



IEEE Standard Classification for Software Anomalies 1044

Verificación y validación de software



Alcance

- Este estándar es aplicable en cualquier fase del desarrollo de software.
- Identifica categorías para establecer criterios únicos.
- Provee clasificaciones o etiquetas que son específicas para cada proyecto.
- Las acciones correctivas son opcionales.
- Podemos saber en qué momento se originan y en qué parte del procesos encontraremos la anomalía.
- Permite tomar decisiones tempranas para futuras correcciones.



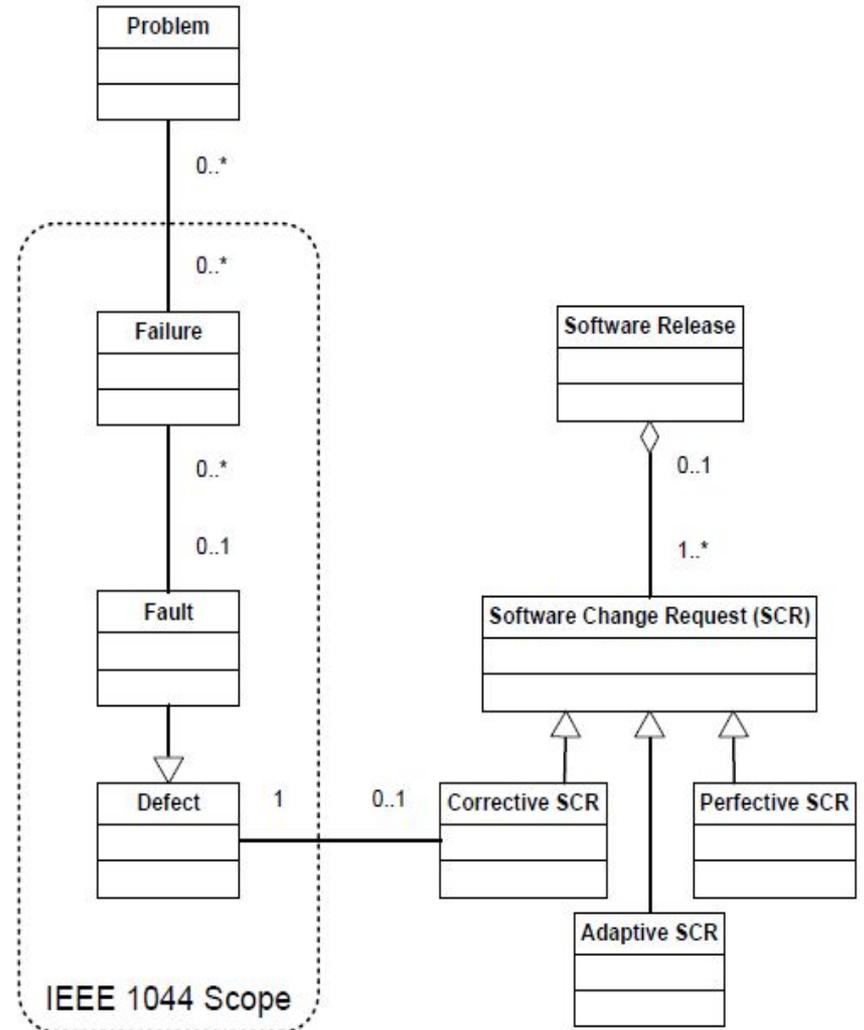
Propósito

El objetivo es definir un vocabulario común con el que diferentes personas y organizaciones puedan comunicarse de manera efectiva sobre anomalías de software

Establecer un conjunto común de atributos que respaldan las técnicas de la industria para analizar datos de defectos y fallas del software

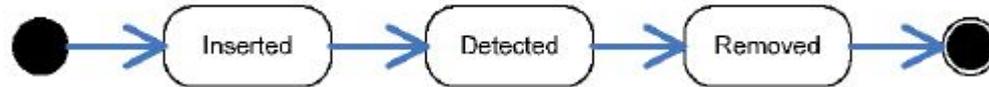
Campo de aplicación (

Este diagrama muestra de forma general el ámbito de aplicación del estándar con notación de diagrama de clases. La línea punteada marca el área de cobertura del estándar.



Campo de aplicación(2/3)

En la siguiente imagen se muestra el ciclo de vida de un defecto, es importante resaltar que no se pueden clasificar los defectos sin haberlos encontrados. Los métodos y procesos para detectar, remover, investigar y resolver defectos están fuera del alcance del estándar.





