La funcionalidad del nuevo sistema X es básicamente la misma que la del actual sistema X, sólo que el nuevo X debe ser en general mejor.”

“El nuevo X debe tener una interfaz gráfica basada en Visual Basic (Windows) que lo haga más fácil de usar, en particular sin las dificultades de uso del actual sistema.”

“El sistema tendrá mejores reportes de modo que se pueda aprovechar al máximo la base de datos.”

“El sistema asegurará datos consistentes.”



Si no hay objetivos claros:

- ❑ No podremos demostrar que hemos logrado los objetivos
- ❑ No podemos demostrar que no los hemos logrado
- ❑ No podemos evaluar alternativas de diseños
- ❑ Se terminan especificando medios y no fines
 - ❑ Si hay más de una forma distinta de expresar un objetivo, tal vez no es un objetivo sino un medio



En resumen ...

Para lograr calidad se necesita:

- Definición del cliente de qué es calidad para este proyecto
- Procesos, personas y tecnologías que propenden a la calidad



Referencias

- ❑ Tom Gilb. *Principles of Software Engineering Management*, Addison-Wesley, 1988.
- ❑ Joc Sanders y Eugene Curran. *Software Quality: A framework for success in software development and support*, Addison-Wesley, 1994.
- ❑ Pablo Straub. “El costo de la calidad y el costo de la mala calidad”, 2006, disponible en www.agileshift.cl/Columnas/CostoDeCalidad.pdf





Universidad de Chile
Departamento de Ciencias de la Computación



Proyecto del Curso: Metodología para un Ambiente Real

Proyecto

□ Expectativa

- Se muestra cómo aplicar una metodología en forma efectiva y eficiente en un contexto no necesariamente metódico

□ Alcance

- Situación real
- Posibles métodos a introducir
- Plan de introducción



Contenido

- ❑ Describir un contexto organizacional real
 - ❑ Indicar los objetivos organizacionales
 - ❑ Indicar los métodos usados (y cuánta variabilidad hay en ellos)
 - ❑ Asegurarse de no entregar información confidencial o bien mantener anonimato de la organización
- ❑ Describir principales problemas recurrentes en los proyectos de desarrollo



Contenido

- ❑ Describir uno o más métodos (rigurosos o ágiles) que pudieran ser aplicables
- ❑ Elegir métodos y justificar por qué son aplicables/útiles
- ❑ Explicar cómo se pueden introducir los métodos
- ❑ Plan de cambio incluyendo a lo menos
 - ❑ Aspectos técnicos y culturales
 - ❑ Entrenamiento
 - ❑ Gestión de riesgos



Criterios de evaluación

- ❑ Presentación
- ❑ Redacción
- ❑ Contexto general
- ❑ Objetivos organización
- ❑ Métodos usados
- ❑ **Problemas recurrentes**
- ❑ **Métodos Posibles**
- ❑ **Aplicabilidad**
- ❑ Idea de implantación
- ❑ Aspectos técnicos
- ❑ Aspectos culturales
- ❑ Entrenamiento
- ❑ Gestión de riesgos

