Pruebas de software

Técnicas Estáticas

Temario

TÉCNICAS ESTÁTICAS

1. Revisiones y el Proceso de Pruebas
2. Análisis Estático basado en Herramientas

TÉCNICAS ESTÁTICAS

1. Revisiones y el Proceso de Pruebas

Notación:

Al contrario que las pruebas dinámicas que exigen la ejecución de software, las Pruebas Estáticas se basan: En el examen manual (revisiones) y en el análisis automatizado (análisis estático) del código o de cualquier documentación sin ejecutar código.

Revisiones: Costos y Beneficios

- Costos: El costo de la revisión consiste en 3 tipos principales:
  - Si vamos a tener una revisión, existe el tiempo necesario para realizar revisiones
  - Si estamos realizado una revisión formal, tenemos el esfuerzo necesario para recopilar y analizar métricas
  - Cuando las revisiones son usadas para producir métricas valiosas, esas métricas servirán para realizar mejora de procesos
- Beneficios: Incluyen 4 tipos principales:
  - Cronogramas más cortos (debido a la eliminación eficiente de defectos)
  - Períodos de pruebas más cortos y costos de pruebas mas bajos
  - Debemos observar la productividad de los desarrolladores porque el reproceso es siempre una forma de perdida
  - Debemos ver la calidad del producto mejorada (lo que reduce los costos posteriores) tanto en el desarrollo como después de las versiones.
- En definitiva: las revisiones de toda clase son técnicas probadas de alto retorno para la mejora de la calidad

1. Revisiones y el Proceso de Pruebas

- Técnicas estáticas y proceso de pruebas
- Proceso de revisión
  - Actividades de una revisión formal
  - Funciones y responsabilidades
  - Tipos de revisiones
  - Factores de éxito de las revisiones

1. Revisiones y el Proceso de Pruebas

- Técnicas estáticas y proceso de pruebas
  - Examen manual del código
  - Examen automático sin ejecución
  - Previas a la ejecución
  - Más baratos
  - Detección temprana de fallos
  - También se llaman pruebas de escritorio
  - Objetivo: seguir los flujos de la aplicación.

1. Revisiones y el Proceso de Pruebas

❑Proceso de revisión

Reunión formal en la que se presenta el estado actual de los resultados de un proyecto a un usuario, cliente u otro tipo de persona interesada, y se realiza un análisis estructurado de los mismos.

Las revisiones son el único método de control de calidad eficaz en las fases iniciales del desarrollo a la hora de identificar desviaciones con respecto a las especificaciones de calidad.

1. Revisiones y el Proceso de Pruebas

❑Proceso de revisión

Un Proceso de Revisión Genérico

1. Planificación
2. Inicio
3. Preparación
4. Reunión de la revisión
5. Reproceso/ Reparación
6. Seguimiento

Incluyen la estimación y la planificación, el entrenamiento de los participantes, asignación de roles, definición, la selección de las partes de los documentos a ser revisados, etc.

Estas actividades se repiten por cada ítem revisado. La preparación mediante la comprobación de los documentos y la observación de los defectos, las preguntas y los comentarios potenciales. El trabajo de la preparación es usualmente de una a dos horas. La reunión es de una a dos horas seguidas. El reproceso y la reparación se refiere a la corrección de los defectos encontrados.

El seguimiento incluye el análisis del mejoramiento de los ítems individuales así como también del proceso completo, la evaluación de la eliminación de lo defectos en la fase de salida de las revisiones (reuniones de salida), etc.

Los detalles del proceso de revisión dependen del tipo de revisión especifico utilizado en el proyecto

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Actividades de una revisión formal
    - Planificar
    - Definir criterios de entradas y salidas
    - Inicio
    - Comprobar criterios de entrada
    - Preparación individual

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión (II)
  - Actividades de una revisión formal
    - Prestar atención a defectos y preguntas
    - Examen y evaluación de resultados
    - Adaptar
    - Corregir defectos
    - Seguimiento
    - Comprobar criterios de salida

1. Revisiones y el Proceso de Pruebas

- **Proceso de revisión**
- **Funciones y responsabilidades**
  - **Jefe:** Planifica, organiza los recursos y la capacitación, etc.
  - **Moderador:** Lidera las reuniones de revisión.
  - **Autor:** Describe, explica, responde a las preguntas acerca del ítem.
  - **Revisores:** Encuentra defectos (bugs) en el ítem.
  - **Registrador (Escribano):** recopila información acerca de los hallazgos.

En algunos casos, una persona puede desempeñar varias funciones.

1. Revisiones y el Proceso de Pruebas

- **Proceso de revisión**
  - **Abordar los productos de software o productos de trabajo asociados desde distintas perspectivas y utilizar listas de comprobación puede contribuir a la efectividad y eficiencia de las revisiones**

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Tipos de revisiones
    - Informal
    - Guiada
    - Técnica
    - Inspección

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Tipos de revisiones
    - Informal
      - Ausencia de proceso formal
      - Puede adoptar distintas formas
      - Los resultados se pueden documentar
      - Su utilidad depende de los revisores
      - Objetivo principal: forma barata de revisar

No hay un proceso real (charlas de pasillo, pruebas entre amigos, programación por pares), tienen efectividad limitada en la eliminación de los defectos, pero son todavía útil, económico, popular.

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Tipos de revisiones
    - Guiada
      - Liderada por el autor
      - Distintos escenarios: simulacro, reunión, etc…
      - Sesiones abiertas
      - Registrador opcional (distinto del autor)
      - Puede variar en grado de formalidad
      - Objetivos: Aprender, entender y encontrar defectos

El autor "guía la revisión" del ítem de revisión. Puede incluir escenarios, ensayos y participación de grupos de compañeros.

1. Revisiones y el Proceso de Pruebas

❑Proceso de revisión

❑Tipos de revisiones

❑Guiada – Recorrido (Walkthrouhgs)

Objetivos

- Detectar defectos
- Examinar alternativas
- Aprender

Problemas

- No se decide QUE HACER con los defectos encontrados en la revisión de requerimientos y en la revisión de diseño
- Hay un presentador que acompaña los desarrolladores y él no conoce a fondo el producto
- Los asistentes son especialistas del negocio, de la tecnología usada o conocedores de los sistemas donde hay impacto.
- No preparan esta actividad
- El autor es el único que decide que hacer con lo que se encuentra

1. Revisiones y el Proceso de Pruebas

- **Proceso de revisión**
  - **Tipos de revisiones**
    - **Técnica**
      - **Proceso documentado y definido para detectar defectos**
      - **Participación de la dirección opcional**
      - **Dirigida por un moderador (distinto del autor)**
      - **Preparación previa por parte de los revisores**
      - **Uso opcional de listas de comprobación**

Proceso de eliminación de defectos documentados y definidos, que involucra técnicos expertos pero no jefes.

