

MOELO TEÓRICO PARA REVISIÓN Y AUDITORIA DE SOFTWARE  
(Norma IEEE-1028-1.988)

YANETH RANGEL SÁNCHEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD INGENIERIA FISICO MECANICAS  
ESCUELA DE INGENIERIA DE SISTEMAS  
ESPECIALIZACIÓN DE INGENIERÍA DEL SOFTWARE  
BUCARAMANGA  
2.004

ESCUELA DE INGENIERÍA DE SISTEMAS  
POSGRADO DE INGENIERÍA DEL SOFTWARE

Trabajo de investigación presentado como requisito para optar al título de  
Especialista en Ingeniería del Software

DIRECTOR:  
LUIS IGNACIO GONZÁLEZ RAMÍREZ  
Ingeniero de Sistemas, MsC

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD INGENIERIA FISICO MECANICAS  
ESCUELA DE INGENIERIA DE SISTEMAS  
ESPECIALIZACIÓN DE INGENIERÍA DEL SOFTWARE  
BUCARAMANGA  
2.004

## **DEDICATORIA**

A Dios porque a diario llena mi vida de bendiciones.

A mi amado esposo VICTOR HUGO por su apoyo incondicional.

A mi bebecito adorado SANTIAGUITO porque llegó a mi vida en el mejor momento.

A mis padres y hermanos porque cuento con ellos siempre.

A la ESSA porque me permite desarrollarme profesionalmente.

A mis familiares y amigos

**YANETH**

## CONTENIDO

INTRODUCCIÓN.....	1
INTRODUCCIÓN.....	1
1. DEFINICIÓN DE LA NORMA IEEE 1028-1.988 .....	8
1.1. OBJETIVO .....	8
1.2. REFERENCIAS .....	8
1.3. DEFINICIONES .....	8
1.3.1. Auditoria .....	8
1.3.2. Revisión.....	8
1.3.3. Elemento del software .....	9
1.4. RESUMEN DE LA NORMA .....	9
1.4.1. INTRODUCCIÓN.....	9
1.4.1.1. Requisitos previos al Proceso de Revisión.....	11
1.4.1.2. Requisitos previos al Proceso de Auditoría .....	12
1.4.1.3. Plantilla de Descripción Procedimental.....	12
1.4.2. EL PROCESO DE REVISIÓN GERENCIAL.....	12
1.4.2.1. Objetivos .....	12
1.4.2.2. Resumen.....	12
1.4.2.3. Responsabilidades Especiales .....	13
1.4.2.4. Entradas .....	13
1.4.2.5. Criterios de Entrada .....	13
1.4.2.6. Procedimientos .....	13
1.4.2.7. Criterios de Salida.....	14
1.4.2.8. Salida.....	14
1.4.2.9. Auditoria .....	14
1.4.3. EL PROCESO DE REVISIÓN TÉCNICA .....	14
1.4.3.1. Objetivos .....	14
1.4.3.2. Resumen.....	14
1.4.3.3. Responsabilidades Especiales .....	14
1.4.3.4. Entrada.....	15
1.4.3.5. Criterios de Entrada .....	15
1.4.3.6. Procedimientos .....	15
1.4.3.7. Criterio de Salida .....	16
1.4.3.8. Salida.....	16
1.4.3.9. Auditoria .....	16
1.4.4. EL PROCESO DE INSPECCIÓN DEL SOFTWARE .....	16
1.4.4.1. Objetivo .....	16
1.4.4.2. Resumen.....	16
1.4.4.3. Responsabilidades Especiales .....	16
1.4.4.4. Entradas .....	17
1.4.4.5. Criterio de Entrada .....	17

1.4.4.6.	Procedimientos .....	18
1.4.4.7.	Criterio de Salida .....	19
1.4.4.8.	Salida.....	19
1.4.4.9.	Auditoria .....	19
1.4.5.	EL PROCESO DE REVISIÓN.....	19
1.4.5.1.	Objetivo .....	20
1.4.5.2.	Resumen.....	20
1.4.5.3.	Responsabilidades Especiales .....	20
1.4.5.4.	Entradas .....	21
1.4.5.5.	Criterio de Entrada .....	21
1.4.5.6.	Procedimientos .....	21
1.4.5.7.	Criterio de Salida .....	21
1.4.5.8.	Salida.....	22
1.4.5.9.	Auditoria .....	22
1.4.6.	EL PROCESO DE AUDITORIA .....	22
1.4.6.1.	Objetivo .....	22
1.4.6.2.	Resumen.....	22
1.4.6.3.	Responsabilidades Especiales .....	22
1.4.6.4.	Entrada.....	23
1.4.6.5.	Criterios de Entrada .....	23
1.4.6.6.	Procedimientos .....	23
1.4.6.7.	Criterio de salida.....	24
1.4.6.8.	Salida.....	25
1.4.6.9.	Auditoria .....	25
1.5.	ANÁLISIS DE LA NORMA IEEE 1028-1.988.....	25
1.5.1.	QUIÉN LA APLICA .....	26
1.5.2.	QUÉ ES.....	26
1.5.3.	PARA QUÉ SIRVE .....	26
1.5.4.	CUÁNDO SE APLICA.....	26
1.5.5.	FRECUENCIA .....	26
1.5.6.	RECURSOS.....	26
1.5.7.	DESARROLLO DE LA NORMA IEEE 1028.....	27
2.	MODELO TEÓRICO.....	29
3.	DESCRIPCIÓN DEL PROCEDIMIENTO STANDARD .....	43
4.	CONCLUSIONES .....	49
5.	RECOMENDACIONES.....	50
	BIBLIOGRAFÍA .....	51
	ANEXOS .....	52

## LISTA DE ANEXOS

<b>ANEXO A:</b>	Traducción de la Norma IEEE 1028 – 1988.....	51
<b>ANEXO B:</b>	Caso de aplicación de la Norma IEEE 1028 – 1988.....	107
<b>ANEXO C:</b>	Plantilla Norma IEEE 1028 – 1988.....	114
<b>ANEXO D:</b>	Norma IEEE 1028 – 1988.....	132

TÍTULO: MODELO TEÓRICO PARA REVISIÓN Y AUDITORIA DE SOFTWARE  
(Norma IEEE-1028-1.988) \*

Yaneth Rangel Sanchez \*\*

Revision  
Auditoria  
Elemento del software  
Aseguramiento de Calidad del Software  
Evaluación  
Verificación  
Validación

La IEEE 1028 - 1.988 es una norma para integrar actividades de revisión y auditoria aplicada a productos y procesos durante todas las etapas del ciclo de vida del software.

La auditoria es una evaluación de productos o procesos del software para determinar cumplimiento, seguimiento, especificaciones y procedimientos basados sobre objetivos.

La revisión es una evaluación de elementos del software o estado del proyecto para determinar discrepancias de los resultados planeados y las mejoras recomendadas.

Para el desarrollo y mantenimiento del software es necesario evaluar, verificar y validar para lograr la conformidad del mismo.

El procedimiento para planear, preparar y ejecutar una revisión o auditoria debe contemplar: objetivo, resumen, responsabilidades, entradas, criterios de entrada, procedimientos de planificación, apreciación global, preparación y revisión, criterios de salida, salida y auditoria.

Los procesos definidos en la norma IEEE 1028 – 1988 incluye revisión gerencial, revisión técnica, inspección del software, revisión y auditoria.

La revisión gerencial provee recomendaciones para hacer actividades progresivas, cambiar la dirección del proyecto y mantener control global.

La revisión técnica evalúa un elemento del software para probar que coincida con sus características técnicas, que el desarrollo o mantenimiento se hace acorde a diseños, normas y pautas aplicables al proyecto y que los cambios son implementados apropiadamente.

La inspección del software descubre e identifica los defectos de los elementos del software para verificar que satisface las características técnicas, verificar que conforman normas aplicables, identificar desviación de normas y especificaciones, reunir datos de administración del software y no examinar alternativas.

La revisión evalúa elementos de software para encontrar defectos, omisiones y contradicciones, perfeccionarlos, considerar implementaciones alternas y descubrir problemas.

La auditoria provee la conformidad de los productos y procesos.

\* Trabajo de investigación

\*\* Facultad de Ingeniería Físico mecánicas, Escuela de Ingeniería de Sistemas, Especialización en Ingeniería del Software. Luis Ignacio González Ramírez

TITLE: THEORETICAL MODEL FOR SOFTWARE REVIEW AND AUDIT (standard IEEE-1028-1.988) \*

Yaneth Rangel Sanchez \*\*

Review

Audit

Element of software

Securing of Quality of Software

Evaluation

Verification

Validation

IEEE 1028 - 1,988 are a norm to integrate activities of review and audit applied to products and processes during all the stages of the service life of software.

The audit is a product evaluation or processes of software to determine fulfillment, pursuit, specifications and procedures based on objectives.

The revision is an evaluation of elements of software or state of the project to determine discrepancies of the planned results and the recommended improvements.

For the development and maintenance of software it is necessary to evaluate, to verify and to validate to obtain the conformity of the same one.

The procedure to plan, to prepare and to execute a revision or audit must contemplate: objective, summary, responsibilities, entrances, criteria of entrance, procedures of planning, global appreciation, preparation and revision, criteria of exit, exit and audit.

The processes defined in the standard IEEE 1028 - 1988 include managerial revision, technical revision, inspection of software, revision and audit.

The managerial revision provides recommendations to make activities progressive, to change the direction of the project and to maintain control global.

The technical revision evaluates an element of software to prove that it agrees with his technical characteristics, that the development or maintenance becomes agreed to designs, norms and guidelines applicable to the project and that the changes are implemented appropriately.

The inspection of software discovers and identifies the defects of the elements of software to verify that it satisfies the technical characteristics, to verify that they conform applicable norms, to identify deviation of norms and specifications, to collect data of administration of software and not to examine alternatives. The revision evaluates software elements to find defects, omissions and contradictions, to perfect them, to consider implementations alternating and to discover problems.

The audit provides the conformity with products and processes.

\* Work of investigation.

\*\* Faculty of Physical Engineering Mechanical, School of Engineering of Systems, Specialization in Engineering of Software. Luis Ignacio González Ramírez

## **INTRODUCCIÓN**

El Instituto de Ingeniería Eléctrica y Electrónica (IEEE) es la sociedad profesional técnica más grande del mundo. Fundada en 1884 por un grupo de practicantes de la nueva disciplina de ingeniería eléctrica, hoy el Instituto lo conforman más de 320.000 miembros quienes dirigen y participan en sus actividades en aproximadamente 150 países. Los hombres y mujeres de la IEEE son profesionales científicos y técnicos que hacen avances revolucionarios en ingeniería los cuales son readaptados al mundo de hoy.

Los objetivos técnicos de la IEEE se centran en avanzar la teoría y práctica de la ingeniería eléctrica, electrónica y de computadores y la ciencia de computadores. Para realizar estos objetivos, la IEEE patrocina conferencias técnicas, simposios y reuniones locales alrededor del mundo: publica aproximadamente el 25% de los artículos técnicos del mundo de ingeniería eléctrica, electrónica y computadores; suministra programas educativos para tener a sus miembros conocedores y expertos en el estado del arte. Los propósitos de todas estas actividades son dos: (1) Mejorar la calidad de vida de todas las personas a través de fomentar el conocimiento público de las aplicaciones e influencias de estas tecnologías; y (2) mejorar la categoría de la profesión de ingeniería y sus miembros.

## **COMO SE HACEN LAS NORMAS**

Las normas de la IEEE son desarrolladas por el Comité Técnico de Sociedades de la IEEE y el Comité Coordinador de Normas de la IEEE. Los miembros de los comités sirven voluntariamente y sin compensación. No son necesariamente miembros del instituto.

El uso de la norma IEEE es totalmente voluntario. La existencia de una norma IEEE no implica que no hay otras maneras de producir, probar, medir, comprar, mercadear o proveer otros géneros y servicios relacionados con la norma IEEE. Además, los puntos de vista expresados en una norma son aprobados y están sujetos a cambios de acuerdo a su desarrollo sobre el estado del arte y comentarios recibidos de usuarios de la norma. Todas las normas de IEEE están sujetas a repasos por lo menos cada cinco años para revisión o reconfirmación. Cuando un documento de más de cinco años no ha sido reconfirmado, es razonable concluir que no refleja totalmente el estado presente del arte. Se avisa a los usuarios para verificar la norma IEEE.

Comentarios para la revisión de la norma IEEE son aceptados de cualquier interesado, indiferentemente de su afiliación como miembro de IEEE. Sugerencias para cambios en los documentos deben estar en el formato del cambio propuesto, apoyado de comentarios apropiados.

## DESCRIPCIÓN DEL PROBLEMA

**"The Roman bridges of antiquity were very inefficient structures. By modern standards, they used too much stone, and as a result, far too much labor to build. Over the years we have learned to build bridges more efficiently, using fewer materials and less labor to perform the same task."**

-- Tom Clancy, The Sum of All Fears

La sociedad del siglo XXI se ha denominado la sociedad del conocimiento, de la información y la comunicación. Este tipo de sociedad según Vergara "funciona sobre el supuesto, que al futuro, la sociedad estará definida por la capacidad de producir conocimiento e información circulante disponible, a través de los desarrollos tecnológicos alcanzados por los avances de la electrónica, la informática, la telemática y las comunicaciones, como efectos de la Segunda Revolución Industrial". (Vergara, 1997).

En este escenario, la formulación, dirección y ejecución de proyectos de desarrollo tecnológico tienen una relevancia sin precedentes en la construcción de nuestros entornos socio económicos y culturales; que en últimas determinan la calidad de vida a que estamos sujetos. No obstante esta situación en nuestro país son muy tímidos los esfuerzos encaminados a lograr que los proyectos exitosos sean la regla y no la excepción a través de la aplicación sistemática y contextualizada de los estándares de clase mundial que se han construido y se mantienen en constante evolución.

Ahora bien, es cierto que existen los mencionados estándares o normas en las que han participado numerosos investigadores y equipos interdisciplinarios de profesionales de todo el mundo, en los cuales se han invertido enormes cantidades de recursos; las estadísticas reflejan una situación que sigue siendo preocupante aunque viene en franca recuperación.

The Standish Group International Inc. en su informe sobre su investigación quinquenal (CHAOS Research 1994-1998) presentado a la comunidad internacional en el año 1999, muestra los avances que se han logrado en términos del promedio de proyectos de desarrollo de software exitosos y la disminución significativa en sobrecostos y tiempos de entrega. Según lo expresado por este grupo en el paper Project Resolution: The 5-year view,

“La mejor noticia es que la dirección de proyectos exitosos se da más a menudo”, pasando de una media del 16% en el año 1994 a una del 26% en el año 1998.