Campo de aplicación(3/3)

- Clasificar defectos
- Clasificar fallas
- Clasificar fracasos
- Definir un conjunto de atributos de clasificación
- Definir el valor que pueden tomar los atributos al momento de realizar la clasificación



Recordemos algunos conceptos (1/2)

Anomalía: Es una “situación” o “condición” que surge de expectativas. La expectativa puede surgir tanto de la documentación del producto, como de la interpretación de alguna persona sobre el desempeño del software. NO Necesariamente una anomalía (en el IEEE 1044) es un problema de software, ya que puede estar vinculado a situaciones ajenas al mismo.

Defecto: Es una imperfección o deficiencia en el funcionamiento del producto, donde el producto no hace lo que se especifica en los requerimientos ni a las necesidades planteadas.

Error: Una acción humana que produce un resultado incorrecto.



Recordemos algunos conceptos(2/2)

Falla: La manifestación de un error en el software.

Fracaso: Evento en el cual un componente del sistema o el sistema no realiza una función requerida dentro de los límites especificados.

Problema: (A) Dificultad o incertidumbre experimentada por una o más personas, resultado de algún encuentro insatisfactorio con el sistema en uso. (B) Una situación negativa a superar.



El proceso de clasificación (1/2)

La organización debe definir su proceso de clasificación de la siguiente manera:

- A. La meta lograda al clasificar defectos y fallas
- B. El estándar de referencia utilizado para determinar qué comportamientos de software constituyen fallas
- C. Cómo se resolverán los desacuerdos relacionados con las decisiones de clasificación
- D. Cuándo la clasificación debe iniciar y terminar dentro del ciclo de vida del proyecto



El proceso de clasificación (2/2)

- E. Los valores específicos del proyecto u organización que son elegibles para la asignación a atributos de clasificación

- F. Quien debe asignar valores a los atributos de clasificación para cada defecto y falla

- G. ¿Dónde? y ¿cómo? se deben mantener los datos de clasificación



Clasificación de defectos

Los atributos propuestos para clasificar los defectos son los siguientes:

*Id	*V. Detectado	*Efecto	*Ref. de falla
*Descripción	*V. Corregido	*Tipo	*Ref. de cambio
*Status	*Prioridad	*Modo	*Disposición
*Asset	*Severidad	*Act. Insertado	
*Artefacto	*Probabilidad	*Act. detectado	



Clasificación de fallas

La organización deberá contemplar todos los atributos siguientes para cada falla encontrada:

*Falla de ID	*Estatus	*Título	*Descripción
*Ambiente	*Configuración	*Severidad	*Análisis
*Disposición	*Observado por	*Abierto por	*Falla de referencia
*Asignado a	*Cerrado por	*Fecha de observación	*Fecha en que se abrió
*Fecha en que se cerró	*Prueba de referencia	*Incidente de referencia	*Defecto de referencia



Motivos para aplicarlo

- Los datos recolectados son útiles para:
 - Análisis causal de defectos
 - Administración del proyecto
 - Mejora del proceso de software (CMMI)
- Ahorros en tiempo y dinero al descubrir errores más pronto
- Mejora la comunicación dentro de la organización



Ejemplo (1/4)

Problema 1

Sue llama al servicio de atención al cliente e informa que no puede iniciar sesión en el sistema porque falta el campo para la contraseña

*causado por defecto insertado durante la codificación del artefacto Login .asp



Ejemplo (2/4)

Problema 2

* Eventos distintos causados por un único defecto

Problema 3

Durante el test del cliente, Sam observó que el color de la fuente no coincide

*Falla - apariencia de color

Defecto - valor de los datos



Ejemplo (3/4)

Problema 4

Durante una revisión por pares de los requerimientos de software para un nuevo sistema de gestión financiera, Alicia descubre que los valores en los requerimientos están definidos en miles de dólares en lugar de millones de dólares

*Defecto detectado antes de que ocurra una falla



Ejemplo (4/4)

Problema 5

La batería de la compañía se quedó sin energía porque no había una advertencia “Batería baja” a pesar de que se especificó en los requerimientos

*Defecto detectado en producción

Preguntas

1. ¿Cuál es la definición de defecto según el estándar?
2. Menciona 3 atributos de clasificación de una falla
3. Completa el siguiente diagrama el ciclo de vida de un defecto



Figure 3—Defect life cycle as a UML statechart diagram