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Tipos de revisiones
    - Técnica (II)
      - Informe de revisión
      - El grado de formalidad puede variar
      - Objetivos: Debatir, tomar decisiones, evaluar alternativas, encontrar defectos, resolver problemas técnicos y comprobar la conformidad con las especificaciones, los planes, la normativa y los estándares

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Tipos de revisiones
    - Inspección
      - Dirigida por un moderador formado (distinto del autor)
      - Celebrada como un examen
      - Funciones definidas
      - Incluye recopilación de métricas
      - Proceso formal basado en normas y listas de comprobación

Es el método mas formal. Un moderador entrenado (diferente del autor) lidera el equipo de inspección formal (reglas, listas de comprobación, criterios de entrada y salida), los cuales incluyen la recolección de métricas de eliminación de defectos.

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Tipos de revisiones
    - Inspección (II)
      - Criterios de entrada y salida especificados para la aceptación del software
      - Preparación previa de la reunión
      - Informe de inspección
      - Seguimiento formal
      - Objetivo principal: identificar defectos

1. Revisiones y el Proceso de Pruebas

❑Proceso de revisión

❑Tipos de revisiones

❑Inspección (III)

"Técnica de evaluación formal: requisitos de software, diseño o codificación se examinan en detalle por una persona o grupo, distintos del autor, para detectar defectos, disconformidades con las normas de desarrollo y otros problemas" (IEEE,1990).

Reglas para garantizar el éxito de una inspección:

- Inspeccionar toda clase posible de defectos
- Participación de personas de todos los niveles (no dirección)
- Etapas predefinidas estrictamente
- Reuniones no superiores a 2h
- Moderadores y directores de la inspección expertos
- Cada miembro tiene funciones detalladas y específicas
- Lista de comprobación, con preguntas a realizar
- Archivar estadística de defectos para posterior análisis

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Tipos de revisiones
    - Inspección (IV)

Mini ejemplo Lista de Comprobación o Lista de Chequeo (Rev. de Diseño)

¿Uniformidad en el diseño?

¿Interfaces entre módulos definidas correctamente

¿Interfaces externas definidas correctamente?

¿El diseño cubre todas las funciones de la especificación de requisitos?

¿El diseño cumple todos los requisitos no funcionales?

¿Se ha aplicado la notación de diseño correctamente?

¿ La documentación del diseño es ambigua?

¿Diseño suficientemente detallado para implementarlo en el lenguaje elegido?

1. Revisiones y el Proceso de Pruebas

❑Proceso de revisión

❑Tipos de revisiones

❑Inspección vs Recorrido

Propiedades	Inspección	Recorrido
Entrenamiento formal del moderador	SI	NO
Roles definidos para participantes	SI	NO
Guía de la revisión	Moderador	Propietario del producto
Se usan listas de comprobación	SI	NO
Se reportan los errores y fallas distribuidos por tipo	SI	NO
Seguimiento para controlar la corrección	SI	NO
Se puede mejorar eficiencia de la revisión (análisis de resultados)	SI	NO

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Factores de éxito de las revisiones
    - Objetivos previos y claros
    - Personal adecuado y preparado
    - Objetividad
    - Tacto a la hora de comunicar fallos
    - Clima de confianza

1. Revisiones y el Proceso de Pruebas

- Proceso de revisión
  - Factores de éxito de las revisiones (II)
    - Técnicas de revisión adecuadas
    - Listas de funciones y comprobación
    - Formación si se precisara
    - Apoyo gerencial
    - Aprendizaje y mejora continua

1. Revisiones y el Proceso de Pruebas

Resumen Proceso de revisión (I)

Las revisiones guiadas, las revisiones técnicas o inspecciones pueden realizadas por grupos de colegas. P, ej. Colegas en el mismo nivel. Este tipo de revisión es denominada revisión entre pares o colegas (Peer review)

1. Revisiones y el Proceso de Pruebas

Sugerencias para revisiones exitosas

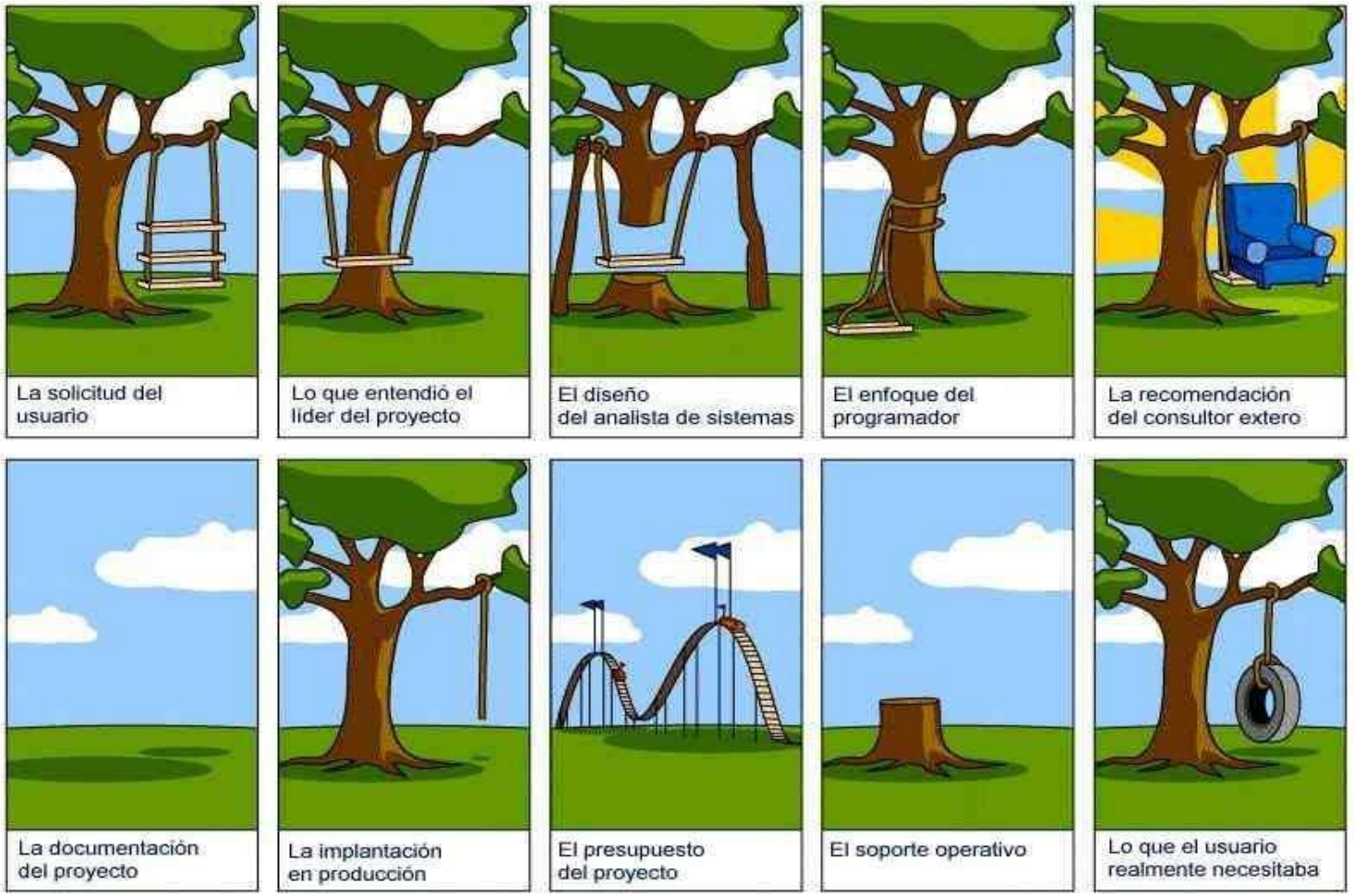
- Proporcionar capacitación
- Revisar el producto, no el productor
- Establecer y seguir la agenda y los objetivos
- Limitar el debate
- Concentrarse en los problemas de hallazgo, no en los de correcciones
- Tomar notas escritas
- Limitar y seleccionar cuidadosamente a los participantes
- Insistir en la preparación
- Incluir probadores
- Elaborar una lista de comprobación para cada tipo del ítem que es revisado
- Revisar las revisiones
- Utilizar las técnicas correctas
- Garantizar el apoyo de la gerencia
- No utilizar mal los hallazgos
- ¡Aprender y mejorar!