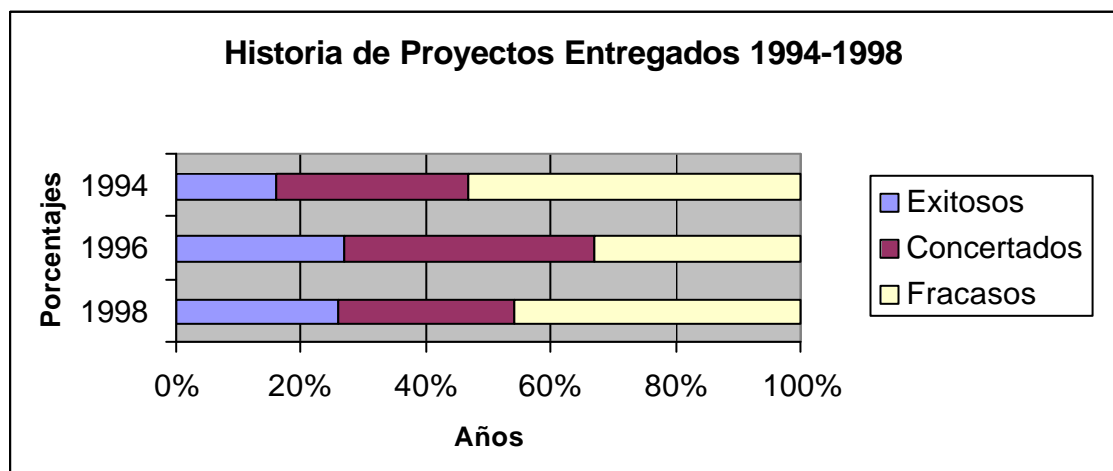
The Standish Group clasifica los proyectos en tres tipos según la forma como son entregados:

- Exitosos: el proyecto es entregado de acuerdo con el tiempo, presupuesto y características contratadas originalmente.
- Concertados: el proyecto es entregado pero con sobre costos, tiempos adicionales, y/o características diferentes a las inicialmente contratadas.
- Fracazos: el proyecto es cancelado antes de completarse.

Los datos arrojados por las investigaciones realizadas entre los años 1994 y 1998 muestran los progresos que se han tenido en este campo.

Tamaño de la Compañía	Casos de éxito 1994	Casos de éxito 1998	Vlr. Prom. 1994	Vlr. Prom. 1998	Delta
Grande	9%	24%	US\$2.3M	US\$1.2M	-65%
Mediana	16%	28%	US\$1.3M	US\$1.1M	-41%
Pequeña	28%	32%	US\$0.4M	US\$0.6M	-4%

Una muestra analizada de 23.000 proyectos entregados, tomada de todos los sectores de la industria y de compañías grandes, medianas y pequeñas; los resultados todos los tamaños de compañías



The Standish Group cree que existen tres factores críticos que potencialmente influyen en el éxito de un proyecto software: el tamaño del proyecto, la duración de la ejecución y el tamaño del equipo de desarrollo. Dicho en otras palabras el riesgo de fracaso es inversamente proporcional a la complejidad del proyecto; por lo que el grupo de investigadores concluye en su paper *The 3 Pillars of Project Success*, “Aunque algunas escuelas podrían argumentar que grandes proyectos con más fondos y recursos deberían ser mas exitosos, y deberían producir más funciones, lo contrario parece ser la realidad”.

Por otro lado, ¿que está sucediendo en los países desarrollados donde además de proveerse de datos o información provenientes de estudios como el mencionado?

Pues bien, estos gobiernos en asocio con universidades y profesionales IT se han dado a la tarea de apropiarse de la metodología para el aseguramiento de la calidad, además de ir desarrollando políticas y lineamientos a las cuales deben acogerse las empresas y profesionales contratistas. Podemos observar a manera de ejemplo los esfuerzos que realiza el gobierno canadiense a través de su Ministerio de Hacienda (TSB Treasury Board of Canadá Secretariat) al desarrollar el EMF (Enhanced Management Framework) que define los principios y parámetros que rigen la administración de proyectos informáticos.

En contraste, ¿qué está sucediendo en nuestro país?

En principio por la inexistencia de datos confiables e información sistematizada los estudios que desnuden la realidad no se han hecho; se presume que la situación es tan o más difícil que en los países avanzados y que la aplicación de normas y estándares se da en proporciones muy reducidas y en entornos de laboratorio o de nichos exclusivos. Existen sin embargo esfuerzos individuales de algunos productores de software por lograr alcanzar alguna certificación que les permita ser más competitivos y abarcar mayores porciones del mercado; proyectos del gobierno (como Proyecto Inteligente de Colciencias) que pretenden formar grupos pequeños de profesionales que puedan obtener certificaciones en diferentes líneas de tecnología con el fin de penetrar mercados internacionales de la industria tecnológica; y proyectos de laboratorios en las universidades que buscan transformar las concepciones en sus estudiantes de pregrado y postgrado.

Sin embargo estos esfuerzos son aislados, desarticulados y poco significativos a la luz de las necesidades que requiere satisfacer el país.

Por las razones y hechos expuestos, es que los estudiantes que participamos en la quinta cohorte de la Especialización en Ingeniería del Software hemos decidido desarrollar de manera conjunta y coordinada en asocio con el grupo de investigación CIDLIS de la Escuela de Ingeniería de Sistemas; el análisis y contextualización de un subconjunto de las normas de la IEEE para el aseguramiento de la calidad en los productos y en la gestión de proyectos de construcción de software; para ofrecerlos a la comunidad técnica de la región y del país.

## **REFERENCIA DEL TRABAJO**

### **METODOLOGÍA**

La monografía fue desarrollada bajo el ambiente de trabajo del CIDLIS (Centro de Investigación Línea de Ingeniería de Sistemas) con la participación de la quinta promoción de estudiantes de la especialización de Ingeniería del Software de la Universidad Industrial de Santander.

La metodología utilizada se basó en el desarrollo de talleres de los siete grupos de trabajo, donde cada uno desarrolló las siguientes actividades:

- ❖ Traducción de la norma
- ❖ Presentación de la norma
- ❖ Explicación de la norma
- ❖ Análisis de la norma
- ❖ Discusión de la norma
- ❖ Aplicación de la norma a un caso de estudio particular

### **ESTRUCTURA DEL DOCUMENTO**

En el presente documento refleja el desarrollo de la monografía de reflexión de la Norma IEEE 1028-1.988: Revisión y Auditoria del Software.

El documento se inicia con la introducción en donde se define qué es la IEEE, quienes la conforman, cual es su objetivo y cómo se desarrollan las normas. Se hace una descripción del problema y se explica la metodología de trabajo utilizada.

En el capítulo uno, se define la Norma IEEE 1028-1.988, explicando su objetivo, las referencias que se hacen a otras normas, la definición de términos claves utilizados por la norma y el resumen general de la norma. En el resumen se presenta la introducción de la norma, los requisitos previos para la revisión y auditoria y la plantilla que emplea la norma. Además, se presentan los cinco procesos que contempla la IEEE en su estándar 1028-1.988: Revisión Gerencial, Revisión Técnica, Inspección del Software,

Revisión y Auditoría. Al final, se incluye el análisis de la norma para Revisión y Auditoría de Software.

En el segundo capítulo se expone el modelo teórico propuesto, utilizando la plantilla diseñada para el análisis de la aplicación de la norma IEEE 1028-1.988 en cualquier caso que se requiera.

En el capítulo tres, se hace una descripción del procedimiento estándar de la norma IEEE 1028-1.988.

En el capítulo cuatro recopila las conclusiones a las cuales llegamos una vez hecha la traducción, el análisis y la descripción del procedimiento estándar de la norma.

El capítulo quinto resume las recomendaciones para la aplicación de la Norma.

Luego se presenta el material bibliográfico utilizado para desarrollar la monografía.

Por último, se reúnen los anexos, resultado de la reflexión hecha a la norma de la IEEE para Revisión y Auditoría de Software.

## **1. DEFINICIÓN DE LA NORMA IEEE 1028-1.988**

### **1.1. OBJETIVO**

Proveer definiciones y requerimientos uniformes para la revisión y auditoria de procesos.

No establece la necesidad de conducir revisiones ó auditorias específicas debido a que es definida por política local.

Cuando se requiere revisiones y auditorias específicas se definen los procedimientos de las normas.

Esta norma provee dichas definiciones para los procesos de revisión y auditoria que son aplicables a productos y procesos durante el ciclo de vida del software.

### **1.2. REFERENCIAS**

(1) Norma ANSI/IEEE 729-1983, Norma IEEE Glosario de Terminología de la Ingeniería del Software.

(2) Norma ANSI/IEEE 829-1983, Norma IEEE Documentación de Prueba del Software.

### **1.3. DEFINICIONES**

#### **1.3.1. Auditoria**

Es una evaluación independiente de productos o procesos del software para determinar cumplimiento, seguimiento, especificaciones y procedimientos de la norma basados sobre criterios objetivos que incluyan documentos que especifiquen

- (1) La forma o contenido de los productos
- (2) El proceso por que se producirá.
- (3) Cómo cumplir las normas o seguimiento que se debe establecer.

#### **1.3.2. Revisión**

Es una evaluación de elemento(s) del software o estado del proyecto para determinar discrepancias de los resultados planeados y las mejoras recomendadas.

Sigue un proceso formal

- ✓ Revisión gerencial
- ✓ Revisión técnica
- ✓ Inspección del software
- ✓ Revisión

### 1.3.3. Elemento del software

Documento en proceso producido o adquirido durante el desarrollo o mantenimiento del software.

Ejemplos

- (1) Documentación de la Planificación del Proyecto
  - ✓ Plan de desarrollo del software
  - ✓ Plan de verificación y validación del software
- (2) Requerimientos del Software y especificaciones de diseño.
- (3) Documentación de prueba de esfuerzo
- (4) Documentación del soporte
- (5) Código fuente del programa
- (6) Representación de soluciones del software implementados
- (7) Informes (Revisiones, Auditoria, Estado del proyecto) y Datos (Descubrimiento del defecto, Pruebas)

## 1.4. RESUMEN DE LA NORMA

### 1.4.1. INTRODUCCIÓN

Las revisiones y Auditorias pueden ser usadas como soporte de objetivos asociados con:

- ✓ Aseguramiento de la calidad del software. Válido a través del desarrollo y mantenimiento del software. Incluyen la necesidad de:
  - ◇ Evaluación
  - ◇ Verificación
  - ◇ Validación. Las pruebas están fuera del alcance de ésta norma
  - ◇ Conformidad. La auditoria debe incluir:
    - ◆ Ejecución de revisiones.
      - ◆ Pruebas para determinar la consistencia del producto con respecto a las bases establecidas (Por ejemplo, especificaciones de requerimientos del software o descripción del diseño).
      - ◆ Dar testimonio de las correcciones en la implementación.
- ✓ Administración del proyecto
- ✓ Administración de la configuración o función de control similar

Los procesos definidos en esta norma incluye

- (1) Revisión Gerencial
- (2) Revisión Técnica
- (3) Inspección del Software

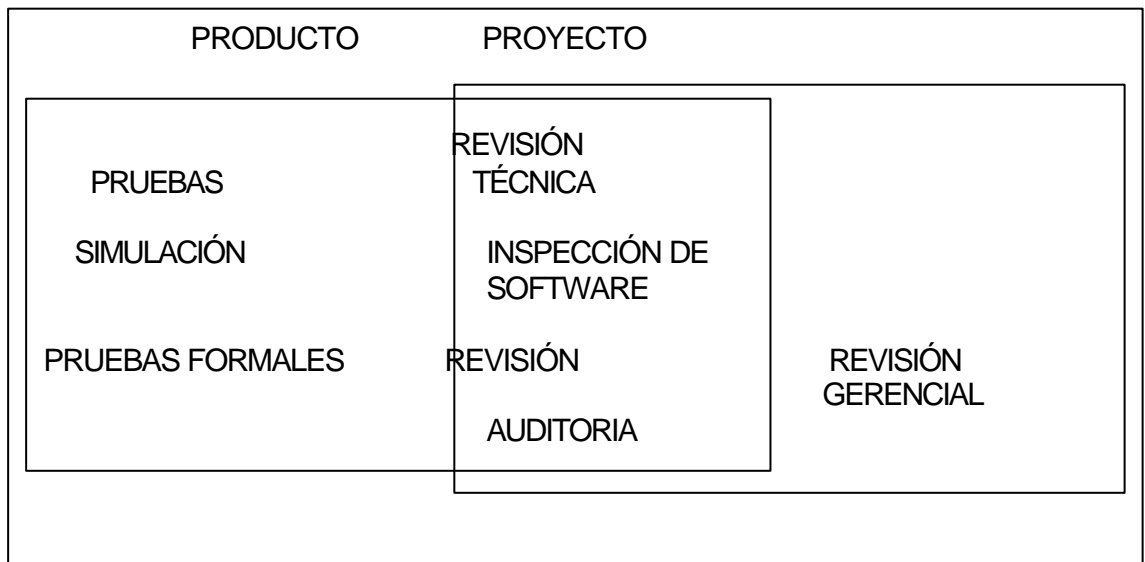
- (4) Revisión
- (5) Auditoria.

**Algunos Procesos Principales para Alcanzar Objetivos de Calidad**

<b>OBJETIVOS</b>	<b>PROCESOS PRINCIPALES QUE INCLUYE</b>
Evaluación	Revisión gerencial, revisión técnica
Verificación	Inspección, Revisión
Validación	Pruebas
Conformidad	Auditoria

El aseguramiento de la calidad de los productos y procesos, provee una visión de cómo varios procesos de aseguramiento de calidad pueden ser trazados para examinar productos y proyectos.

**Relación de Unos Procesos de Aseguramiento de Calidad para Productos y Proyectos**



El examen técnico y gerencial ocurren en varias fases durante el ciclo de vida del proyecto. Los resultados de dichos exámenes permiten mejorar el método de aseguramiento de la calidad del software y la habilidad para encontrar disminución de tiempo y costos. La evaluación de elementos del

software ocurre durante la producción de los elementos y su realización. Esto asegura que los elementos encarnen completa y correctamente sus especificaciones básicas.

Cualquier proceso normal tiene condiciones de requisitos previos; esas condiciones, no son suficientes para la realización del proceso.

#### **1.4.1.1. Requisitos previos al Proceso de Revisión**

El objetivo de revisar los elementos del software es evaluar el software o el estado del proyecto para identificar diferencias con los resultados planeados y hacer las recomendaciones apropiadas.

Los siguientes cargos administrativos están para alcanzar estos objetivos:

- **Administrador**

- (1) Responsable de las revisiones, y el producto anticipado.
- (2) Responsable de ejecutar las revisiones y reportar el resultado.
- (3) Provee los recursos necesarios de tiempo, personal, presupuesto, y facilidades requeridas para planear, definir, ejecutar y manejar las revisiones.

- **Grupo de Desarrolladores**

- (1) Experiencia de desarrollo y suficiente conocimiento del producto.
- (2) Instruyen y orientan en el uso de los procesos de revisión.

- **Desarrolladores del proceso de planificación**

- (1) Definen los criterios de entrada y salida estándar.
- (2) El producto está definido por el estándar de desarrollo.
- (3) El producto se divide en unidades manejables para la revisión.
- (3) Revisan los elementos del software
  - ✓ Plan de aseguramiento de calidad del software
  - ✓ Plan de revisión del software
  - ✓ Plan de administración de proyecto software.

- **Revisores del proceso de planificación**

- (1) Especifican tipo de revisión.
- (2) Definen procedimientos administrativos.
- (3) Asignan responsabilidades del proceso de revisión para
  - ✓ Planificación
  - ✓ Especificación
  - ✓ Administración
  - ✓ Mantenimiento.

