



Universidad
del Cauca®

CALIDAD DE SOFTWARE
Mag. Carlos Alberto Ardila Albarracín

MÓDULO 2.1
ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE
Conceptos, Técnicas estáticas y estándares

1. Control en Proyectos de Software

¿Recorridos, Revisiones, Inspecciones?

Un **recorrido** es "una técnica de análisis estático en la que un diseñador o programador dirige a los miembros del equipo de desarrollo y otras partes interesadas (técnicas) a través de un producto de software y los participantes hacen preguntas y hacen comentarios sobre posibles errores, violaciones de los estándares de desarrollo y otros problemas".

Una **revisión** es "un proceso o una reunión durante la cual un producto software se presenta al personal del proyecto, a los gerentes, a los usuarios, a los clientes, a los representantes de los usuarios u otras partes interesadas para el comentario o la aprobación"

Una **inspección** es "un examen visual de un producto de software para detectar e identificar anomalías de software, incluyendo errores y desviaciones respecto de las normas y especificaciones".

1. Control en Proyectos de Software

¿Recorridos, Revisiones, Inspecciones?

Método	Metas Típicas	Atributos típicos
Recorridos	Sobrecarga mínima Formación para desarrolladores Respuesta rápida	Poco / sin preparación Proceso informal Sin mediciones NO SON RTF!
Revisiones Técnicas	Elicitación de requisitos Resolución de ambigüedad Formación / Entrenamiento	Proceso formal Presentación del autor Amplia gama de discusiones
Inspecciones	Detectar y eliminar todos los defectos de manera eficiente y efectiva	Proceso formal Listas de control (Checklists) Mediciones

1. Control en Proyectos de Software Inspecciones

DESCRIPCION	OBJETIVOS
Las inspecciones son realizadas por personal altamente calificado. La resolución de defectos es obligatoria y el trabajo rehecho es formalmente verificado.	<ul style="list-style-type: none">✓ Detectar defectos.✓ Verificar la resolución de defecto.
ENTRADA	CRITERIO DE ENTRADA
<ul style="list-style-type: none">✓ Elemento(s) de software a inspeccionar.✓ Especificaciones de los elementos de software a inspeccionar.✓ Lista de chequeo de la inspección.✓ Planes, estándares o guías para contrastar el elemento de software a inspeccionar	La inspección puede iniciarse cuando: <ul style="list-style-type: none">✓ Los elementos de software estén disponibles para su inspección.✓ La documentación pertinente esté disponible.✓ Para reinspecciones, todos los ítems especificados en la lista de defectos deben haber sido resueltos.
SALIDA	CRITERIO DE SALIDA
Lista de defectos que incluya: <ul style="list-style-type: none">✓ Localización del defecto.✓ Descripción del defecto✓ Categorización del defecto.✓ Resumen de los defectos por categoría	<ul style="list-style-type: none">✓ La inspección se considera completa cuando todos los defectos de los elementos de software han sido detectados y resueltos.

1. Control en Proyectos de Software **Revisiones Técnicas Formales**

DESCRIPCION	OBJETIVOS
Evaluación formal de uno o varios elementos de software.	<ul style="list-style-type: none">✓ Evaluar la conformidad con especificaciones y planes.✓ Asegurar la integridad de los cambios.
ENTRADA	CRITERIO DE ENTRADA
<ul style="list-style-type: none">✓ Objetivos de la revisión.✓ Elemento(s) de software a examinar.✓ Especificaciones de los elementos de software a examinar.✓ Planes, estándares o guías para contrastar el elemento de software a revisar.	<p>La revisión puede iniciarse cuando:</p> <ul style="list-style-type: none">✓ Se establecen los objetivos de la revisión.✓ Los responsables de los elementos de software están listos para la revisión.✓ El líder establezca que los elementos de software y la documentación pertinente está suficientemente completa.
SALIDA	CRITERIO DE SALIDA
<p>Informe de Revisión que incluya:</p> <ul style="list-style-type: none">✓ Las entradas para la revisión.✓ Lista de deficiencias no resueltas, de los elementos de software examinados.✓ Recomendaciones para eliminar las deficiencias no resueltas.	<p>La revisión se considera completa cuando:</p> <ul style="list-style-type: none">✓ Todos los aspectos considerados en los objetivos de la revisión han sido tratados.✓ El informe de la revisión ha sido producido.

1. Control en Proyectos de Software **Revisiones Técnicas Formales**

Plantilla para Registro de errores tipográficos [shorturl.at/krT17]

Plantilla para Registro de problemas detectados [shorturl.at/dEFNP]

Plantilla para Reporte Sumario [shorturl.at/IPRW9]

Ejemplos de Listas de Verificación:

Lista Wiegner para especificación de requerimientos [shorturl.at/dhklW]

Lista NASA (JPL) para requerimientos funcionales [shorturl.at/bcPWZ]

Lista para inspección de código Java [shorturl.at/isTZ1]

2. Aseguramiento de Calidad de Software (SQA)

2.1. Motivación

¡ENFOQUES DE CALIDAD!