REVISIÓN DE REQUERIMIENTOS

1. INTRODUCCIÓN
2. PROBLEMAS COMUNES
3. CARACTERÍSTICAS DE UN REQUERIMIENTO
4. BENEFICIOS DE LA GESTIÓN DE REQUERIMIENTOS
5. CICLO DE VIDA DE LOS REQUERIMIENTOS
6. DEFINICIÓN DE REQUERIMIENTOS
7. ESPECIFICACIÓN DE REQUERIMIENTOS
8. ADMINISTRACIÓN DE REQUERIMIENTOS
9. TRABAJO A REALIZAR

INTRODUCCIÓN

“Se que crees que comprendes lo que piensas que he dicho, pero no estoy seguro de que lo que creíste oír sea lo que yo quise decir”



INTRODUCCIÓN

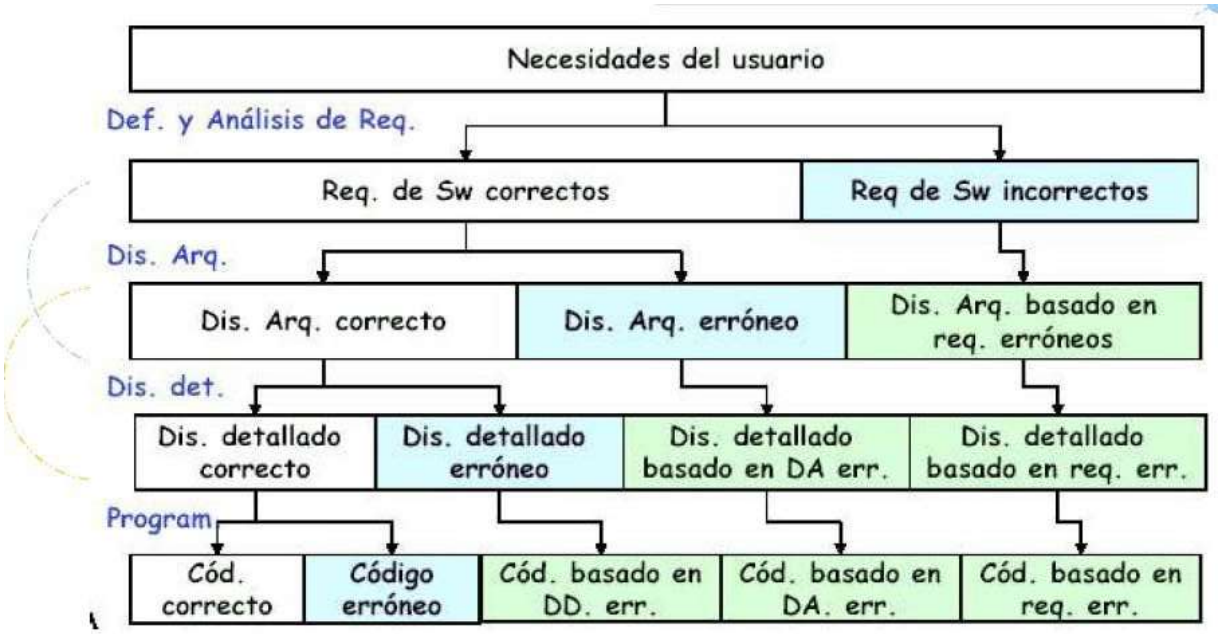
Cuando las personas realizan su trabajo diario detectan que parte de este puede ser automatizado o mejorado, ahí nace el "Requerimiento".

¿Qué son Requerimientos? de las muchas definiciones que existen, a continuación se presenta la definición que da el glosario de la IEEE:

- (1) Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

INTRODUCCIÓN

Se puede decir que la **Gestión de Requerimientos** es una actividad fundamental en el proceso de Producción de Software, ya que se enfoca en la definición de lo que se desea producir, sin embargo también se han identificado varias dificultades que afectan notablemente el proceso de desarrollo de Software.



INTRODUCCIÓN

Porque los proyectos de Software son Exitosos?

Involucra a Usuarios	15.9%
Soporte Administración	13.9%
Clara definición de Requerimientos	13.0%
Apropiado Planeamiento	9.6%
Expectativas Realistas	8.2%
Hitos no Extensos	7.7%
Staff Competente de profesionales	7.2%
Propietario	5.3%

Fuente: Quality Systems & Software - 1997

INTRODUCCIÓN

Porque los proyectos de Software son fallan?

Requerimientos Incompletos	13.1%
Falta de Requerimientos	12.4%
Falta de Recursos	10.6%
Expectativas no Realistas	9.9%
Cambio Requerimientos/Especificaciones	8.7%
Falta de Planeamiento	8.1%
No se especifico el tiempo adecuado	7.5%

Fuente: Quality Systems & Software - 1997

PROBLEMAS COMUNES

- Los requerimientos normalmente no son obvios y provienen de muchas fuentes.
- Son difíciles de expresar en palabras (el lenguaje es ambiguo).
- Existen muchos tipos de requerimientos y diferentes niveles de detalle.
- La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
- Nunca son iguales. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.