#### **1.4.1.2. Requisitos previos al Proceso de Auditoría**

El objetivo de la auditoría de software es proveer una evaluación objetiva de los productos y procesos que confirmen el cumplimiento de la norma, guías, especificaciones y procedimientos.

Requisitos previos para alcanzar el objetivo:

- (1) Criterio de auditoría objetivo
  - ✓ Contratos
  - ✓ Planes
  - ✓ Especificaciones
  - ✓ Normas
- (2) Seleccionar un equipo de auditoría y deben ser externos.
- (3) Dar autoridad al personal de auditoría.

#### **1.4.1.3. Plantilla de Descripción Procedimental**

Plantilla para planear, preparar, y ejecutar cualquier revisión o auditoría

- (1) Objetivo: Metas.
- (2) Resumen: Apreciación global.
- (3) Responsabilidades Especiales
- (4) Entrada: Productos e información.
- (5) Criterios de la Entrada: Condiciones antes de empezar.
- (6) Procedimientos: Pasos.
  - Planificación
  - Apreciación global
  - Preparación
  - Revisión
- (7) Criterios de la Salida: Condiciones satisfechas antes de completar el proceso.
- (8) Salida: Resultados mínimos para la realización del proceso.
- (9) Auditoría: Descripción de evidencias para continuar el proceso.

### **1.4.2. EL PROCESO DE REVISIÓN GERENCIAL**

#### **1.4.2.1. Objetivos**

Proveer recomendaciones para

- ❖ Hacer actividades progresivas
- ❖ Cambiar la dirección del proyecto o hacer planificación alternativa
- ❖ Mantener control global

#### **1.4.2.2. Resumen**

- ❖ Evaluar el nivel o estado del proyecto con relación al plan.

- ❖ Identificar y registrar cada área del problema.

#### **1.4.2.3. Responsabilidades Especiales**

- ❖ El jefe de la revisión
  - Realizar tareas administrativas
  - Asegurar una revisión ordenada
  - Asegurar reporte de la revisión
- ❖ Responsabilidades individuales de tener a disposición el estado del proyecto y toda su documentación
- ❖ Responsabilidades del equipo de revisión:
  - Estar preparados para las reuniones
  - Asegurar revisión objetiva.

#### **1.4.2.4. Entradas**

- ❖ Objetivos
- ❖ Productos
- ❖ Horario y costo.
- ❖ Reportes
- ❖ Informes de recursos asignados
- ❖ Datos sobre elementos del software

#### **1.4.2.5. Criterios de Entrada**

- **Autorización.**

Se establece en el documento de planificación del proyecto

- **Comienzo del evento**

Ocurre cuando el jefe de la revisión:

- ❖ Establece objetivos para la reunión
- ❖ Considera que los elementos del software o documentación o informes están completos y disponibles.

#### **1.4.2.6. Procedimientos**

- **Planificación.**

El jefe de revisión y el director del proyecto

- ❖ Identifican el equipo de revisión
- ❖ Definen horario y lugar de reunión
- ❖ Distribuyen el material de entrada y dejan un tiempo adecuado para la preparación

- **Apreciación global**

- **Preparación individual para reunión de revisión**

- **Examen**

- ❖ Determinar si este cumple con el plan.
- ❖ Determinar si cubre factores externos e internos no considerados en el plan

- ❖ Registrar desviaciones
- ❖ Generar lista de productos y recomendaciones.
- ❖ Recomendar cursos de acción
- ❖ Recomendar autorización para revisiones o auditorias adicionales.
- ❖ Identificar otros productos.
- **Reformas**

#### **1.4.2.7. Criterios de Salida**

- ❖ Dirección de todos los productos identificados en la revisión de objetivos.
- ❖ Emisión del reporte de la revisión gerencial

#### **1.4.2.8. Salida**

Reporte de Revisión de la administración que identifica:

- ❖ El proyecto
- ❖ El equipo de revisión
- ❖ Entradas de la revisión
- ❖ Objetivos de la revisión
- ❖ Propiedad y estado del ítem
- ❖ Lista de productos y recomendaciones
- ❖ Recomendaciones de cualquier revisión y auditoria futura

#### **1.4.2.9. Auditoria**

Informe de seguimiento

### **1.4.3. EL PROCESO DE REVISIÓN TÉCNICA**

#### **1.4.3.1. Objetivos**

Evaluar un elemento del software específico

- ❖ Probar que coincida con sus características técnicas
- ❖ Probar que el desarrollo o mantenimiento se hace acorde a diseños, normas, y pautas aplicables al proyecto.
- ❖ Probar que los cambios son implementados apropiadamente y afectan solo las áreas identificadas por la especificación del cambio.

#### **1.4.3.2. Resumen**

- ❖ Identificar discrepancias de especificaciones y estándares
- ❖ Proveer recomendaciones

#### **1.4.3.3. Responsabilidades Especiales**

- ❖ Jefe.
  - Realizar tareas administrativas
  - Asegurar una revisión ordenada
  - Asegurar reporte de la revisión

- ❖ Registrador
  - Documentar Mantenimiento
    - Defectos
    - Inconsistencias
    - Omisiones
    - Ambigüedades
  - Documentar Decisiones
  - Documentar Recomendaciones
- ❖ Miembros del equipo.
  - Son responsables de su propia preparación
  - Aseguran el cumplimiento de objetivos
  - Formulan recomendaciones.

#### **1.4.3.4. Entrada**

- ❖ Objetivos.
- ❖ Elementos software.
- ❖ Especificaciones para los elementos software.
- ❖ Planes, estándares o pautas.
- ❖ Material de referencia requerido por el jefe de revisión.

#### **1.4.3.5. Criterios de Entrada**

- **Autorización**

Esta definido por el documento de la planificación.

- **Comienzo del evento**

Ocurre cuando:

- ❖ Se declaran objetivos
- ❖ Las responsabilidades individuales muestran facilidad para la revisión.
- ❖ El jefe considera los elementos del software completos para una revisión técnica que valga la pena.

#### **1.4.3.6. Procedimientos**

- **Planificación:**

El jefe es responsable de.

- ❖ Identificar equipo de revisión.
- ❖ Programar y anunciar el lugar de reunión.
- ❖ Distribuir los materiales de entrada para los participantes, permitiendo tiempo para su preparación.

- **Apreciación global.**

- **Preparación individual para reunión de revisión**

- **Examen**

- ❖ Examinar los elementos software y verificar que cumplan con las especificaciones y estándares. Se registran las desviaciones.

- ❖ Documentar productos técnicos, recomendaciones relacionadas y el responsable de hacer las ediciones.
- ❖ Identificar otras emisiones.
- ❖ Generar un informe para documentar la reunión que contenga
  - Lista de deficiencias
  - Recomendaciones para la administración.

#### **1.4.3.7. Criterio de Salida**

- ❖ Cuando se han registrado los sucesos identificados en los objetivos
- ❖ Cuando se ha terminado el reporte de revisión técnica

#### **1.4.3.8. Salida**

Reporte de revisión técnica que identifica:

- ❖ Los miembros del grupo de revisión
- ❖ Los elementos de software revisados
- ❖ Entradas específicas para la revisión
- ❖ Lista de deficiencias de los elementos del software sin resolver
- ❖ Lista de administración de emisiones
- ❖ Ítem de acción de pertenencia y estado
- ❖ Recomendación del grupo de revisión para acabar con las deficiencias.

#### **1.4.3.9. Auditoria**

Informe de seguimiento

### **1.4.4. EL PROCESO DE INSPECCIÓN DEL SOFTWARE**

#### **1.4.4.1. Objetivo**

Descubrir e identificar los defectos de los elementos del software para:

- ❖ Verificar que los elementos del software satisfacen las características técnicas.
- ❖ Verificar que los elementos del software conforman normas aplicables
- ❖ Identificar desviación de normas y especificaciones
- ❖ Reunir datos de administración del software
- ❖ No examinar alternativas o emisiones estilizadas.

#### **1.4.4.2. Resumen**

- ❖ La inspección del software está guiada por un moderador imparcial que no es el autor de los elementos del software.
- ❖ Almacenar los datos de defectos en una base de datos de inspección.
- ❖ Presentar los requisitos mínimos para la colección de los datos

#### **1.4.4.3. Responsabilidades Especiales**

Todos los miembros del grupo son inspectores:

- **Moderador**
  - ❖ Planear y dirigir la reunión
  - ❖ Producir el reporte de inspección.
- **Lector**

Guiar el grupo sobre los elementos del software en forma lógica y comprensiva.
- **Registrador**
  - ❖ Documentar los defectos detectados en las reuniones
  - ❖ Registrar los datos de inspección requeridos para el proceso de análisis.
- **Inspector**
  - ❖ Identificar y describir defectos en los elementos del software
  - ❖ Conocer el proceso de inspección.
  - ❖ Representar diferentes puntos de vista
- **Autor**

#### 1.4.4.4. Entradas

- ❖ Elementos de software que deben satisfacer criterios de entrada.
- ❖ Especificación del elemento de software aceptado.
- ❖ La lista de inspección
- ❖ Estándar y pautas.
- ❖ Formas necesarias para informar de la inspección.
- ❖ La lista de defectos de la inspección previa (Reinspección)

#### 1.4.4.5. Criterio de Entrada

- **Autorización**

Esta definido por el documento de la planificación.
- **Comienzo del evento**
  - ❖ Disponibilidad de los elementos del Software
  - ❖ Acatamiento del diseño del proyecto
  - ❖ Aprobación del horario SQAP o SVVP
  - ❖ Revisión programada
  - ❖ Al requerir administración funcional, administración del proyecto, administración de la calidad del software o ingeniería del software.
- **Criterios de la entrada mínimos**

Antes de planear la inspección:

  - ❖ Los elementos del software conforme a las normas de contenido y formato del proyecto
  - ❖ Se satisface todo hito anterior, así como identificado en el documento de planificación apropiado.
  - ❖ Documentación de soporte disponible
  - ❖ Satisfacer todos los ítem anotados en la lista de defectos

#### 1.4.4.6. Procedimientos

- **Planificación**
  - ❖ El autor ensambla los materiales en el paquete de inspección para el moderador
  - ❖ El moderador es responsable de:
    - Asegurar que los materiales satisfagan el criterio de entrada
    - Asegurar la selección del grupo de inspección y la asignación de sus papeles en la reunión de inspección
    - Programar los encuentros de inspección
    - Distribuir materiales de inspección.
- **Apreciación global:**
  - ❖ Conducido por el moderador y el autor que hace la presentación.
  - ❖ Se usa para educar a otros inspectores acerca de los elementos del software
  - ❖ Puede asistir cualquier personal del proyecto.
- **Preparación individual para la reunión de revisión**
- **Examen**
  - **Introducción.**
    - El moderador presenta los participantes y describe sus roles.
    - Los inspectores dirigen sus comentarios al lector.
    - El propósito es detectar defectos y resolver inquietudes
  - **Establecer preparativos.**

Ocurre si los inspectores no están adecuadamente preparados.
  - **Revisar la lista de Inspección**
  - **Leer elementos del Software y registrar defectos**
    - ❖ El lector presenta el material al equipo de inspección.
    - ❖ El equipo de inspección examina el software
    - ❖ El moderador crea lista de defectos.
    - ❖ El registrador inserta cada defecto, localización, descripción y clasificación en la lista de defectos
    - ❖ El autor responde preguntas específicas y contribuye a detectar defectos.
  - **Revisar la lista de defectos**
  - **Tomar decisiones de salida**
    - ❖ Brindar una clausura sin ambigüedades
    - ❖ Determinar si el material reúne el criterio de éxito de inspección
    - ❖ Prescribir verificación apropiada
    - ❖ Identificar la disposición de los elementos del software para
      - Aceptarlo
      - Verificarlo
      - Reinspeccionarlo
  - **Repetir el trabajo**
  - **Continuar**

- ❖ Repetir la verificación por disposición de la inspección.
- ❖ Informando datos de inspección.

#### **1.4.4.7. Criterio de Salida**

- ❖ Tener resueltos todos los defectos encontrados.
- ❖ Asegurar que el material coincide con el criterio de salida
- ❖ Asegurar que los resultados de inspección han sido reportados.

#### **1.4.4.8. Salida**

Reportes:

- ❖ Lista de defectos
  - Localización
  - Descripción
  - Categorización
- ❖ Resumen del defecto (total de defectos por cada categoría).
- ❖ Reporte de inspección:
  - Numero de participantes
  - Duración de la reunión
  - Tamaño de los materiales
  - Tiempo total de preparación del equipo de inspección
  - Disposición del elemento de software
  - Estimación del esfuerzo y fecha de realización de la reanudación del trabajo.

#### **1.4.4.9. Auditoria**

- ❖ Procedimientos de inspección documentados
- ❖ Reportes de inspección retenidos
- ❖ Datos de defectos de la inspección retenidos

#### **○ Requisitos de colección de datos**

- El tipo de defecto
- La clase de defecto
- Las clases de defectos adicionales
  - ❖ Ambigüedad
  - ❖ Inconsistencia
- Defectos agrupados por gravedad
  - ❖ Mayor
  - ❖ Menor

### **1.4.5. EL PROCESO DE REVISIÓN**

#### 1.4.5.1. **Objetivo**

Evaluar un elemento de software

- ❖ Encontrar defectos, omisiones y contradicciones
- ❖ Perfeccionar los elementos del software
- ❖ Considerar implementaciones alternativas.
- ❖ Intercambiar variación de estilo y técnicas y educación de los participantes
- ❖ Descubrir problemas
  - Eficiencia y calidad en el código
  - Modularidad en diseño
  - Especificaciones de diseño sin probar.

#### 1.4.5.2. **Resumen**

- ❖ El autor presenta los elementos del software bajo revisión
- ❖ Los participantes realizan una discusión general
- ❖ El presentador detalla el elemento del software
  - Progreso de la revisión
  - Errores
  - Cambios sugeridos
  - Mejoras.

#### 1.4.5.3. **Responsabilidades Especiales**

- **Moderador**
  - ❖ Conduce la revisión de manera ordenada
  - ❖ Maneja las tareas administrativas
  - ❖ Prepara los objetivos para guiar al equipo a través de la revisión.
- **Registrador**

Escribe todos los comentarios

  - ❖ Errores encontrados
  - ❖ Preguntas de estilo
  - ❖ Omisiones
  - ❖ Contradicciones
  - ❖ Sugerencias para improvisación
  - ❖ Propuestas alternativas
- **Autor**
  - ❖ Responsable de que se examinen los elementos del software
  - ❖ Presenta los materiales.
- **Miembros del equipo de revisión**
  - ❖ Responsables de revisar cualquier material de entrada previo a la revisión
  - ❖ Participar durante la revisión para asegurar que se están cumpliendo los objetivos

#### 1.4.5.4. Entradas

- ❖ Declaración de objetivos
- ❖ Elemento software
- ❖ Estándares vigentes para el desarrollo del elemento software.
- ❖ c

#### 1.4.5.5. Criterio de Entrada

##### ○ Autorización

Se establece en el documento de planificación del proyecto

##### ○ Comienzo del evento

- ❖ El autor de los elementos software ha indicado que esta listo, y esta preparado para presentar los materiales.
- ❖ El moderador esta equipado por practicas o estándares locales y ha concluido que el elemento software es suficientemente completo para conducir la revisión.