ENFOQUE GLOBAL

Orientado para las medianas/grandes corporaciones, con un plazo de implantación largo (entre 3 y 5 años). Su objetivo es conseguir el liderazgo. Está orientado a conseguir la satisfacción del cliente mediante la *mejora continua*. Los tres puntos fundamentales para conseguir esto son:

- Liderazgo de la dirección en la implantación del sistema.
- Participación de todo el personal de la organización.
- Formación continua.

2. Aseguramiento de Calidad de Software (SQA)

2.1. Motivación

ENFOQUE PRÁCTICO

Enfoque que puede ser adoptado por cualquier tipo de empresa o industria (fundamentalmente PYMES), y cuyo objetivo es *mejorar en el corto plazo*.

Una forma de operar en estos casos es:

1. Identificando y cuantificando los defectos
2. Analizando las causas que los producen
3. Estudiando las acciones a tomar para eliminarlos

Existen distintas herramientas para la realización de estas actividades, como son las Metodologías de Evaluación y Mejora de Procesos, la Detección de Problemas, etc.

2. Aseguramiento de Calidad de Software (SQA)

2.1. Motivación

ENFOQUE TÁCTICO

Orientado a cualquier tipo de empresa. Este tipo de implantación de Sistemas de Calidad suele tener como fin la certificación del Sistema de Calidad por un organismo externo acreditado para ello. El periodo de implantación suele estar entre 1 y 2 años dependiendo de la madurez de la organización).

Esta implantación se hace en base a un modelo. Los más utilizados son los modelos para el Aseguramiento de la Calidad definidos en las normas ISO 9000.

¿Qué va primero la táctica o la estrategia?

Estrategia: conjunto de acciones que permitan desarrollar un plan previamente establecido para conseguir un objetivo (QUÉ).

La táctica se encarga de aspectos puntuales y concretos que forman parte del plan maestro (EL CÓMO).

Lo anterior nos lleva a concluir que la táctica hace parte de la estrategia.

ESTRATEGIA puede ser una sola, en tanto que pueden existir varias tácticas en dicho plan

2. Aseguramiento de Calidad de Software (SQA)

2.2. Actividades de SQA

El aseguramiento de calidad de software comprende una gran variedad de tareas asociadas con siete actividades principales:

- ✓ Aplicación de Métodos Técnicos y Herramientas de Análisis, diseño, codificación y prueba.
- ✓ Realización de Revisiones Técnicas Formales que se aplican durante cada paso de la ingeniería de software.
- ✓ Prueba del software.
- ✓ Control de la documentación y los cambios hechos a éste.
- ✓ Aseguramiento de que se cumple con los estándares de desarrollo del software.
- ✓ Mediciones.
- ✓ Registro y realización de informes.

2. Aseguramiento de Calidad de Software (SQA)

2.2. Actividades de SQA

La lista a continuación muestra las evaluaciones a realizar para cada disciplina asociada al proceso de desarrollo:

✓ **Requerimientos de Software**

- Plan de desarrollo de software
- Estándares del software y manual de procedimientos
- Plan de administración de la configuración
- Plan del programa de calidad del software
- Especificación de requerimientos del software
- Especificación de requerimientos de interfaces
- Documentos con concepto de operación

2. Aseguramiento de Calidad de Software (SQA)

2.2. Actividades de SQA

✓ Análisis

- Planes y programas revisados
- Documentos con diseño de alto nivel del testing del producto de software
- Plan de testing del software
- Manual del operador
- Manual del usuario

✓ Diseño

- Planes y programas revisados
- Documentos de análisis de software
- Documentos de diseño de interfaces
- Documentos de diseño de bases de datos
- Casos para tests unitarios
- Descripción de test del software
- Manuales revisados

2. Aseguramiento de Calidad de Software (SQA)

2.2. Actividades de SQA

✓ **Codificación y test unitario**

- Planes y programas revisados
- Código fuente
- Código objeto
- Procedimientos de tests unitarios
- Resultados de tests unitarios
- Procedimientos de tests de integración
- Manuales revisados

✓ **Testing del sistema**

- Planes y programas revisados
- Informe de test de sistema
- Código fuente revisado
- Código objeto revisado
- Manuales

2. Aseguramiento de Calidad de Software (SQA)

2.3. Garantía de Calidad Estadística

La garantía de calidad estadística refleja una tendencia creciente en toda la industria de establecer la calidad en forma más cuantitativa. Para el software, la garantía de calidad estadística implica los siguientes pasos:

- ✓ Agrupar y clasificar la información sobre los errores y defectos del software.
- ✓ Intentar encontrar la causa subyacente de cada defecto.
- ✓ Usando el principio de Pareto (80% de los problemas de calidad se pueden encontrar en el 20% de todas las posible causas), se aísla el 20% (los pocos, pero vitales).
- ✓ Una vez que se han identificado los defectos vitales, se actúa para corregir los problemas que han producido los defectos.

Las raíces de la mayoría de los defectos (fallas) realmente difíciles, se pueden encontrar en un número relativamente limitado de causas.