PROBLEMAS COMUNES

- Los requerimientos están relacionados unos con otros, y a su vez se relacionan con otras partes del proceso.
- Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas.
- Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
- Son difíciles de cuantificar, ya que cada conjunto de requerimientos es particular para cada proyecto.

CLASIFICACION DE LOS REQUERIMIENTOS

Los requerimientos pueden dividirse en:

Requerimientos Funcionales

- Definen las funciones que el sistema será capaz de realizar.
- Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Requerimientos No-Funcionales

- Tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), la usabilidad de las interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

CARACTERISTICAS DE UN REQUERIMIENTO

IEEE 830

- ✓ **Necesario**: Operativamente se requiere
- ✓ **No Ambiguo**: Interpretación clara
- ✓ **Verificable**: Soporte, Evidencias
- ✓ **Completo**: Información no fácil de detectar
- ✓ **Correcto**: Validar la fuente del mismo
- ✓ **Viable**: Limitaciones técnicas y económicas
- ✓ **Priorizable**: Justificación
- ✓ **Consistente**: Información no fácil de detectar

CARACTERISTICAS DE UN REQUERIMIENTO

☑ **Necesario**:	Operativamente se requiere
☑ **No Ambiguo**:	Interpretación
☑ **Verificable**:	Soportes
☑ **Completo**:	Información no fácil de detectar
☑ **Correcto**:	Validar la fuente del mismo.
☑ **Viable**:	Limitaciones técnicas y económicas
☑ **Priorizable**:	Justificación
☑ **Consistente**:	Información no fácil de detectar

NECESARIO

- Cada requerimiento debería documentar algo que el usuario realmente necesite
- Algo que es requerido por un estándar, o por variables externas indispensables
- Es necesario aquel requerimiento que forma parte de un contrato firmado y establecido.
- Una manera de identificar requerimientos necesarios, es si la fuente del requerimiento viene de una fuente CON AUTORIDAD para especificar requerimientos.
- Al revisar un requerimiento, navegue hacia atrás en este requerimiento (la fuente) y valide si es o no necesario

CARACTERISTICAS DE UN REQUERIMIENTO

- ☑ **Necesario**: Operativamente se requiere
- ☑ **No Ambiguo**: Interpretación
- ☑ **Verificable**: Soportes
- ☑ **Completo**: Información no fácil de detectar
- ☑ **Correcto**: Validar la fuente del mismo.
- ☑ **Viable**: Limitaciones técnicas y económicas
- ☑ **Priorizable**: Justificación
- ☑ **Consistente**: Información no fácil de detectar

NO AMBIGUOS

- Un requerimiento es “no ambiguo” si todos los lectores del requerimiento pueden llegar a una simple y consistente interpretación “igual” del mismo

- Escribir los requerimientos en lenguaje natural, y no en TÉRMINOS COMPUTACIONALES, ayuda que un requerimiento pueda ser interpretado y entendido sin ambigüedades.

- Una forma efectiva para revelar ambigüedad es hacer revisiones de los documentos, diagramas, casos de uso, desarrollar prototipos, pintar borradores de interfaces y revisar escenarios de cada uno de ellos.

CARACTERISTICAS DE UN REQUERIMIENTO

NO AMBIGUOS – Ejemplo

- “El sistema debe permitir firmar un formato electrónico.”

- “El sistema debe permitir firmar los formatos electrónicos, adjuntando una firma previamente escaneada del usuario en sesión, a los formatos electrónicos seleccionados.”

CARACTERISTICAS DE UN REQUERIMIENTO

- ☑ **Necesario**: Operativamente se requiere
- ☑ **No Ambiguo**: Interpretación
- ☑ **Verificable**: Soportes
- ☑ **Completo**: Información no fácil de detectar
- ☑ **Correcto**: Validar la fuente del mismo.
- ☑ **Viable**: Limitaciones técnicas y económicas
- ☑ **Priorizable**: Justificación
- ☑ **Consistente**: Información no fácil de detectar

VERIFICABLES

- Trate de aplicar métodos formales, para realizar pruebas que confirmen que el requerimiento está correctamente especificado

- Debería poderse contar con herramientas que, al aplicárselas al requerimiento, puedan demostrarse que esté está correctamente especificado.

- Se puede utilizar, soportes, libros, documentos legales, para verificar el requerimiento.

CARACTERISTICAS DE UN REQUERIMIENTO

VERIFICABLES - Ejemplo

- A partir de producido el evento X, se obtendrá el output del proceso con la mayor cantidad de transacciones en el menor tiempo posible.

CARACTERISTICAS DE UN REQUERIMIENTO

VERIFICABLES - Ejemplo

- A partir de producido el evento X, se obtendrá el output del proceso con la mayor cantidad de transacciones en el menor tiempo posible.

- A partir de producido el evento X, se obtendrá el output del proceso: debe estar dentro de los 20 segundos el 60% de las transacciones y dentro de los 30 segundos el 99% de las transacciones

CARACTERISTICAS DE UN REQUERIMIENTO

- ☑ **Necesario**: Operativamente se requiere
- ☑ **No Ambiguo**: Interpretación
- ☑ **Verificable**: Soportes
- ☑ **Completo**: Información no fácil de detectar
- ☑ **Correcto**: Validar la fuente del mismo.
- ☑ **Viable**: Limitaciones técnicas y económicas
- ☑ **Priorizable**: Justificación
- ☑ **Consistente**: Información no fácil de detectar

COMPLETOS

- Requerimientos mal especificados podrían esconder información que no es fácil de detectar

- Puede ayudar a no tener requerimientos incompletos, enfocarse en conocer las tareas del usuario y no sesgarse en las funciones de un sistema

- Si usted sabe que tiene información que falta algo por conocer, utilice estrategia de “marcas” o “banderas”, lo cual generen tareas “PENDIENTES POR DETERMINAR”.

- Estas tareas deberán ser trabajadas y maduradas en los procesos de validación con el usuario y en cada iteración profundizar para encontrar su completitud.