#### 1.4.5.6. Procedimientos

##### ○ Planificación

El moderador:

- ❖ Identifica el equipo de revisión
- ❖ Programa la reunión y selecciona el lugar de encuentro.
- ❖ Distribuye todo el material de entrada necesario a los participantes, permitiendo el tiempo de preparación adecuado.

##### ○ Apreciación global

##### ○ Preparación

Los participantes

- ❖ Revisan el material de entrada
- ❖ Preparan una lista de inquietudes y eventos a discutir.
- Examen
- ❖ El presentador hace una presentación rápida de los elementos de software.
- ❖ El autor recorre los elementos de software para que los miembros del equipo puedan hacer preguntas o crear discusiones o documentar su importancia.
- ❖ El registrador escribe los comentarios y decisiones para incluirlas en el informe de revisión.
- ❖ El equipo de revisión puede recomendar una revisión de refuerzo

#### 1.4.5.7. Criterio de Salida

- ❖ El total de elementos de software han sido repasados en detalle
- ❖ Todas las deficiencias, omisiones, y sugerencias para mejoramiento han sido anotadas.

- ❖ El informe de apreciación global ha sido producido.

#### **1.4.5.8. Salida**

Informe de revisión que contiene:

- ❖ Identificación de equipo de revisión
- ❖ Identificación de los elementos de software examinados.
- ❖ La declaración de objetivos que fueron seguidos durante la reunión de revisión
- ❖ Una lista de deficiencias anotadas, omisiones, contradicciones y sugerencias para mejorar
- ❖ Cualquier recomendación hecha por el equipo de revisión en como acabar con las deficiencias y problemas no resueltos.
- ❖ Sugerencias de las revisiones de refuerzo.

#### **1.4.5.9. Auditoria**

El moderador de la revisión archiva estos documentos

- ❖ El reporte de revisión
- ❖ Minutos de reunión
- ❖ Materiales en los cuales las conclusiones están basadas

### **1.4.6. EL PROCESO DE AUDITORIA**

#### **1.4.6.1. Objetivo**

Proveer una conformidad de productos y procesos

#### **1.4.6.2. Resumen**

- ❖ Establecer procedimientos para conducir la auditoria y reforzar la acción en los descubrimientos de auditoria.
- ❖ Evaluar los elementos de software y los procesos para producir un criterio de auditoria objetivo comparado con requerimientos, planes especificaciones o procedimientos, guías y estándares.
- ❖ Documentar los resultados
  - Lista de los ítem que no cumplen
  - Otros productos para revisión y acción subsiguiente.
  - Recomendaciones

#### **1.4.6.3. Responsabilidades Especiales**

- ❖ Jefe
  - Organizar y dirigir la auditoria
  - Coordinar la preparación y edición del informe de auditoria.
  - Asegurar que el equipo de auditoria esta preparado para conducirla
  - Asegurar que los procedimientos de auditoria son desarrollados y producen informes en concordancia con el alcance de la auditoria.

#### **1.4.6.4. Entrada**

- ❖ El propósito y alcance de la auditoria
- ❖ Criterio de auditoria objetiva, tales como compendiar requerimientos, planes, especificaciones, procedimientos, pautas y normas
- ❖ Los elementos y procesos de software a ser auditados y cualquier historia pertinente.
- ❖ Información de antecedentes respecto a responsable de la organización de los productos y procesos auditados

#### **1.4.6.5. Criterios de Entrada**

Uno de los siguientes eventos

- ❖ Cuando un proyecto pilar especial ha sido rechazado, la auditoria es iniciada por planes mas tempranos
- ❖ Cuando partes externas solicitan una auditoria en una fecha calendario especifico o proyecto pilar.
- ❖ Cuando un elemento de organización local ha requerido la auditoria, estableciendo una necesidad clara y especifica.
- ❖ Cuando un proyecto pilar especial, fecha calendario, u otro criterio que ha sido encontrado como parte de la carta constitucional de la organización intervenida, en respuesta se inicia la auditoria

#### **1.4.6.6. Procedimientos**

##### **o Planificación**

- ❖ Procesos del proyecto y el tiempo de ajuste para observación del equipo de auditoria.
- ❖ Software requerido y su disponibilidad
- ❖ Las máscaras de los reportes son identificados
- ❖ Distribución de reportes.
- ❖ Flujo de actividades requeridas.
- ❖ Requerimientos
- ❖ Criterio objetivo de auditoria
- ❖ Lista de chequeo y procedimientos de la auditoria.
- ❖ Personal de la auditoria
  - Número requerido
  - Habilidades
  - Experiencia
  - Responsabilidades.
- ❖ Organizaciones involucradas
- ❖ Fecha, tiempo, lugar, agenda, y público intencional de sesión de apreciación global

##### **o Apreciación global**

Guiada por el jefe de la auditoria, provee lo siguiente:

- ❖ De los acuerdos existentes
- ❖ De productos y procesos que intervienen
- ❖ De procesos de auditoría, sus objetivos, y salidas
- ❖ Contribuciones esperadas de la organización que interviene en el proceso de auditoría
  - Número personas a ser entrevistadas
  - Facilidades para la reunión
- ❖ Especificar horario de la auditoría.

○ **Preparación**

El equipo de auditoría

- ❖ Entender la organización.
- ❖ Entender los productos y procesos
- ❖ Entender el criterio de auditoría objetiva
- ❖ Preparar el reporte de auditoría
- ❖ Detallar el plan de auditoría

El jefe del equipo debe

- ❖ Orientar e instruir el equipo como se requiera
- ❖ Facilitar las entrevistas de la auditoría.
- ❖ Facilitar materiales, documentos y herramientas requeridas para los procedimientos de la auditoría
- ❖ Facilitar elementos del software
- ❖ Programar las entrevistas.

○ **Examen**

- ❖ Revisar procedimientos e instrucciones
- ❖ Examinar la estructura del trabajo
- ❖ Examinar evidencia de implementación y controles balanceados
- ❖ El personal entrevistado determina el estado y funcionamiento de los procesos y de los productos
- ❖ Examinar documentos de los elementos.
- ❖ Probar los elementos.

○ **Informes**

Resultado anterior de los resultados formales para

- ❖ Resolver cualquier inquietud o ambigüedad mientras mantiene objetividad y correcciones.
- ❖ Agregar consistencia al nivel de reporte detallado
- ❖ Agregar cualquier información recientemente verificada.

**1.4.6.7. Criterio de salida**

Cuando

- ❖ Cada elemento (s) dentro de la mira de la auditoría se ha examinado
- ❖ Los hallazgos han sido presentados al auditor de la organización

- ❖ Se tiene respuesta del bosquejo de los hallazgos recibidos y evaluados
- ❖ Los hallazgos finales han sido presentados formalmente al auditor de la organización y a la entidad
- ❖ Los informes de auditoria han sido preparados y sometidos a destinatarios designados en el plan de la auditoria
- ❖ El informe de recomendaciones, si es requerido por el plan, ha sido preparado y sometido a los designados en el plan de la auditoria
- ❖ Todo el flujo de las acciones de la organización incluida en la meta (o contrato) de la auditoria ha sido ejecutado

#### **1.4.6.8. Salida**

Reportes de auditoria final deben, como mínimo, contener lo siguiente:

- ❖ Identificación de la auditoria.
  - Título del informe
  - Auditor de la organización
  - Organización auditada
  - Fecha de la auditoria.
- ❖ Alcance.
  - Enumeración de los estándares
  - Especificaciones
  - Prácticas
  - Procedimientos que constituyen el criterio objetivo de auditoria, contra los cuales se condujo la auditoria de los elementos del software y procesos auditados.
- ❖ Conclusiones
- ❖ Sinopsis. Lista de todos los elementos del software auditados, procesos, y hallazgos.
- ❖ Flujo. Tipo y tiempo de las actividades de la auditoria.
- ❖ Recomendaciones. Deben ser reportadas por separado de los resultados.

#### **1.4.6.9. Auditoria**

El material de documentación del proceso que incluye lo siguiente:

- ❖ Todos los programas de trabajo, listas de chequeo, etc.
- ❖ Equipo de trabajo
- ❖ Notas de interventoría, notas de observaciones.
- ❖ Completas pruebas de evidencia
- ❖ Copia de ítem examinados, anotados
- ❖ Informes del proyecto con respuesta del auditor de la organización
- ❖ Memorando del flujo como sea necesario

### **1.5. ANÁLISIS DE LA NORMA IEEE 1028-1.988**

### **1.5.1. QUIÉN LA APLICA**

La norma IEEE 1028-1.988 es aplicada por las organizaciones voluntariamente usando los procedimientos definidos mediante un formato

### **1.5.2. QUÉ ES**

La IEEE 1028-1.988 es una norma para integrar actividades de revisión y auditoría del software durante todas las etapas de su ciclo de vida

### **1.5.3. PARA QUÉ SIRVE**

Sirve para realizar revisiones de administración, revisiones técnicas, inspección del software y auditoría.

### **1.5.4. CUÁNDO SE APLICA**

La norma IEEE 1028-1.988 se aplica cuando se ejecutan procesos de revisión y auditoría aplicado para software crítico Revisión de requerimientos del software(SRR)

- ❖ Revisión del diseño preliminar (PDR)
- ❖ Revisión del diseño crítico (CDR)
- ❖ Revisión del plan de verificación y validación del software (SVVPR)
- ❖ Auditoría funcional (FA)
- ❖ Auditoría física (PA)
- ❖ Auditoría en proceso (IPA)
- ❖ Revisión de Dirección (MR)

### **1.5.5. FRECUENCIA**

Normalmente se aplica cuando una falla causa impacto de seguridad, o aumento financiero o impacto social, sin embargo se debe aplicar durante todas las etapas del ciclo de vida de los sistemas

### **1.5.6. RECURSOS**

El tiempo y dinero dependen de cada proyecto, y exigen mayor recurso cuando se hace revisión y auditoría de sistemas (hardware y software). Las

empresas poco invierten recursos para auditoria porque se requiere de grandes esfuerzos, sin embargo muy pocas veces se analiza que el beneficio que se obtiene es superior.

### 1.5.7. DESARROLLO DE LA NORMA IEEE 1028

#### **Cómo realizar revisiones y auditorias en proyectos de desarrollo de software**

##### INTRODUCCIÓN

**Cómo realizar revisiones y auditorias** es una versión simplificada de la norma IEEE-1028-1988 – Estándar para revisión y auditoria de software

Este cómo abarca la gama de las actividades de revisión y auditoria comprendidas durante un proyecto.

La revisión y auditoria comprende los siguientes cinco (5) módulos:

**Revisión gerencial.** La evaluación formal del plan del proyecto o el estado del proyecto en relación con lo que plantea el equipo de revisión asignado

**Revisión técnica.** La evaluación de los módulos del software especificados y documentación para asegurar que los módulos del software y los documentos cumplan con los estándares aplicados mientras se conforman las especificaciones.

La revisión técnica puede proveer recomendaciones después examinar las alternativas.

**Verificación.** Es un proceso de evaluación que puede dar lugar a las recomendaciones para mejorar o identificar alternativas de los módulos del software actual o documentos que están siendo desarrollados

**Auditoria.** Provee evidencia objetiva de la conformidad de los productos y procesos con los estándares, pautas, especificaciones y procedimientos. Incluye auditoria del sistema de gerencia de la calidad.

**Inspección.** Evaluación formal rigurosa diseñada para detectar e identificar defectos en el material revisado.

Normalmente conducido después del acontecimiento e iniciado por personas fuera del equipo del proyecto

##### ALCANCE

**Cómo ejecutar revisión y auditoria** aplica para toda la documentación del proyecto, revisiones o auditorias especificas de la conducta del software cuyas necesidades son definidas por el proyecto y los planes de calidad. También se aplica a la auditoria de del sistema de gerencia de calidad.

##### OBJETIVOS

**Cómo ejecutar revisiones y auditorias** proporciona definiciones y requerimientos uniformes para permitir al personal del proyecto realizar las revisiones y auditorias necesarias de productos y procesos.

**La contribución es calidad.** Un elemento esencial en el desarrollo de la documentación del proyecto es la verificación y validación que su contenido es verdadero y exacto. Se conoce en la literatura de la calidad del software que los autores de documentos y códigos intrínsecamente no desean encontrar errores en su propio trabajo.

Ellos a menudo consciente o inconscientemente pasan por alto errores e inexactitudes. Por lo tanto no es recomendable dejar la revisión de la documentación del proyecto a los mismo que lo producen. Es importante para dicha comprobación conducirla de una manera constante y sistémica. Si se hace esto, la documentación del proyecto servirá al desarrollo del proceso de la manera en la cual se piensa