CARACTERISTICAS DE UN REQUERIMIENTO

COMPLETOS - Ejemplo

- “El sistema debe permitir consultar la información de las iniciativas para generar proyectos”

- “El sistema debe permitir a los investigadores, docentes, estudiantes, egresados, empleados y Directivos de la Universidad, consultar la información de las iniciativas para generar proyectos de Investigación”

CARACTERISTICAS DE UN REQUERIMIENTO

- ☑ **Necesario**: Operativamente se requiere
- ☑ **No Ambiguo**: Interpretación
- ☑ **Verificable**: Soportes
- ☑ **Completo**: Información no fácil de detectar
- ☑ **Correcto**: Validar la fuente del mismo.
- ☑ **Viable**: Limitaciones técnicas y económicas
- ☑ **Priorizable**: Justificación
- ☑ **Consistente**: Información no fácil de detectar

CORRECTOS

- Cada requerimiento especifica una funcionalidad o condición que debe contener el software
- No podemos contar con interpretaciones incorrectas de funcionalidades a implantar
- El punto de referencia para validar si un requerimiento es correcto o no, es la fuente del mismo. El usuario directamente, o la documentación de alto nivel que originó el requerimiento.
- Un usuario representativo, puede determinar si el requerimiento es correcto o no. Por ello es esencial incluir los usuarios durante el proceso de revisión de los requerimientos.

CARACTERISTICAS DE UN REQUERIMIENTO

☑ **Necesario**: Operativamente se requiere
☑ **No Ambiguo**: Interpretación
☑ **Verificable**: Soportes
☑ **Completo**: Información no fácil de detectar
☑ **Correcto**: Validar la fuente del mismo.
☑ **Viable**: Limitaciones técnicas y económicas
☑ **Priorizable**: Justificación
☑ **Consistente**: Información no fácil de detectar

VIABLES

- Es mas fácil implementar requerimientos cuando se conocen limitaciones técnicas, la capacidad, y las condiciones del ambiente que rodea el proyecto

- Para detectar requerimientos “NO VIABLES” es importante incluir dentro del equipo de análisis de requerimientos un miembro de la parte técnica o de ingeniería, como un miembro del dominio del producto o mercado.

- Este tipo de personas, pueden ayudar a determinar de una manera real, que es posible técnicamente o no, o que es excesivamente costoso de implementar

CARACTERISTICAS DE UN REQUERIMIENTO

VIABLES - Ejemplo

- “Capacidad Financiera”, solicitamos que la Razón Corriente exigida sea 2.5 veces. Igualmente en el mismo Ítem apreciamos que el Nivel de Endeudamiento es muy alto (50%) lo cual implica que la mitad de la Empresa pertenece a terceras personas; solicitamos se considere un endeudamiento no superior al 30%.”

- Este requerimiento no es viable por cuanto los parámetros para hallar la capacidad financiera, de Razón Corriente, Endeudamiento y Capital de Trabajo son indicadores estándar para medir el estado financiero de las empresas, por tal motivo no pueden ser modificables.

CARACTERISTICAS DE UN REQUERIMIENTO

☑ **Necesario**:	Operativamente se requiere
☑ **No Ambiguo**:	Interpretación
☑ **Verificable**:	Soportes
☑ **Completo**:	Información no fácil de detectar
☑ **Correcto**:	Validar la fuente del mismo.
☑ **Viable**:	Limitaciones técnicas y económicas
☑ **Priorizable**:	Justificación
☑ **Consistente**:	Información no fácil de detectar

QUE SE PUEDAN PRIORIZAR

- Si todos los requerimientos tiene el mismo nivel de prioridad, hay una mala identificación y especificación de requerimientos

- Cada requerimiento o un conjunto de ellos debe poderse distribuir en los diferentes *releases* de liberación de software o producto.

- Si todos los requerimientos son de alta prioridad, el equipo de desarrollo no tendrá facilidad para incluir o modificar nuevos requerimientos durante la etapa de desarrollo

CARACTERISTICAS DE UN REQUERIMIENTO

☑ **Necesario**:	Operativamente se requiere
☑ **No Ambiguo**:	Interpretación
☑ **Verificable**:	Soportes
☑ **Completo**:	Información no fácil de detectar
☑ **Correcto**:	Validar la fuente del mismo.
☑ **Viable**:	Limitaciones técnicas y económicas
☑ **Priorizable**:	Justificación
☑ **Consistente**:	Información no fácil de detectar

CONSISTENTES

- Si un requerimiento tiene conflicto con otro requerimiento de software, este requerimiento o ambos tiene problemas en la especificación.
  - No se contradice con otro

- Los requerimientos deben estar acordes con las reglas del negocio y con las variables externas que afectan el dominio del negocio

BENEFICIOS DE LA GESTION DE REQUERIMIENTOS

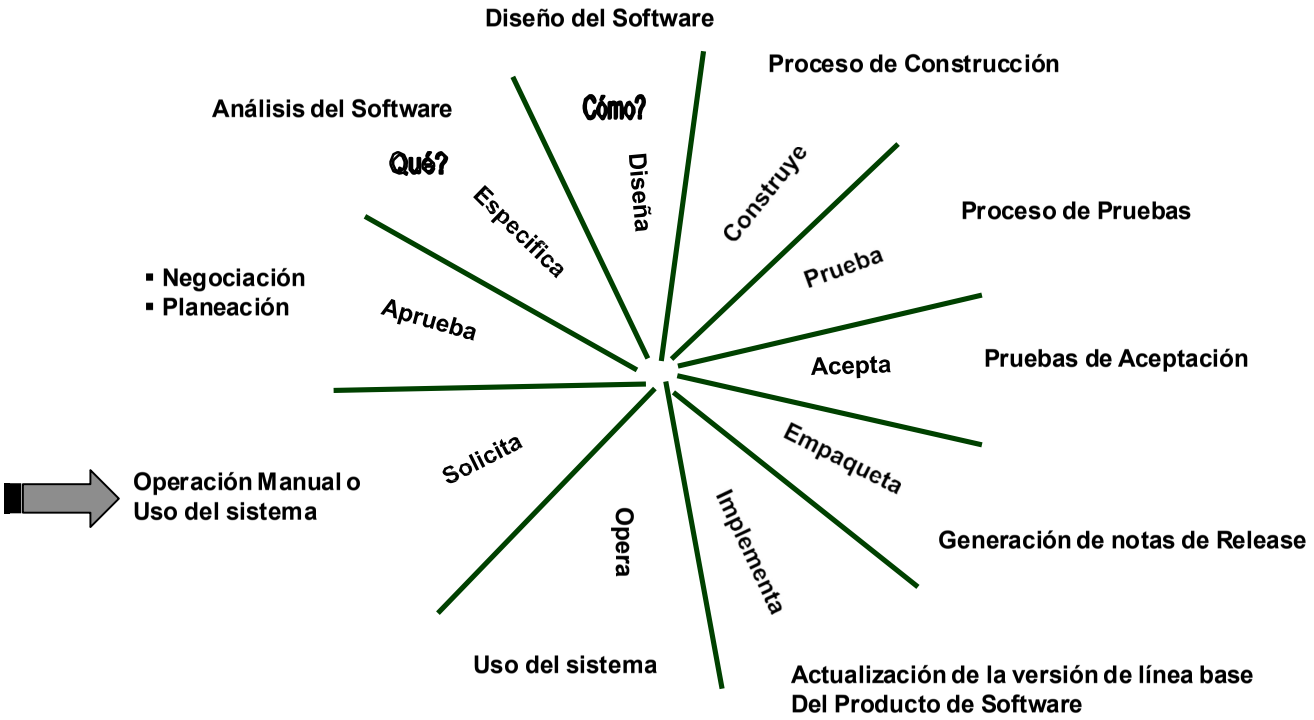
- Mejor control de proyectos complejos.
- Facilita la conformidad con estándares y regulaciones.
- Permite gestionar las necesidades del proyecto en forma organizada.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La GR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto.
- Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).

BENEFICIOS DE LA GESTION DE REQUERIMIENTOS

- Mejora la comunicación entre equipos: La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.

- Evita rechazos de usuarios finales: Cuando esta situación aplica, la gestión de requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.

CICLO DE VIDA DE LOS REQUERIMIENTOS



PROVEEDORES DE REQUERIMIENTOS

Identificar los elementos válidos como fuente de información para identificar los requerimientos:

1. Clientes Piloto
   - Usuario Líder
   - Usuario Final
2. Clientes Potenciales
3. Otras Aplicaciones Software
4. Expertos arquitectos en aplicaciones semejantes

- Validar con los proveedores la validez del requerimiento

DEFINICION DE LOS REQUERIMIENTOS

Redacción de las necesidades que deben hacer parte de la solución de Software:

MODELO REDACCIÓN DE REQUERIMIENTOS (COMPLETITUD)

[LUGAR, TIEMPO, EVENTO, OBJETO] +

“debe, deberá, no debe, no deberá” +

[ACCIÓN, VERBO, SENTENCIA] +

[A QUIEN, SUJETO] [RESULTADO, CONSECUENCIA]. +

DEFINICION DE LOS REQUERIMIENTOS

Algunos ejemplos para el modelo:

- “El sistema en el módulo de ventas debe ofrecer una opción mediante la cual el usuario pueda ver una comparación de lo presupuestado versus la venta real en un rango de fechas”

- “Cuando el tiempo de conexión exceda el valor predeterminado como máximo el sistema debe cancelar la sección informándole a el usuario que el tiempo se agotó y que por lo tanto será desconectado”

El listado de los requerimientos del software, se deberían mantener en un repositorio Central.

DEFINICION DE LOS REQUERIMIENTOS

Algunos ejemplos para el modelo:

- "El sistema en el módulo de ventas debe ofrecer una opción mediante la cual el usuario pueda ver una comparación de lo presupuestado versus la venta real en un rango de fechas"

- "Cuando el tiempo de conexión exceda el valor predeterminado como máximo el sistema debe cancelar la sección informándole a el usuario que el tiempo se agotó y que por lo tanto será desconectado"

[LUGAR, TIEMPO, EVENTO, OBJETO]

DEFINICION DE LOS REQUERIMIENTOS

Algunos ejemplos para el modelo:

“El sistema en el módulo de ventas debe ofrecer una opción mediante la cual el usuario pueda ver una comparación de lo presupuestado versus la venta real en un rango de fechas”

“Cuando el tiempo de conexión exceda el valor predeterminado como máximo el sistema debe cancelar la sección informándole a el usuario que el tiempo se agotó y que por lo tanto será desconectado”

[LUGAR, TIEMPO, EVENTO, OBJETO] +
“debe, deberá, no debe, no deberá”

Algunos ejemplos para el modelo:

“El sistema en el módulo de ventas debe ofrecer una opción mediante la cual el usuario pueda ver una comparación de lo presupuestado versus la venta real en un rango de fechas”

“Cuando el tiempo de conexión exceda el valor predeterminado como máximo el sistema debe cancelar la sección informándole a el usuario que el tiempo se agotó y que por lo tanto será desconectado”

[LUGAR, TIEMPO, EVENTO, OBJETO] + “debe, deberá, no debe, no deberá” + [ACCIÓN, VERBO, SENTENCIA]

DEFINICION DE LOS REQUERIMIENTOS

Algunos ejemplos para el modelo:

“El sistema en el módulo de ventas debe ofrecer una opción mediante la cual el usuario pueda ver una comparación de lo presupuestado versus la venta real en un rango de fechas”

“Cuando el tiempo de conexión exceda el valor predeterminado como máximo el sistema debe cancelar la sección informándole a el usuario que el tiempo se agotó y que por lo tanto será desconectado”

[LUGAR, TIEMPO, EVENTO, OBJETO] +
“debe, deberá, no debe, no deberá” +
[ACCIÓN, VERBO, SENTENCIA] +
[A QUIEN, SUJETO] +

DEFINICION DE LOS REQUERIMIENTOS

Algunos ejemplos para el modelo:

"El sistema en el módulo de ventas debe ofrecer una opción mediante la cual el usuario pueda ver una comparación de lo presupuestado versus la venta real en un rango de fechas"

"Cuando el tiempo de conexión exceda el valor predeterminado como máximo el sistema debe cancelar la sección informándole a el usuario que el tiempo se agotó y que por lo tanto será desconectado"

[LUGAR, TIEMPO, EVENTO, OBJETO] +
"debe, deberá, no debe, no deberá" +
[ACCIÓN, VERBO, SENTENCIA] +
[A QUIEN, SUJETO] +
[RESULTADO, CONSECUENCIA].

HERRAMIENTAS PARA GESTION DE REQUERIMIENTOS

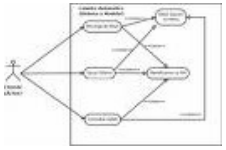
Id	Código	Descripción corta	Organización	Producto/Servicio	Módulo	Fecha Creación	Usuario Inicial	Edad	Fecha Modificación	Usuario Remite	Estado
1	1094	Filtros en Auditoria de Sistema	GreenSQA	GreenNotes [2.0.0]	Configuración	2005/12/14	phoyos	40	2005-12-30 00:00:00	phoyos	Desarrollo
2	1096	Adjuntar Graficos	GreenSQA	GreenNotes [2.0.0]	Gestión de Notas	2005/12/27	phoyos	27	2005-12-30 00:00:00	phoyos	Desarrollo
3	968	Registro de Procesos Relacionados	GreenSQA	GreenNotes [2.0.0]	Gestión de Notas	2005/08/16	phoyos	160	2005-12-30 00:00:00	phoyos	Desarrollo
4	969	Registro de Objetos modificados en un release	GreenSQA	GreenNotes [2.0.0]	Gestión de Notas	2005/08/16	phoyos	160	2005-12-30 00:00:00	phoyos	Desarrollo
5	1045	Buscador para Documentos Release	GreenSQA	GreenNotes [2.0.0]	Buscador	2005/11/29	phoyos	55	2005-12-30 00:00:00	phoyos	Desarrollo

ESPECIFICACION DE LOS REQUERIMIENTOS

Detalle de la definición de requerimientos donde se especifican las entradas, el proceso, las salidas y las restricciones.