## 2. MODELO TEÓRICO

### PLANTILLA NORMA IEEE 1028-1.988 REVISIÓN Y AUDITORIA DE SOFTWARE

#### 1. EL PROCESO DE REVISIÓN DE ADMINISTRACIÓN

##### 1.1. OBJETIVOS. PROVEER RECOMENDACIONES PARA

1.1.1. HACER ACTIVIDADES PROGRESIVAS

1.1.2. CAMBIAR LA DIRECCIÓN DEL PROYECTO O HACER PLANIFICACIÓN ALTERNATIVA

1.1.3. MANTENER CONTROL GLOBAL

##### 1.2 RESUMEN

1.2.1. EVALUAR EL NIVEL O ESTADO DEL PROYECTO CON RELACIÓN AL PLAN.

1.2.2. IDENTIFICAR Y REGISTRAR CADA ÁREA DEL PROBLEMA.

##### 1.3 RESPONSABILIDADES ESPECIALES.

###### 1.3.1. EL JEFE DE LA REVISIÓN

1.3.1.1. REALIZAR TAREAS ADMINISTRATIVAS

1.3.1.2. ASEGURAR UNA REVISIÓN ORDENADA

1.3.1.3. ASEGURAR REPORTE DE LA REVISIÓN

1.3.2. RESPONSABILIDADES INDIVIDUALES DE TENER A DISPOSICIÓN EL ESTADO DEL PROYECTO Y TODA SU DOCUMENTACIÓN

1.3.3. RESPONSABILIDADES DEL EQUIPO DE REVISIÓN:

1.3.3.1. ESTAR PREPARADOS PARA LAS REUNIONES

1.3.3.2. ASEGURAR REVISIÓN OBJETIVA.

##### 1.4 ENTRADAS

1.4.1. OBJETIVOS

1.4.2. PRODUCTOS

1.4.3. HORARIO Y COSTO.

1.4.4. REPORTES

1.4.5. INFORMES DE RECURSOS ASIGNADOS

1.4.6. DATOS SOBRE ELEMENTOS DEL SOFTWARE

##### 1.5 CRITERIOS DE LA ENTRADA

1.5.1. AUTORIZACIÓN. SE ESTABLECE EN EL DOCUMENTO DE PLANIFICACIÓN DEL PROYECTO

1.5.2. COMIENZO DEL EVENTO OCURRE CUANDO EL JEFE DE LA REVISIÓN:

1.5.2.1. ESTABLECE OBJETIVOS PARA LA REUNIÓN

1.5.2.2. CONSIDERA QUE LOS ELEMENTOS DEL SOFTWARE O DOCUMENTACIÓN O INFORMES ESTÁN COMPLETOS Y DISPONIBLES.

## 1.6 PROCEDIMIENTOS.

1.6.1 PLANIFICACIÓN. EL JEFE DE REVISIÓN Y EL DIRECTOR DEL PROYECTO

1.6.1.1. IDENTIFICAN EL EQUIPO DE REVISIÓN

1.6.1.2. DEFINEN HORARIO Y LUGAR DE REUNIÓN

1.6.1.3. DISTRIBUYEN EL MATERIAL DE ENTRADA Y DEJAN UN TIEMPO ADECUADO PARA LA PREPARACIÓN

1.6.2 APRECIACIÓN GLOBAL

1.6.3 PREPARACIÓN INDIVIDUAL PARA REUNIÓN DE REVISIÓN

1.6.4 EXAMEN

1.6.4.1. DETERMINAR SI ESTE CUMPLE CON EL PLAN.

1.6.4.2. DETERMINAR SI CUBRE FACTORES EXTERNOS E INTERNOS NO CONSIDERADOS EN EL PLAN

1.6.4.3. REGISTRAR DESVIACIONES

1.6.4.4. GENERAR LISTA DE PRODUCTOS Y RECOMENDACIONES.

1.6.4.5. RECOMENDAR CURSOS DE ACCIÓN

1.6.4.6. RECOMENDAR AUTORIZACIÓN PARA REVISIONES O AUDITORIAS ADICIONALES.

1.6.4.7. IDENTIFICAR OTROS PRODUCTOS.

1.6.5 REFORMAS

## 1.7 CRITERIOS DE SALIDA

1.7.1. DIRECCIÓN DE TODOS LOS PRODUCTOS IDENTIFICADOS EN LA REVISIÓN DE OBJETIVOS.

1.7.2. EMISIÓN DEL REPORTE DE LA REVISIÓN DE LA ADMINISTRACIÓN.

## 1.8 SALIDA: REPORTE DE REVISIÓN DE LA ADMINISTRACIÓN QUE IDENTIFICA:

1.8.1. EL PROYECTO

1.8.2. EL EQUIPO DE REVISIÓN

1.8.3. ENTRADAS DE LA REVISIÓN

1.8.4. OBJETIVOS DE LA REVISIÓN

1.8.5. PROPIEDAD Y ESTADO DEL ÍTEM

1.8.6. LISTA DE PRODUCTOS Y RECOMENDACIONES

1.8.7. RECOMENDACIONES DE CUALQUIER REVISIÓN Y AUDITORIA FUTURA

## 1.9 AUDITORIA. INFORME TRACEABLE

## 2. EL PROCESO DE REVISIÓN TÉCNICA

2.1. OBJETIVOS. EVALUAR UN ELEMENTO DEL SOFTWARE ESPECÍFICO

- 2.1.1. PROBAR QUE COINCIDA CON SUS CARACTERÍSTICAS TÉCNICAS
- 2.1.2. PROBAR QUE EL DESARROLLO O MANTENIMIENTO SE HACE ACORDE A DISEÑOS, NORMAS, Y PAUTAS APLICABLES AL PROYECTO.
- 2.1.3. PROBAR QUE LOS CAMBIOS SON IMPLEMENTADOS APROPIADAMENTE Y AFECTAN SOLO LAS ÁREAS IDENTIFICADAS POR LA ESPECIFICACIÓN DEL CAMBIO.

## 2.2. RESUMEN.

- 2.2.1. IDENTIFICAR DISCREPANCIAS DE ESPECIFICACIONES Y ESTÁNDARES
- 2.2.2. PROVEER RECOMENDACIONES

## 2.3. RESPONSABILIDAD ESPECIAL.

### 2.3.1. JEFE.

- 2.3.1.1. REALIZAR TAREAS ADMINISTRATIVAS
- 2.3.1.2. ASEGURAR UNA REVISIÓN ORDENADA
- 2.3.1.3. ASEGURAR REPORTE DE LA REVISIÓN

### 2.3.2. REGISTRADOR

#### 2.3.2.1. DOCUMENTAR MANTENIMIENTO

- 2.3.2.1.1. DEFECTOS
- 2.3.2.1.2. INCONSISTENCIAS
- 2.3.2.1.3. OMISIONES

#### 2.3.2.1.4. AMBIGÜEDADES

#### 2.3.2.2. DOCUMENTAR DECISIONES

#### 2.3.2.3. DOCUMENTAR RECOMENDACIONES

### 2.3.3 MEMBROS DEL EQUIPO.

#### 2.3.3.1. SON RESPONSABLES DE SU PROPIA PREPARACIÓN

#### 2.3.3.2. ASEGURAN EL CUMPLIMIENTO DE OBJETIVOS

#### 2.3.3.3. FORMULAN RECOMENDACIONES.

## 2.4 ENTRADA.

### 2.4.1. OBJETIVOS.

### 2.4.2. ELEMENTOS SOFTWARE .

### 2.4.3. ESPECIFICACIONES PARA LOS ELEMENTOS SOFTWARE .

### 2.4.4. PLANES, ESTÁNDARES O PAUTAS.

### 2.4.5. MATERIAL DE REFERENCIA REQUERIDO POR EL JEFE DE REVISIÓN.

## 2.5. CRITERIOS DE ENTRADA.

### 2.5.1. AUTORIZACIÓN: ESTA DEFINIDO POR EL DOCUMENTO DE LA PLANIFICACIÓN.

### 2.5.2. COMIENZO DEL EVENTO OCURRE CUANDO:

#### 2.5.2.1. SE DECLARAN OBJETIVOS

#### 2.5.2.2. LAS RESPONSABILIDADES INDIVIDUALES MUESTRAN FACILIDAD PARA LA REVISIÓN.

#### 2.5.2.3. EL JEFE CONSIDERA LOS ELEMENTOS DEL SOFTWARE COMPLETOS PARA UNA REVISIÓN TÉCNICA QUE VALGA LA PENA .

## 2.6. PROCEDIMIENTOS

2.6.1. PLANIFICACIÓN: EL JEFE ES RESPONSABLE DE .

2.6.1.1. IDENTIFICAR EQUIPO DE REVISIÓN.

2.6.1.2. PROGRAMAR Y ANUNCIAR EL LUGAR DE REUNIÓN.

2.6.1.3. DISTRIBUIR LOS MATERIALES DE ENTRADA PARA LOS PARTICIPANTES, PERMITIENDO TIEMPO PARA SU PREPARACIÓN.

2.6.2 APRECIACIÓN GLOBAL

2.6.3 PREPARACIÓN INDIVIDUAL PARA REUNIÓN DE REVISIÓN

2.6.4. EXAMEN

2.6.4.1. EXAMINAR LOS ELEMENTOS SOFTWARE Y VERIFICAR QUE CUMPLAN CON LAS ESPECIFICACIONES Y ESTÁNDARES. SE REGISTRAN LAS DESVIACIONES.

2.6.4.2. DOCUMENTAR PRODUCTOS TÉCNICOS, RECOMENDACIONES RELACIONADAS Y EL RESPONSABLE DE HACER LAS EDICIONES.

2.6.4.3. IDENTIFICAR OTRAS EMISIONES.

2.6.4.4. GENERAR UN INFORME PARA DOCUMENTAR LA REUNIÓN QUE CONTenga

2.6.4.4.1. LISTA DE DEFICIENCIAS

2.6.4.4.2. RECOMENDACIONES PARA LA ADMINISTRACIÓN.

2.7. CRITERIO DE SALIDA .

2.7.1. CUANDO SE HAN REGISTRADO LOS SUCESOS IDENTIFICADOS EN LOS OBJETIVOS

2.7.2. CUANDO SE HA TERMINADO EL REPORTE DE REVISIÓN TÉCNICA

2.8. SALIDA. REPORTE DE REVISIÓN TÉCNICA QUE IDENTIFICA:

2.8.1. LOS MIEMBROS DEL GRUPO DE REVISIÓN

2.8.2. LOS ELEMENTOS DE SOFTWARE REVISADOS

2.8.3. ENTRADAS ESPECIFICAS PARA LA REVISIÓN

2.8.4. LISTA DE DEFICIENCIAS DE LOS ELEMENTOS DEL SOFTWARE SIN RESOLVER

2.8.5. LISTA DE ADMINISTRACIÓN DE EMISIONES

2.8.6. ÍTEM DE ACCIÓN DE PERTENENCIA Y ESTADO

2.8.7. RECOMENDACIÓN DEL GRUPO DE REVISIÓN PARA ACABAR CON LAS DEFICIENCIAS.

5.9. AUDITORIA. INFORME TRACEABLE

## 3. PROCESOS DE INSPECCIÓN DEL SOFTWARE

3.1. OBJETIVO. DESCUBRIR E IDENTIFICAR LOS DEFECTOS DE LOS ELEMENTOS DEL SOFTWARE PARA:

3.1.1. VERIFICAR QUE LOS ELEMENTOS DEL SOFTWARE SATISFACE LAS CARACTERÍSTICAS TÉCNICAS.

- 3.1.2. VERIFICAR QUE LOS ELEMENTOS DEL SOFTWARE CONFORMAN NORMAS APLICABLES
- 3.1.3. IDENTIFICAR DESVIACIÓN DE NORMAS Y ESPECIFICACIONES
- 3.1.4. REUNIR DATOS DE ADMINISTRACIÓN DEL SOFTWARE
- 3.1.5. NO EXAMINAR ALTERNATIVAS O EMISIONES ESTILIZADAS.

### 3.2. RESUMEN.

- 3.2.1. LA INSPECCIÓN DEL SOFTWARE ESTÁ GUIADA POR UN MODERADOR IMPARCIAL QUE NO ES EL AUTOR DE LOS ELEMENTOS DEL SOFTWARE .
- 3.2.2. ALMACENAR LOS DATOS DE DEFECTOS EN UNA BASE DE DATOS DE INSPECCIÓN.
- 3.2.3. PRESENTAR LOS REQUISITOS MÍNIMOS PARA LA COLECCIÓN DE LOS DATOS

### 3.3. RESPONSABILIDADES ESPECIALES. TODOS LOS MIEMBROS DEL GRUPO SON INSPECTORES:

#### 3.3.1. MODERADOR

- 3.3.1.1. PLANEAR Y DIRIGIR LA REUNIÓN
- 3.3.1.2. PRODUCIR EL REPORTE DE INSPECCIÓN.

#### 3.3.2. LECTOR

- 3.3.2.1. GUIAR EL GRUPO SOBRE LOS ELEMENTOS DEL SOFTWARE EN FORMA LÓGICA Y COMPRENSIVA.

#### 3.3.3. REGISTRADOR

- 3.3.3.1. DOCUMENTAR LOS DEFECTOS DETECTADOS EN LAS REUNIONES
- 3.3.3.2. REGISTRAR LOS DATOS DE INSPECCIÓN REQUERIDOS PARA EL PROCESO DE ANÁLISIS.

#### 3.3.4. INSPECTOR

- 3.3.4.1. IDENTIFICAR Y DESCRIBIR DEFECTOS EN LOS ELEMENTOS DEL SOFTWARE
- 3.3.4.1. CONOCER EL PROCESO DE INSPECCIÓN.
- 3.3.4.2. REPRESENTAR DIFERENTES PUNTOS DE VISTA

#### 3.3.5. AUTOR

### 3.4. ENTRADAS.

- 3.4.1. ELEMENTOS DE SOFTWARE QUE DEBEN SATISFACER CRITERIOS DE ENTRADA.
- 3.4.2. ESPECIFICACIÓN DEL ELEMENTO DE SOFTWARE ACEPTADO.
- 3.4.3. LA LISTA DE INSPECCIÓN
- 3.4.4. ESTÁNDAR Y PAUTAS.
- 3.4.5. FORMAS NECESARIAS PARA INFORMAR DE LA INSPECCIÓN.
- 3.4.6. LA LISTA DE DEFECTOS DE LA INSPECCIÓN PREVIA (RE-INSPECCIÓN)

### 3.5. CRITERIO DE ENTRADA

- 3.5.1. AUTORIZACIÓN: ESTA DEFINIDO POR EL DOCUMENTO DE LA PLANIFICACIÓN.
- 3.5.2. COMIENZO DEL EVENTO
- 3.5.2.1. DISPONIBILIDAD DE LOS ELEMENTOS DEL SOFTWARE

- 3.5.2.2. ACATAMIENTO DEL DISEÑO DEL PROYECTO
- 3.5.2.3. APROBACIÓN DEL HORARIO SQAP O SVVP
- 3.5.2.4. REVISIÓN PROGRAMADA
- 3.5.2.5. AL REQUERIR ADMINISTRACIÓN FUNCIONAL, ADMINISTRACIÓN DEL PROYECTO, ADMINISTRACIÓN DE LA CALIDAD DEL SOFTWARE O INGENIERÍA DEL SOFTWARE.
- 3.5.3. CRITERIOS DE LA ENTRADA MÍNIMOS ANTES DE PLANEAR LA INSPECCIÓN:
  - 3.5.3.1. LOS ELEMENTOS DEL SOFTWARE CONFORME A LAS NORMAS DE CONTENIDO Y FORMATO DEL PROYECTO
  - 3.5.3.2. SE SATISFACE TODO HITO ANTERIOR, ASÍ COMO IDENTIFICADO EN EL DOCUMENTO DE PLANIFICACIÓN APROPIADO.
  - 3.5.3.3. DOCUMENTACIÓN DE SOPORTE DISPONIBLE
  - 3.5.3.4. SATISFACER TODOS LOS ÍTEMS ANOTADOS EN LA LISTA DE DEFECTOS
  
- 3.6. PROCEDIMIENTOS
  - 3.6.1. PLANIFICACIÓN
    - 3.6.1.1. EL AUTOR ENSAMBLA LOS MATERIALES EN EL PAQUETE DE INSPECCIÓN PARA EL MODERADOR
    - 3.6.1.2. EL MODERADOR ES RESPONSABLE DE:
      - 3.6.1.1. ASEGURAR QUE LOS MATERIALES SATISFAGAN EL CRITERIO DE ENTRADA
      - 3.1.1.2. ASEGURAR LA SELECCIÓN DEL GRUPO DE INSPECCIÓN Y LA ASIGNACIÓN DE SUS PAPELES EN LA REUNIÓN DE INSPECCIÓN
      - 3.1.1.3. PROGRAMAR LOS ENCUENTROS DE INSPECCIÓN
      - 3.1.1.4. DISTRIBUIR MATERIALES DE INSPECCIÓN.
  - 3.6.2. APRECIACIÓN GLOBAL:
    - 3.6.2.1. CONDUcido POR EL MODERADOR Y EL AUTOR QUE HACE LA PRESENTACIÓN.
    - 3.6.2.2. SE USA PARA EDUCAR A OTROS INSPECTORES ACERCA DE LOS ELEMENTOS DEL SOFTWARE
    - 3.2.2.3. PUEDE ASISTIR CUALQUIER PERSONAL DEL PROYECTO.
  - 3.6.3. PREPARACIÓN INDIVIDUAL PARA LA REUNIÓN DE REVISIÓN
  - 3.6.4. EXAMEN
    - 3.6.4.1. INTRODUCCIÓN.
      - 3.6.4.1.1. EL MODERADOR PRESENTA LOS PARTICIPANTES Y DESCRIBE SUS ROLES.
      - 3.6.4.1.2. LOS INSPECTORES DIRIGEN SUS COMENTARIOS AL LECTOR.
      - 3.6.4.1.3. EL PROPÓSITO ES DETECTAR DEFECTOS Y RESOLVER INQUIETUDES
    - 3.6.4.2. ESTABLECER PREPARATIVOS SI LOS INSPECTORES NO ESTÁN ADECUADAMENTE PREPARADOS.
    - 3.6.4.3. REVISAR LA LISTA DE INSPECCIÓN
    - 3.6.4.4. LEER ELEMENTOS DEL SOFTWARE Y REGISTRAR DEFECTOS.
      - 3.6.4.4.1. EL LECTOR PRESENTA EL MATERIAL AL EQUIPO DE INSPECCIÓN.
      - 3.6.4.4.2. EL EQUIPO DE INSPECCIÓN EXAMINA EL SOFTWARE
      - 3.6.4.4.3. EL MODERADOR CREA LISTA DE DEFECTOS.

3.6.4.4.4.EL REGISTRADOR INSERTA CADA DEFECTO, LOCALIZACIÓN, DESCRIPCIÓN Y CLASIFICACIÓN EN LA LISTA DE DEFECTOS

3.6.4.4.5.EL AUTOR RESPONDE PREGUNTAS ESPECIFICAS Y CONTRIBUYE A DETECTAR DEFECTOS.

3.6.4.5.REVISAR LA LISTA DE DEFECTOS.

3.6.4.6.TOMAR DECISIONES DE SALIDA.

3.6.4.6.1.BRINDAR UNA CLAUSURA SIN AMBIGÜEDADES

3.6.4.6.2.DETERMINAR SI EL MATERIAL REÚNE EL CRITERIO DE ÉXITO DE INSPECCIÓN

3.6.4.6.3.PRESCRIBIR VERIFICACIÓN APROPIADA

3.6.4.6.4.IDENTIFICAR LA DISPOSICIÓN DE LOS ELEMENTOS DEL SOFTWARE PARA

3.6.4.6.4.1.ACEPTARLO

3.6.4.6.4.2.VERIFICARLO

3.6.4.6.4.3.REINSPECCIONARLO

3.6.5.REPETIR EL TRABAJO

3.6.6.CONTINUAR.

3.6.6.1.REPETIR LA VERIFICACIÓN POR DISPOSICIÓN DE LA INSPECCIÓN.

3.6.6.2.INFORMANDO DATOS DE INSPECCIÓN.

3.7.CRITERIO DE SALIDA.

3.7.1.TENER RESUELTOS TODOS LOS DEFECTOS ENCONTRADOS.

3.7.2.ASEGURAR QUE EL MATERIAL COINCIDE CON EL CRITERIO DE SALIDA

3.7.3.ASEGURAR QUE LOS RESULTADOS DE INSPECCIÓN HAN SIDO REPORTADOS.

3.8.SALIDA. REPORTES:

3.8.1.LISTA DE DEFECTOS

3.8.1.1.LOCALIZACIÓN

3.8.1.2.DESCRIPCIÓN

3.8.1.3.CATEGORIZACIÓN

3.8.2.RESUMEN DEL DEFECTO (TOTAL DE DEFECTOS POR CADA CATEGORÍA).

3.8.3.REPORTE DE INSPECCIÓN:

3.8.3.1.NUMERO DE PARTICIPANTES

3.8.3.2.DURACIÓN DE LA REUNIÓN

3.8.3.3.TAMAÑO DE LOS MATERIALES

3.8.3.4.TIEMPO TOTAL DE PREPARACIÓN DEL EQUIPO DE INSPECCIÓN

3.8.3.5.DISPOSICIÓN DEL ELEMENTO DE SOFTWARE

3.8.3.6.ÉSTIMACIÓN DEL ESFUERZO Y FECHA DE REALIZACIÓN DE REANUDACIÓN DEL TRABAJO.

3.9.AUDITORIA:

3.9.1.PROCEDIMIENTOS DE INSPECCIÓN DOCUMENTADOS

3.9.2.REPORTES DE INSPECCIÓN RETENIDOS

3.9.3.DATOS DE DEFECTOS DE LA INSPECCIÓN RETENIDOS

### 3.10. REQUISITOS DE COLECCIÓN DE DATOS

#### 3.10.1. EL TIPO DE DEFECTO

#### 3.10.2. LA CLASE DE DEFECTO

#### 3.10.3. LAS CLASES DE DEFECTOS ADICIONALES

##### 3.10.3.1. AMBIGÜEDAD

##### 3.10.3.2. INCONSISTENCIA

#### 3.10.4. DEFECTOS CATEGORIZADOS POR GRAVEDAD (MAYOR O MENOR)

## 4. LOS PROCESOS DE REVISIÓN

### 4.1. OBJETIVO. EVALUAR UN ELEMENTO DE SOFTWARE

#### 4.1.1. ENCONTRAR DEFECTOS, OMISIONES Y CONTRADICCIONES

#### 4.1.2. PERFECCIONAR LOS ELEMENTOS DEL SOFTWARE

#### 4.1.3. CONSIDERAR IMPLEMENTACIONES ALTERNATIVAS.

#### 4.1.4. INTERCAMBIAR DE VARIACIÓN DE ESTILO Y TÉCNICAS Y EDUCACIÓN DE LOS PARTICIPANTES

#### 4.1.5. DESCUBRIR PROBLEMAS

##### 4.1.5.1. EFICIENCIA Y CALIDAD EN EL CÓDIGO

##### 4.1.5.2. MODULARIDAD EN DISEÑO

##### 4.1.5.3. ESPECIFICACIONES DE DISEÑO SIN PROBAR.

### 4.2. RESUMEN

#### 4.2.1. EL AUTOR PRESENTA LOS ELEMENTOS DEL SOFTWARE BAJO REVISIÓN

#### 4.2.2. LOS PARTICIPANTES REALIZAN UNA DISCUSIÓN GENERAL

#### 4.2.3. EL PRESENTADOR DETALLA EL ELEMENTO DEL SOFTWARE

##### 4.2.3.1. PROGRESO DE LA REVISIÓN

##### 4.2.3.2. ERRORES

##### 4.2.3.3. CAMBIOS SUGERIDOS

##### 4.2.3.4. MEJORAS.

### 4.2. RESPONSABILIDADES ESPECIALES:

#### 4.2.1. MODERADOR

##### 4.2.1.1. CONDUCE LA REVISIÓN DE MANERA ORDENADA

##### 4.2.1.2. MANEJA LAS TAREAS ADMINISTRATIVAS

##### 4.2.1.3. PREPARA LOS OBJETIVOS PARA GUIAR AL EQUIPO A TRAVÉS DE LA REVISIÓN.

#### 4.3. REGISTRADOR. ESCRIBE TODOS LOS COMENTARIOS

##### 4.3.2.1. ERRORES ENCONTRADOS

##### 4.3.2.2. PREGUNTAS DE ESTILO

##### 4.3.2.3. OMISIONES

##### 4.3.2.4. CONTRADICCIONES

##### 4.3.2.5. SUGERENCIAS PARA IMPROVISACIÓN

#### 4.3.2.5.PROPUESTAS ALTERNATIVAS

#### 4.3.3.AUTOR.

##### 4.3.3.1.RESPONSABLE DE QUE SE EXAMINEN LOS ELEMENTOS DEL SOFTWARE

##### 4.3.3.2.PRESENTA LOS MATERIALES.

#### 4.3.4.MEMBROS DEL EQUIPO DE REVISIÓN

##### 4.3.4.1.RESPONSABLES POR LA REVISAR CUALQUIER MATERIAL DE ENTRADA PREVIO A LA REVISIÓN

##### 4.3.4.2. PARTICIPAR DURANTE LA REVISIÓN PARA ASEGURAR QUE SE ESTÁN CUMPLIENDO LOS OBJETIVOS.

#### 4.4.ENTRADAS:

##### 4.4.1.DECLARACIÓN DE OBJETIVOS

##### 4.4.2.ELEMENTO SOFTWARE

##### 4.4.3.ESTÁNDARES VIGENTES PARA EL DESARROLLO DEL ELEMENTO SOFTWARE.

##### 4.4.4.ESPECIFICACIONES DE LOS ELEMENTOS SOFTWARE.

#### 4.5.CRITERIO DE ENTRADA

##### 4.5.1.AUTORIZACIÓN. SE ESTABLECE EN EL DOCUMENTO DE PLANIFICACIÓN DEL PROYECTO

##### 4.5.2.COMIENZO DEL EVENTO

##### 4.5.2.1.EL AUTOR DE LOS ELEMENTOS SOFTWARE HA INDICADO QUE ESTA LISTO, Y ESTA PREPARADO PARA PRESENTAR LOS MATERIALES.

##### 4.5.2.2.EL MODERADOR ESTA EQUIPADO POR PRACTICAS O ESTÁNDARES LOCALES Y HA CONCLUIDO QUE EL ELEMENTO SOFTWARE ES SUFICIENTEMENTE COMPLETO PARA CONDUCIR LA REVISIÓN.

#### 4.6.PROCEDIMIENTOS

##### 4.6.1.PLANIFICACIÓN. EL MODERADOR:

##### 4.6.1.1. IDENTIFICA EL EQUIPO DE REVISIÓN

##### 4.6.1.2.PROGRAMA LA REUNIÓN Y SELECCIONA EL LUGAR DE ENCUENTRO.

##### 4.6.1.3.DISTRIBUYE TODO EL MATERIAL DE ENTRADA NECESARIO A LOS PARTICIPANTES, PERMITIENDO EL TIEMPO DE PREPARACIÓN ADE CUADO.

##### 4.6.2.APRECIACIÓN GLOBAL

##### 4.6.3.PREPARACIÓN. LOS PARTICIPANTES

##### 4.6.3.1.REVISAN EL MATERIAL DE ENTRADA

##### 4.6.3.2.PREPARAN UNA LISTA DE INQUIETUDES Y EVENTOS A DISCUTIR.

##### 4.6.4.EXAMEN

##### 4.6.4.1.EL PRESENTADOR HACE UNA PRESENTACIÓN RÁPIDA DE LOS ELEMENTOS DE SOFTWARE.

##### 4.6.4.2.EL AUTOR RECORRE LOS ELEMENTOS DE SOFTWARE PARA QUE LOS MIEMBROS DEL EQUIPO PUEDAN HACER PREGUNTAS O CREAR DISCUSIONES O DOCUMENTAR SU IMPORTANCIA.

##### 4.6.4.3.EL REGISTRADOR ESCRIBE LOS COMENTARIOS Y DECISIONES PARA INCLUIRLAS EN EL INFORME DE REVISIÓN.

4.6.4.4. EL EQUIPO DE REVISIÓN PUEDE RECOMENDAR UNA REVISIÓN DE REFUERZO

4.7. CRITERIO DE SALIDA

4.7.1. EL TOTAL DE ELEMENTOS DE SOFTWARE HAN SIDO REPASADOS EN DETALLE

4.7.2. TODAS LAS DEFICIENCIAS, OMISIONES, Y SUGERENCIAS PARA MEJORAMIENTO HAN SIDO ANOTADAS.

4.7.3. EL INFORME DE APRECIACIÓN GLOBAL HA SIDO PRODUCIDO.

4.8. SALIDA. INFORME DE REVISIÓN QUE CONTIENE:

4.8.1. IDENTIFICACIÓN DE EQUIPO DE REVISIÓN

4.8.2. IDENTIFICACIÓN DE LOS ELEMENTOS DE SOFTWARE EXAMINADOS.

4.8.3. LA DECLARACIÓN DE OBJETIVOS QUE FUERON SEGUIDOS DURANTE LA REUNIÓN DE REVISIÓN

4.8.4. UNA LISTA DE DEFICIENCIAS ANOTADAS, OMISIONES, CONTRADICCIONES Y SUGERENCIAS PARA MEJORAR

4.8.5. CUALQUIER RECOMENDACIÓN HECHA POR EL EQUIPO DE REVISIÓN EN COMO ACABAR CON LAS DEFICIENCIAS Y PROBLEMAS NO RESUELTOS.

4.8.6. SUGERENCIAS DE LAS REVISIONES DE REFUERZO.

4.9. AUDITORIA. EL MODERADOR DE LA REVISIÓN ARCHIVA ESTOS DOCUMENTOS

4.9.1. EL REPORTE DE REVISIÓN

4.9.2. MINUTOS DE REUNIÓN

4.9.3. MATERIALES EN LOS CUALES LAS CONCLUSIONES ESTÁN BASADAS

5. EL PROCESO DE LA AUDITORIA

5.1. OBJETIVO. PROVEER UNA CONFORMIDAD DE PRODUCTOS Y PROCESOS

5.2. RESUMEN.

5.2.1. ESTABLECER PROCEDIMIENTOS PARA CONDUCIR LA AUDITORIA Y REFORZAR LA ACCIÓN EN LOS DESCUBRIMIENTOS DE AUDITORIA.

5.2.2. EVALUAR LOS ELEMENTOS DE SOFTWARE Y LOS PROCESOS PARA PRODUCIR UN CRITERIO DE AUDITORIA OBJETIVO COMPARADO CON REQUERIMIENTOS, PLANES ESPECIFICACIONES O PROCEDIMIENTOS, GUÍAS Y ESTÁNDARES.

5.2.3. DOCUMENTAR LOS RESULTADOS

5.2.3.1. LISTA DE LOS ÍTEMS QUE NO CUMPLEN

5.2.3.2. OTROS PRODUCTOS PARA REVISIÓN Y ACCIÓN SUBSECUENTE.

5.2.3.3. RECOMENDACIONES

5.3. RESPONSABILIDADES ESPECIALES.

5.3.1. JEFE

5.3.1.1. ORGANIZAR Y DIRIGIR LA AUDITORIA

5.3.1.2. COORDINAR LA PREPARACIÓN Y EDICIÓN DEL INFORME DE AUDITORIA.

5.3.1.3. ASEGURAR QUE EL EQUIPO DE AUDITORIA ESTA PREPARADO PARA CONducIRLA

5.3.1.4. ASEGURAR QUE LOS PROCEDIMIENTOS DE AUDITORIA SON DESARROLLADOS Y PRODUCEN INFORMES EN CONCORDANCIA CON EL ALCANCE DE LA AUDITORIA.

#### 5.4. ENTRADA

5.4.1. EL PROPÓSITO Y ALCANCE DE LA AUDITORIA

5.4.2. CRITERIO DE AUDITORIA OBJETIVA, TALES COMO COMPENDIAR REQUERIMIENTOS, PLANES, ESPECIFICACIONES, PROCEDIMIENTOS, PAUTAS Y NORMAS

5.4.3. LOS ELEMENTOS Y PROCESOS DE SOFTWARE A SER AUDITADOS Y CUALQUIER HISTORIA PERTINENTE.

5.4.4. INFORMACIÓN DE ANTECEDENTES RESPECTO A RESPONSABLE DE LA ORGANIZACIÓN DE LOS PRODUCTOS Y PROCESOS AUDITADOS

#### 5.5. CRITERIO DE ENTRADA: UNO DE LOS SIGUIENTES EVENTOS

5.5.1. CUANDO UN PROYECTO PILAR ESPECIAL HA SIDO RECHAZADO, LA AUDITORIA ES INICIADA POR PLANES MAS TEMPRANOS

5.5.2. CUANDO PARTES EXTERNAS SOLICITAN UNA AUDITORIA EN UNA FECHA CALENDARIO ESPECIFICO O PROYECTO PILAR.

5.5.3. CUANDO UN ELEMENTO DE ORGANIZACIÓN LOCAL HA REQUERIDO LA AUDITORIA, ESTABLECIENDO UNA NECESIDAD CLARA Y ESPECIFICA.