Técnicas

- Entrevistas y Cuestionarios
- Lluvia de Ideas
- Prototipos
- Casos de Uso



Se deben especificar tanto los Requerimientos Funcionales como los No Funcionales

ADMINISTRACION DE LOS REQUERIMIENTOS

- Seguimiento y Control en el ciclo de vida de desarrollo del software, de las actividades realizadas sobre los requerimientos del sistema.

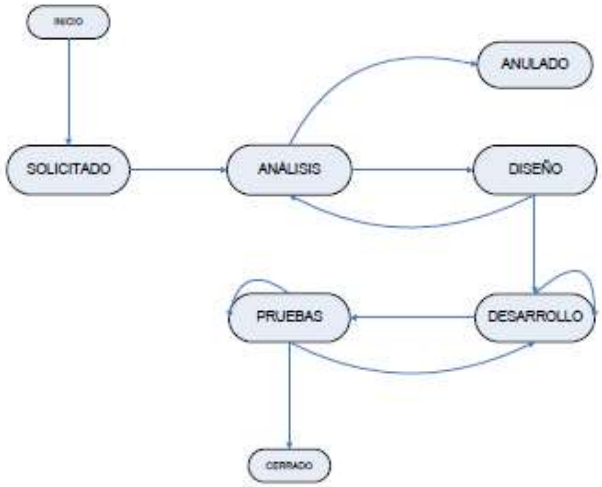
- Priorización de Requerimientos – Matriz de Priorización (Criticidad, Importancia, Complejidad, Riesgo).

Requerimientos	Criterios Priorización			Prioridad
	Criticidad	Importancia	Complejidad	
Req 1				0
Req 2				0
Req n				0

Escalas	Criterios		
	Criticidad	Importancia	Complejidad
Alta			
Media			
Baja			

ADMINISTRACION DE LOS REQUERIMIENTOS

Trazabilidad de Requerimientos - Mediante un Repositorio Central de Requerimientos.



De esta forma, se puede conocer el estado del proyecto en cualquier momento o intervalo de tiempo.

ENTREGABLES EN LA GESTION DE REQUERIMIENTOS

- Mapa Conceptual del modelo de negocio que soporta la aplicación (Opcional)
- Matriz de descomposición funcional del producto (Opcional)
- Repositorio Central de Requerimientos. (Requerido)
- Especificación de Requerimientos (SRS) (Casos Uso o Formato Requerimientos) (Requerido)
- Plan de Desarrollo de Requerimientos (Requerido)
- Matriz Trazabilidad de Requerimientos (Requerido)

REALIZAR LECTURA SOBRE ESPECIFICACIÓN DE REQUERIMIENTOS EN ARTICULOS Y LIBROS Y HACER RESUMEN DE LAS CARACTERISTICAS DE LOS REQUERIMIENTOS Y SU IMPORTANCIA

TRABAJO

REVISION DE DISEÑO Y ARQUITECTURA

OBJETIVO DE LA REVISIÓN DE DISEÑO

Detectar e identificar no conformidades en el diseño antes de pasar a la codificación, así como también identificar aspectos de mejoramiento.

Entre otros, en esta actividad se verifica la arquitectura y utilización de patrones en el diseño

CRITERIOS A EVALUAR

- **Arquitectura de Software y Hardware**
  - Se selecciona y diseña con base en unos objetivos y restricciones.
  - Los objetivos son prefijados para el desarrollo del sistema.
  - Las restricciones son limitaciones derivadas de las tecnologías disponibles para implementar el sistema.
  - Se pueden identificar las diferentes vistas en la aplicación?

- **Uso de Patrones de Diseño**
  - Tiene una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado (bibliotecas, script, otro programa, etc) para ayudar a desarrollar y unir los diferentes componentes de un proyecto.
  - Patrones de interfaces de usuario, Patrón creador, etc

CRITERIOS A EVALUAR

- **Flexibilidad del software**
  - Que tan configurable es el software de tal manera que permita adaptarse a diferentes condiciones del modelo de negocio.
- **Interoperabilidad**
  - Capacidad del producto software para interactuar con uno o más sistemas especificados. SSO, Integración, etc
- **Mantenibilidad**
  - La capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación al software por cambios en el ambiente, o en requisitos y/o en las especificaciones funcionales. (Capacidad de ser analizado, Capacidad para ser cambiado, Estabilidad, etc)

CRITERIOS A EVALUAR

- **Fiabilidad**
  - La capacidad del producto de software para mantener un nivel especificado de desempeño cuando está siendo utilizado bajo condiciones especificas. (Madurez, Tolerancia a fallos, Capacidad de recuperación, etc).
  - Mecanismo de seguridad para el manejo de la información

- **Trazabilidad**
  - ¿Se tienen políticas internas para administración, actualización y control de versiones de paquetes de software?
  - ¿Se evidencia en el diseño un mecanismo de registro de errores (log)?

CRITERIOS A EVALUAR

- **Eficiencia**
  - La capacidad del producto de software para proporcionar el desempeño apropiado, coherente a la cantidad de recursos usados, bajo condiciones indicadas.(Comportamiento temporal , Utilización de recursos, etc)

- **Portabilidad**
  - La capacidad del producto de software para ser transferido de un ambiente a otro.

- **Dimensionamiento**
  - Cantidad de usuarios a utilizar la aplicación
  - Espacio en disco para la operación de la aplicación

Defectos Comunes de Requisitos y Diseño

- Ambigüedades: ¿Qué significa eso exactamente?
  - P. ej.: El sistema permitirá a los usuarios leer el correo electrónico del ISP
  - ¿Cuáles ISPs? ¿Qué tamaño de e-mails? ¿Archivos adjuntos?
- Incompletitud: Bueno, ¿Y después qué?
  - P. ej.: Con tres contraseñas invalidas, el sistema deberá bloquear la cuenta de usuario…
  - ¿Por cuánto tiempo? ¿Cómo se desbloquea? ¿Quién puede desbloquear?
- Incomprobabilidad: ¿Cómo puedo comprobar este ítem ?
  - P. ej.: El sistema proporcionará una disponibilidad del 100%
  - No existe técnica conocida para demostrar la disponibilidad perfecta
- Dependencia, acoplamiento y complejidad excesivos
  - Buscar diagramas de díselo feos y requisitos confusos