5.5.4. CUANDO UN PROYECTO PILAR ESPECIAL, FECHA CALENDARIO, U OTRO CRITERIO QUE HA SIDO ENCONTRADO COMO PARTE DE LA CARTA CONSTITUCIONAL DE LA ORGANIZACIÓN INTERVENIDA, EN RESPUESTA SE INICIA LA AUDITORIA

#### 5.6. PROCEDIMIENTOS

##### 5.6.1. PLANIFICACIÓN:

5.6.1.1. PROCESOS DEL PROYECTO Y EL TIEMPO DE AJUSTE PARA OBSERVACIÓN DEL EQUIPO DE AUDITORIA.

5.6.1.2. SOFTWARE REQUERIDO Y SU DISPONIBILIDAD

5.6.1.3. LAS MÁSCARAS DE LOS REPORTES SON IDENTIFICADOS

5.6.1.4. DISTRIBUCIÓN DE REPORTES.

5.6.1.5. FLUJO DE ACTIVIDADES REQUERIDAS.

5.6.1.6. REQUERIMIENTOS

5.6.1.7. CRITERIO OBJETIVO DE AUDITORIA

5.6.1.8. LISTA DE CHEQUEO Y PROCEDIMIENTOS DE LA AUDITORIA.

5.6.1.9. PERSONAL DE LA AUDITORIA

5.6.1.9.1. NÚMERO REQUERIDO

5.6.1.9.2. HABILIDADES

5.6.1.9.3. EXPERIENCIA

5.6.1.9.4. RESPONSABILIDADES.

5.6.1.10. ORGANIZACIONES INVOLUCRADAS

5.6.1.11.FECHA, TIEMPO, LUGAR, AGENDA, Y PÚBLICO INTENCIONAL DE SESIÓN DE APRECIACIÓN GLOBAL

5.6.2 APRECIACIÓN GLOBAL GUIADA POR EL JEFE DE LA AUDITORIA, PROVEE LO SIGUIENTE:

5.6.2.1.DE LOS ACUERDOS EXISTENTES

5.6.2.2.DE PRODUCTOS Y PROCESOS QUE INTERVIENEN

5.6.2.3.DE PROCESOS DE AUDITORIA, SUS OBJETIVOS, Y SALIDAS

5.6.2.4.CONTRIBUCIONES ESPERADAS DE LA ORGANIZACIÓN QUE INTERVIENE EN EL PROCESO DE AUDITORIA

5.6.2.4.1.NÚMERO PERSONAS A SER ENTREVISTADAS

5.6.2.4.2.FACILIDADES PARA LA REUNIÓN

5.6.2.5.ESPECIFICAR HORARIO DE LA AUDITORIA.

5.6.3.PREPARACIÓN.

5.6.3.1. EL EQUIPO DE AUDITORIA:

5.6.3.1.1.ENTENDER LA ORGANIZACIÓN.

5.6.3.1.2.ENTENDER LOS PRODUCTOS Y PROCESOS

5.6.3.1.3.ENTENDER EL CRITERIO DE AUDITORIA OBJETIVA

5.6.3.1.4.PREPARAR EL REPORTE DE AUDITORIA

5.6.3.1.5. DETALLAR EL PLAN DE AUDITORIA

5.6.3.2 EL JEFE DEL EQUIPO DEBE

5.6.3.2.1.ORIENTAR E INSTRUIR EL EQUIPO COMO SE REQUIERA

5.6.3.2.2.FACILITAR LAS ENTREVISTAS DE LA AUDITORIA.

5.6.3.2.3.FACILITAR MATERIALES, DOCUMENTOS Y HERRAMIENTAS REQUERIDAS PARA LOS PROCEDIMIENTOS DE LA AUDITORIA

5.6.3.2.4.FACILITAR ELEMENTOS DEL SOFTWARE

5.6.3.2.5.PROGRAMAR LAS ENTREVISTAS.

5.6.4.EXAMEN

5.6.4.1.REVISAR PROCEDIMIENTOS E INSTRUCCIONES

5.6.4.2.EXAMINAR LA ESTRUCTURA DEL TRABAJO

5.6.4.3.EXAMINAR EVIDENCIA DE IMPLEMENTACIÓN Y CONTROLES BALANCEADOS

5.6.4.4.EL PERSONAL ENTREVISTADO DETERMINA EL ESTADO Y FUNCIONAMIENTO DE LOS PROCESOS Y DE LOS PRODUCTOS

5.6.4.5.EXAMINAR DOCUMENTOS DE LOS ELEMENTOS.

5.6.4.6.PROBAR LOS ELEMENTOS.

5.6.5 INFORMES. RESULTADO ANTERIOR DE LOS RESULTADOS FORMALES PARA

5.6.5.1.RESOLVER CUALQUIER INQUIETUD O AMBIGÜEDAD MIENTRAS MANTIENE OBJETIVIDAD Y CORRECCIONES. 8.6.5.2.AGREGAR CONSISTENCIA AL NIVEL DE REPORTE DETALLADO

5.6.5.3.AGREGAR CUALQUIER INFORMACIÓN RECIENTEMENTE VERIFICADA.

## 5.7. CRITERIOS DE SALIDA. CUANDO

5.7.1.CADA ELEMENTO (S) DENTRO DE LA MIRA DE LA AUDITORIA SE HA EXAMINADO

5.7.2.LOS HALLAZGOS HAN SIDO PRESENTADOS AL AUDITOR DE LA ORGANIZACIÓN

5.7.3.SE TIENE RESPUESTA DEL BOSQUEJO DE LOS HALLAZGOS RECIBIDOS Y EVALUADOS

5.7.4.LOS HALLAZGOS FINALES HAN SIDO PRESENTADOS FORMALMENTE AL AUDITOR DE LA ORGANIZACIÓN Y A LA ENTIDAD

5.7.5.LOS INFORMES DE AUDITORIA HAN SIDO PREPARADOS Y SOMETIDOS A DESTINATARIOS DESIGNADOS EN EL PLAN DE LA AUDITORIA

5.7.6.EL INFORME DE RECOMENDACIONES, SI ES REQUERIDO POR EL PLAN, HA SIDO PREPARADO Y SOMETIDO A LOS DESIGNADOS EN EL PLAN DE LA AUDITORIA

5.7.7.TODO EL FLUJO DE LAS ACCIONES DE LA ORGANIZACIÓN INCLUIDA EN LA META (O CONTRATO) DE LA AUDITORIA HA SIDO EJECUTADO

5.8 SALIDA. REPORTES DE AUDITORIA FINAL DEBEN, COMO MÍNIMO, CONTENER LO SIGUIENTE:

5.8.1.IDENTIFICACIÓN DE LA AUDITORIA.

5.8.1.1.TÍTULO DEL INFORME

5.8.1.2.AUDITOR DE LA ORGANIZACIÓN

5.8.1.3.ORGANIZACIÓN AUDITADA

5.8.1.4.FECHA DE LA AUDITORIA.

5.8.2.ALCANCE .

5.8.2.1.ENUMERACIÓN DE LOS ESTÁNDARES

5.8.2.2.ESPECIFICACIONES

5.8.2.3.PRÁCTICAS

5.8.2.4.PROCEDIMIENTOS QUE CONSTITUYEN EL CRITERIO OBJETIVO DE AUDITORIA, CONTRA LOS CUALES SE CONDUJO LA AUDITORIA DE LOS ELEMENTOS DEL SOFTWARE Y PROCESOS AUDITADOS.

5.8.3.CONCLUSIONES

5.8.4.SINOPSIS. LISTA DE TODOS LOS ELEMENTOS DEL SOFTWARE AUDITADOS, PROCESOS , Y HALLAZGOS

5.8.5.FLUJO. TIPO Y TIEMPO DE LAS ACTIVIDADES DE LA AUDITORIA.

5.8.6.RECOMENDACIONES. DEBEN SER REPORTADAS POR SEPARADO DE LOS RESULTADOS.

5.9 AUDITORIA. EL MATERIAL DE DOCUMENTACIÓN DEL PROCESO QUE INCLUYE LO SIGUIENTE:

5.9.1.TODOS LOS PROGRAMAS DE TRABAJO, LISTAS DE CHEQUEO, ETC

5.9.2.EQUIPO DE TRABAJO

5.9.3.NOTAS DE INTERVENTORIA, NOTAS DE OBSERVACIONES.

5.9.4.COMPLETAS PRUEBAS DE EVIDENCIA

5.9.5.COPIA DE ÍTEMS EXAMINADOS, ANOTADOS

5.9.6. INFORMES DEL PROYECTO CON RESPUESTA DEL AUDITOR DE LA ORGANIZACIÓN

5.9.7. MEMORANDOS DEL FLUJO COMO SEA NECESARIOS

### 3. DESCRIPCIÓN DEL PROCEDIMIENTO STANDARD

Las secciones que siguen dan un contorno detallado de los procedimientos para cada tipo de auditoria / revisión (Por ejemplo revisión técnica y gerencial, inspecciones, revisiones y auditorias).

#### DESCRIPCIÓN

##### ○ **Cuando**

Figura 1 abajo ilustra la manera en la cual se utilizan las diferentes revisiones y auditorias y cómo se diferencian unas de otras. Los procesos tienen dos distinciones principales como siguen:

- **Tema** – Las revisiones y las auditorias gerenciales evalúan el proyecto en su totalidad, mientras que las verificaciones, las revisiones y las auditorias técnicas se aplican a los documentos o a los módulos del software.
- **Medios de iniciación** – Las auditorias y las inspecciones son iniciadas normalmente por partes externas al equipo del proyecto, tales como el encargado de calidad o el comité de dirección. Son hechas por la gente seleccionada por el partido que inicia y que es independiente del equipo del proyecto, aunque el equipo del proyecto ayuda con la información. Verificaciones y revisiones técnicas y en un grado menor las revisiones gerenciales son actividades comunes del proyecto iniciadas por el equipo del proyecto durante el curso normal del proyecto. La verificación es una revisión opcional que se realiza antes de terminar el documento o módulo del software.

##### ○ **Registro del estado de la Revisión**

El registro del estado de la revisión debe ser creado para registrar el progreso de la revisión. el registro debe registrar por los menos los siguientes detalles para cada revisión realizada.

- Número del documento del informe de la revisión (de acuerdo con el Standard-03)
- Número del documento del ítem principal revisado (de acuerdo con el Standard-03)
- Fecha de inicio de la revisión.
- Fecha de finalización de la revisión.
- Número de decisiones.
- Número de los puntos de acción.
- Número de defectos y ediciones más importantes.
- Esfuerzos de la revisión.
- Tamaño del material revisado
- Decisión de salida (paso, paso provisional o falla).
- Fechas de los puntos de acción que han sido terminados.

- Fecha de todas las ediciones / defectos que fueron corregidos.
- Fecha de las revisiones concluidas
- Estado (planeado, activo o completo).
- **Práctica obligatoria y opcional**
  - Obligatoria** – en la conducción de proyectos QMS las actividades siguientes son obligatorias.
    - Revisión técnica de todos los documentos antes de la aprobación (referirse al Standard-01).
    - Revisión gerencial de proyectos en la terminación de cada fase.
  - Opcional (recomendado)** – en el conducto de proyectos QMS las siguientes actividades son opcionales pero recomendadas.
    - Verificación de documentos durante su desarrollo como un medio de mejorar la calidad del documento.
    - Revisión técnica de los módulos del software seleccionado para asegurar conformidad del estándar y las especificaciones
  - Opcional** – en la conducción de proyectos QMS las siguientes actividades son únicamente opcionales
    - Verificación del módulos del software para identificar las áreas para mejorar.
    - Auditorias de proyectos para establecer de conformidad con requerimientos QMS
    - Inspecciones de documentos y módulos del software para verificar conformidad con los estándares QMS.
    - Revisiones técnicas de todos los módulos del software.
- **Revisiones parciales o completas**

Las revisiones parciales revisan únicamente los elementos que han cambiado. Las revisiones parciales se pueden hacer únicamente después de una revisión completa.

Cuando se desarrolla un documento, un ítem del software u otro ítem del proyecto se experimenta una revisión (revisión técnica) para validar su contenido, y conforme a los estándares aplicables. (Para documentos la aprobación puede ocurrir sólo después que las acciones de revisión y seguimiento han sido resueltas)

Esta revisión debe ser una revisión completa del ítem. Esto es, debe revisar el contenido entero del ítem. Si el ítem se modifica más adelante de cierta manera, debe realizarse otra revisión. Esta revisión puede elegir cubrir solamente la porción modificada del material. Tales revisiones se refieren como revisiones parciales.

Lo siguiente aplica para revisiones parciales:

  - Pueden ser utilizadas si el material a ser revisado ha sido previamente sujeto a una revisión completa.
  - No deben ser utilizados si el material ha experimentado cambios importantes.

- La sección a ser revisada debe ser indicada en los objetivos de la revisión.
- La reunión de revisión debe convenir que una revisión parcial es apropiada y que los cambios no afectan el resto del documento.

## **PROCESO DE REVISIÓN GERENCIAL**

### ○ **Propósito**

Una revisión gerencial es una evaluación formal de un proyecto por un equipo de revisión determinado para determinar si el proyecto está procediendo de acuerdo con los planes del proyecto y de calidad.

Durante la reunión de revisión el equipo de revisión examina el plan, o el progreso aplicado al plan, estándares y/o pautas aplicables. Se registran los problemas de cada área. Cuando los datos e información críticos no pueden ser proporcionados, debe programarse una reunión adicional para completar el proceso de revisión gerencial.

El resultado de la revisión es publicado en un informe el cual es archivado de acuerdo con el Standard - 03.

Mientras las revisiones gerenciales se pueden hacer en cualquier momento, estas se hacen normalmente al final de cada fase del proyecto

Los objetivos de los procesos de revisión gerencial son los siguientes.

- Asegurar el movimiento de las siguientes actividades según el plan, basados en una evaluación del estado de desarrollo del proyecto.
- Cambiar la dirección del proyecto o identificar las necesidades del planeamiento alternativo.
- Mantener el control total del proyecto con la asignación de recursos adecuada.

### ○ **Iniciación**

#### ○ **Inicializando eventos**

La revisión gerencial son actividades programadas, su sincronización está determinada por los planes del proyecto y de calidad. Las revisiones no programadas de la gerencia se pueden llamar por el gerente del proyecto, el comité de dirección del proyecto o el encargado de calidad.