El Estándar de Revisión de Software IEEE 1028

1. Descripción general
   - ❑ Propósito, alcance, conformidad, organización, aplicación
2. Referencias
3. Definiciones
4. Revisiones de Gestión
   - ❖ Responsabilidad, entradas/salidas, criterios de entrada/salida, procedimientos
5. Revisiones técnicas
   - ❖ Responsabilidades, entradas/salidas, criterios de entrada/salida, procedimientos

El Estándar de Revisión de Software IEEE 1028

6. Inspecciones
   - ❖ Responsabilidades, entradas/salidas, criterios de entrada/salida, procedimientos, colección de datos, mejora de procesos
7. Revisiones guiadas
   - ❖ Responsabilidades entradas/salidas, criterios de entrada/salida, procedimientos, colección de datos, mejora de procesos
8. Auditorías
   - ❖ Responsabilidades, entradas/salidas, criterios de entrada/salida, procedimientos

Importante

- Las revisiones varían desde muy informales a muy formales
- Depende de los objetivos acordados
- Una revisión formal consiste de una serie de actividades como planificación, definición de criterios, inicio, comprobación de la entrada, preparación, hallazgos, reunión de revisión, análisis de la reunión, reproceso, corrección de defectos, seguimiento y comprobación de la salida
- Una revisión formal implica roles como director (jefe), moderador, autor, revisores y escribano

- Las revisiones constituyen una forma de probar los productos de trabajo del software (incluyendo el código) y pueden realizarse antes de ejecutar las pruebas dinámicas. Los defectos detectados durante las revisiones a menudo son mucho mas baratos de eliminar que los detectados durante las pruebas realizadas ejecutando el código.
- Una revisión podría hacerse íntegramente como una actividad manual, pero también existen herramientas de soporte. La principal actividad manual consiste en examinar un trabajo y hacer comentarios.

Importante :

- Cualquier producto de trabajo de software puede ser objeto de una revisión, incluyendo:
  - las especificaciones de requisitos,
  - Las especificaciones de diseño
  - El código
  - Los planes de prueba
  - Las especificaciones de prueba
  - Los casos de prueba
  - Las guías de usuario o las páginas web.
- Los beneficios de las revisiones incluyen:
  - La detección y corrección temprana de defectos.
  - El desarrollo de mejoras de productividad
  - La reducción de los tiempos de desarrollo
  - El ahorro de tiempo y dinero invertido en pruebas
  - El menor coste de la vida
  - Menos defectos y comunicación mejorada.

2. Análisis Estático basado en Herramientas

2. Análisis Estático basado en Herramientas

- El objetivo principal del análisis estático es la detección de defectos en el código fuente del software y en los modelos de software
- Se realiza sin que la herramienta llegue a ejecutar el software
- Encuentra defectos en lugar de fallos

2. Análisis Estático basado en Herramientas

Objetivos:

- **Detección temprana de fallos**
- **Encontrar defectos (no fallos)**
- **Detectar inconsistencias en el modelo**
- **Mejorar la mantenibilidad**
- **Prevención de defectos futuros**

2. Análisis Estático basado en Herramientas

Defectos típicamente detectados:

- **Variables mal definidas o mal utilizadas**
- **Variables que nunca son utilizadas**
- **Errores en interfaces.** Interfaz inconsistente entre los modulos y componentes.
- **Código inaccesible (muerto)**
- **Lógica errónea o faltante**
- **Estándares incumplidos:** Violaciones de los estándares de programación.
- **Seguridad vulnerable**
- **Errores de sintaxis: Violaciones de la sintaxis de código y los modelos de software.**
- **Complejidad excesiva**

Relación entre las Pruebas Estáticas y Dinámicas

Similitudes

- Ambos procuran identificar defectos
- Funcionan mejor cuando una amplia gama de partes interesadas están involucradas
- Ahorran dinero y tiempo a la compañía respecto al beneficio de las pruebas tempranas y el aseguramiento temprano de la calidad.

Diferencias

- Cada técnica puede encontrar diferentes tipos de defectos de forma más eficaz y eficiente. Ejemplo la mantenibilidad.
- Las técnicas estáticas encuentran defectos en lugar de fallas

Importante

- Las técnicas de pruebas estáticas se apoyan en la inspección manual o el análisis automatizado
- Los defectos detectados tempranamente son a menudo más baratos de eliminar que aquellos detectados después
- Cualquier producto del trabajo de software puede ser revisado
- Las revisiones ofrecen muchos beneficios
- las revisiones, el análisis estático y las pruebas dinámicas son complementarias
- Algunos defectos son más fáciles de encontrar en las revisiones que en las pruebas dinámicas

Beneficios del Análisis Estáticos

- La detección temprana y más barata de defectos (antes de que la ejecución de pruebas comienza)
- Las advertencias acerca de dónde pudieran existir agrupaciones de defectos, debido a la programación peligrosa, alta complejidad, etc.
- Las pruebas dinámicas podrían perder la localización de defectos
- La detección de dependencias e inconsistencias en los modelos de software (p.ej., tales como problemas de enlace en páginas Web)
- La Mantenibilidad mejorada del código y diseño.
- La prevención de defectos basada en métricas recogidas y lecciones aprendidas a partir del análisis

Utilizando Herramienta de Análisis Estático

- Los usuarios típicos son …
  - Programadores, a menudo durante las pruebas de componentes e integración
  - Diseñadores y arquitectos de sistemas durante el diseño
- Durante la introducción inicial contra un sistema existente, las herramientas de análisis estático pueden producir un gran número de mensajes de advertencia
- Los compiladores hacen algún análisis estático, pero hay disponibles muchas herramientas sofisticadas

Importante

- El análisis estático encuentra defectos en el código fuente del software y en los modelos del software
- El análisis estático proporciona numerosos beneficios
- Las herramientas de análisis estático descubren varios defectos
- Las herramientas de análisis estático son típicamente utilizadas por los desarrolladores
- Las herramientas de análisis estáticos pueden producir una gran cantidad de positivos falsos
- Los compiladores apoyan algún análisis estático.

- Las revisiones, el análisis estático y las pruebas dinámicas tienen el mismo objetivo: Identificar defectos.
- Se trata de procesos complementarios; las distintas técnicas pueden encontrar distintos tipos de defectos de una manera eficiente y efectiva.
- En comparación con las pruebas dinámicas, las técnicas estáticas localizan las causas o los fallos (defectos) mas que los propios fallos. Entre los defectos típicos que resultan fáciles de localizar que en las pruebas dinámicas se incluyen:
  - Desviaciones de los estándares
  - Defectos de requisitos, de Diseño
  - Mantenibilidad insuficiente y
  - Especificaciones de interfaz incorrectas

Para recordar

https://www.youtube.com/watch?v=u8ZZcbpxsdQ

Gracias ...