#### ○ **Personal disponible**

Las tareas de revisión para programar las revisiones de gerencia están normalmente en cabeza del gerente del proyecto, salvo se especifique lo contrario en el proyecto o plan de calidad. Las revisiones gerenciales no programadas deben ser llamadas por el director del proyecto, comité de dirección del proyecto o el encargado de calidad. La tarea de revisión para una revisión gerencial no programada es determinada por el partido iniciando la revisión en consulta con el director del proyecto. Una revisión gerencial debe consistir al menos

de tres personas. Los papeles y las responsabilidades de estas personas son los siguientes

Papel	Responsabilidad
Líder de revisión	<ul style="list-style-type: none"> <li>• Publica la notificación de la revisión.</li> <li>• Produce y distribuye el material de revisión adelantado.</li> <li>• Preside la reunión de revisión para asegurar que sus objetivos se cumplan.</li> <li>• Publica el informe de revisión.</li> <li>• Se prepara para la revisión estudiando el material distribuido.</li> <li>• Participa en la revisión para asegurar que sus objetivos se logren.</li> </ul>
Registrador	<ul style="list-style-type: none"> <li>• Registra los puntos y decisiones de acción durante la revisión.</li> <li>• Refleja cualquier cambio a los objetivos y/o a la agenda de la revisión en el informe de la revisión.</li> <li>• Prepararse para la revisión estudiando el material distribuido.</li> <li>• Participar en la revisión para asegurar que sus objetivos se logren.</li> </ul>
Revisores	<ul style="list-style-type: none"> <li>• Prepararse para la revisión estudiando el material distribuido.</li> <li>• Participar en la revisión para asegurar que sus objetivos se logren</li> </ul>

- **Notificación**

Dar a los participantes de la revisión gerencial con por lo menos cinco días laborales de anticipación noticias sobre la revisión. La notificación de la revisión debe incluir la siguiente información

  - Nombre y número del proyecto
  - Fecha, hora y lugar de la reunión de revisión
  - Tipo de revisión (Por ejemplo gerencial, técnica, inspección o verificación)
  - Lista de material de entrada.
  - Declaración de los objetivos de la revisión.
  - Lista de ediciones asegurando que los objetivos cubran todas las ediciones a ser discutidas.
  - Equipo de revisión – quién participa y en que papel (Por ejemplo moderador / líder, autor, registrador, etc).
  - Agenda
  - Fecha, hora, lugar y duración prevista del informa preliminar (si se requiere). Donde una revisión gerencial determine que las reuniones siguientes son requeridas para completar la revisión el día y la hora se deciden en la reunión (no se requiere notificación).
- **Material de Entrada**

El material de entrada debe estar disponible para los participantes por el líder de la revisión al menos con dos días laborales de anticipación a la revisión

Las entradas mínimas para la revisión gerencial son las siguientes:

  - Declaración de objetivos (en la notificación de la revisión)
  - Plan actual del proyecto, cronograma y datos de costos
  - Informes (por ejemplo informes de la revisión gerencial, informes de la revisión técnica, informes de auditoría) de otras revisiones / auditorías que ya han sido hechos (Los documentos del proyecto se archivan de acuerdo con el Std-03 – Limadura del Documento del Proyecto)
  - Informes de recursos asignados al proyecto
  - Datos sobre los documentos y los módulos del software terminado o en progreso
  - Reports of resources assigned to the project.
  - Información adicional según los requisitos del líder de la revisión
- **Objetivos**

Los objetivos de una revisión gerencial deben variar considerablemente. Un ejemplo de objetivos que pueden ser usados por un líder de revisión son los siguientes.

  - Comprobar el estado del proyecto para saber si hay conformidad con el estado previsto en el plan del proyecto (por ejemplo logros alcanzados sobre el cronograma, y que los documentos y los módulos del software existan y han sido revisados y aprobados).

- Identificar cualquier salida del plan del proyecto, anotando cualquier riesgo que pueda aplicarse
- Determinar si el proyecto esta siendo afectado por los factores internos y/o externos no identificados en el plan del proyecto
- Compilar una lista de las ediciones y las recomendaciones para ser tratadas por el comité director del proyecto.
- Compilar una lista de ediciones de ediciones y recomendaciones que pueden ser direccionadas por otras personas involucradas en el proyecto.
- Identificar un curso de acción continuado – expresado como punto de acción que resuelve para hacer ciertas recomendaciones

#### 4. CONCLUSIONES

- Conocer la importancia y ayuda que ofrecen las normas internacionales existentes para el desarrollo de proyectos de ingeniería del software
- Lograr un modelo teórico orientar a las organizaciones en la ejecución de proyectos de ingeniería del software
- Luego del estudio realizado a la norma IEEE-1028 de 1988, se concluye que las normas existentes en los diferentes campos de aplicación de la ingeniería en organizaciones especializadas, y elaborada por comités de personas especializadas en cada tema, es desaprovechada en muchos casos por falta de conocimiento. Es importante que estos análisis realizados por las universidades se puedan difundir de manera que sean aplicadas en el campo de la ingeniería del software.
- Para el caso particular de la norma IEEE 1028-1988 “Revisión y Auditoria del Software”, proporciona definiciones y requerimientos que permiten al personal de los proyectos realizar las revisiones y auditorias necesarias de productos hardware y software y de los procesos. La contribución es calidad. En el desarrollo de la documentación del proyecto es muy importante la verificación y validación para que su contenido sea verdadero y exacto.
- Es importante que las organizaciones tengan en cuenta que aunque un proceso de revisión y auditoria requiere un gran esfuerzo, los beneficios obtenidos son superiores.

## **5.RECOMENDACIONES**

- Se recomienda que las auditorias e inspecciones sean realizadas por partes externas al equipo del proyecto
- Las verificaciones y revisiones técnicas y gerenciales son actividades comunes del proyecto iniciadas por el equipo del proyecto durante el curso normal del proyecto.

## **BIBLIOGRAFÍA**

Bollinger, T. Y McGowen, C. A Critical Look At Software Capability Evaluations. IEEE Software, Julio 1991.

IEEE. Software engineering standards collection. Spring 1991 edition. ISBN 1-55937-080-7

McConnell Steve. Desarrollo Y Gestión de Proyectos Informáticos. 1997 ISBN 1-55615-900-5.

Paulk, M. y otros, Capability Maturity Model for Software, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

Roger S. Pressman. Ingeniería del Software, Un Enfoque Práctico.

## **Enlaces de interés**

[http://standards.ieee.org/IEEE Standards Association](http://standards.ieee.org/IEEE_Standards_Association)

[http://www.sei.cmu.edu/Software\\_Engineering\\_Institute\\_\(SEI\)](http://www.sei.cmu.edu/Software_Engineering_Institute_(SEI))

[http://pegasus.cc.ucf.edu/~ya999579/eel6885/product\\_evaluation.doc](http://pegasus.cc.ucf.edu/~ya999579/eel6885/product_evaluation.doc)

## **ANEXOS**

### **ANEXO A: Traducción de la Norma IEEE 1028 - 1988**

Norma 1028-1.988

Norma IEEE para  
Revisión y Auditoría de Software

Edición corregida Junio 30 de 1.989

Se han hecho correcciones sobre las páginas 5, 8, 10, 11, 12, 25, 26, 27, 29,  
y 35. Aparecen con negrilla los cambios hechos sobre ésta impresión.

Expositor

Comité Técnico de Ingeniería del Software de la Sociedad de Computadores  
de la IEEE

Aprobado Junio 9, 1.988

Norma IEEE

Derechos reservados de autor en 1.989 por  
El Instituto de Ingenieros Eléctricos y Electrónicos  
345 East 47 Street, New York NY 10017 E.E.U.U.

No se reproduce ninguna parte de esta publicación en cualquier forma, en un  
sistema electrónico u otro, sin permiso escrito del publicador.

Los documentos de las normas de la IEEE son desarrollados por el Comité Técnico de Sociedades de la IEEE y el Comité Coordinador de Normas de la IEEE. Miembros de los comités sirven voluntariamente y sin compensación. No son necesariamente miembros del Instituto. Las normas desarrolladas por la IEEE representan un consenso de la gran experiencia en el tema dentro y fuera del Instituto de la IEEE que han expresado un interés en participar en el desarrollo de la norma.

El uso de las normas IEEE es totalmente voluntario. La existencia de una norma IEEE no implica que no hay otras maneras de producir, probar, medir, comprar, mercadear, o proveer otros géneros y servicios relacionados con la norma IEEE. Además, los puntos de vista expresados en una norma son aprobados y están sujetos a cambios de acuerdo a su desarrollo sobre el estado del arte y comentarios recibidos de usuarios de la norma. Todas las normas de IEEE están sujetas a repasos por lo menos cada cinco años para revisión o reafirmación. Cuando un documento de más de cinco años no ha sido reafirmado, es razonable concluir que no refleja totalmente el estado presente del arte. Se advierte a los usuarios para que determinen que tengan la última edición de cualquier norma IEEE.

Comentarios para la revisión de la norma IEEE son aceptados de cualquier interesado, indiferentemente de su afiliación como miembro de IEEE. Sugerencias para cambios en los documentos deben estar en el formato de un cambio propuesto del texto, apoyado de comentarios apropiados.

Interpretaciones: De vez en cuando se generan preguntas del significado de partes de las normas para una aplicación específica. Cuando se requiere la atención de IEEE por necesidad de interpretación, el Instituto prepara respuestas apropiadas. Una vez la norma IEEE llega a un consenso de los interesados, es importante para asegurar que cualquier interpretación recibe la concurrencia de un balance de los interesados. Por esta razón IEEE y los miembros de su comité técnico no están habilitados para proveer una respuesta contestando al instante el requerimiento de la interpretación, excepto en esos casos donde el autor ha recibido previamente esta consideración formal.

Comentarios sobre las normas y requerimientos para interpretaciones deben dirigirse a:

Secretaria, Normas IEEE  
345 East 47<sup>th</sup> Street  
New York, NY 10017  
E.E.U.U.

Los documentos de las normas IEEE son adoptados por el Instituto de Ingenieros Eléctricos y Electrónicos sin tener en cuenta si su adopción requiere patentes sobre artículos, materiales o procesos. Tal adopción no asume ninguna obligación a cualquier dueño de patente, ni asume cualquier

obligación a cualquiera de las partes adoptadas a los documentos de las normas.

## Prólogo

(Este prólogo no es una parte de la norma IEEE-1028-1988. Norma IEEE para Revisión y Auditoria de Software.)

Este Prólogo provee al usuario la razón y fondo de la revisión y procedimientos de la Auditoria perfilados en esta norma, y su relación con otras normas IEEE.

**Propósito.** Esta norma define los procesos de revisión y Auditoria aplicable a software crítico y no crítico, y los procedimientos específicos requeridos para la ejecución de las revisiones y Auditorias descritas. Esta norma soporta ANSI/IEEE Norma 730-1.984 para definir revisión y Auditoria de procedimientos.

Esta norma no establece medidas o mecanismos para juzgar individualmente las contribuciones para el producto en desarrollo. Conforme a esta norma debe ser aclarada cuando las revisiones y Auditorias son ejecutadas usando procedimientos definidos por esta norma.

Los documentos adicionales que deben ser referenciados para una aplicación específica de la norma son los siguientes:

- (1) Norma ANSI/IEEE 729-1983, Norma IEEE Glosario de Terminología de la Ingeniería del Software.
- (2) Norma ANSI/IEEE 730.1984, Norma IEEE para Plan de Aseguramiento de Calidad del Software.
- (3) Norma ANSI/IEEE 828-1983, Norma IEEE para Plan de Administración de la Configuración del Software.
- (4) Norma ANSI/IEEE 983-1986, IEEE Guía para Aseguramiento de la Calidad del Software.
- (5) Norma ANSI/IEEE 1012-1986 Norma IEEE para Plan de Verificación y Validación del Software.

Estas normas están dirigidas a aplicaciones específicas. Esta norma sólo referencia normas de la IEEE. Esta norma puede ser aplicada con normas locales que requieren revisión y Auditoria.

Sin embargo, para soportar el uso de esta norma por aseguramiento de clasificación uniforme de códigos, use el estándar IEEE 1044-1.988, Norma IEEE Clasificación para Errores, Fallas y Fracayos del Software.

**Intento General de Uso.** Esta norma aplica a todas las fases del ciclo de la vida del software típico y provee una norma que planea la revisión y Auditoría que puede ser preparada y asegurada. El máximo beneficio puede ser derivado de esta norma por planificación del ciclo de la vida del proyecto.

Se puede usar esta norma con diferentes ciclos de vida. La norma ANSI/IEEE 730-1984 identifica especificaciones (y mínimo) de revisión y Auditoría aplicado para software crítico, mientras que esta norma define los procesos específicos disponibles para resolver las necesidades de la aplicación. La planificación para revisiones y Auditorías son parte de una verificación y validación aprobada que pueden ser determinadas por la norma ANSI/IEEE 1012-1.986.

Esta norma para revisión y Auditoría del software se escribió considerando el software y el ambiente del sistema operacional. Se puede usar cuando el software es la entidad del sistema total o una parte del un gran sistema. Puede tomarse para integrar actividades de revisión y Auditoría del software en la planificación ó el desarrollo del sistema total, debe realizarse en conjunto con la revisión y Auditoría del sistema hardware para los beneficios del sistema entero. Para tener la posibilidad de aplicar esta norma, simplemente reemplace el elemento software por el elemento sistema cuando este leyendo.

Se escribió esta norma asumiendo que las organizaciones no son corruptas. Esto puede usarse, por ejemplo, durante el curso de una Auditoría cuando se descubre robo ó malversación. En tal caso aunque la Auditoría puede proceder, la aplicabilidad de esta norma desaparece.

**Apéndices.** Aunque no se considera parte de norma, se proveen los apéndices para asistir al lector en la comprensión:

- (1) Aplicación de procesos específicos requeridos por la norma ANSI/IEEE 730-1.984
- (2) Cómo los procesos definidos en esta norma pueden ser aplicados para encontrar las aplicaciones necesarias.

**Audiencia.** Esta norma ha sido escrita por quienes son responsables para definir, planear, implementar o soportar revisiones y Auditorías de software.

**Conformidad.** La conformidad para esta norma pueden ser solicitadas cuando los procesos de revisión y Auditoría sean ejecutados como se define en ésta norma.

## Contenido

### SECCIÓN

#### 1. Alcance y Referencias

##### 1.1 Alcance

##### 1.2 Referencias

#### 2. Definiciones

#### 3. Introducción

##### 3.1 Requisitos previos al Proceso de Revisión

##### 3.2 Requisitos previos al Proceso de Auditoria

##### 3.3 Planilla de Descripción Procedimental

#### 4. El Proceso de Revisión Gerencial

##### 4.1 Objetivos

##### 4.2 Resumen

##### 4.3 Responsabilidades Especiales

##### 4.4 Entrada

##### 4.5 Criterios de Entrada

##### 4.6 Procedimientos

##### 4.7 Criterios de Salida

##### 4.8 Salida

##### 4.9 Auditoria

#### 5. El Proceso de Revisión Técnica

##### 5.1 Objetivos

##### 5.2 Resumen

##### 5.3 Responsabilidades Especiales

##### 5.4 Entrada

##### 5.5 Criterios de Entrada

##### 5.6 Procedimientos

##### 5.7 Criterios de Salida

##### 5.8 Salida

5.9 Auditoria

6. El Proceso de Inspección del Software

6.1 Objetivos

6.2 Resumen

6.3 Responsabilidades Especiales

6.4 Entrada

6.5 Criterios de Entrada

6.6 Procedimientos

6.7 Criterios de Salida

6.8 Salida

6.9 Auditoria

7. El Proceso de Revisión

7.1 Objetivos

7.2 Resumen

7.3 Responsabilidades Especiales

7.4 Entrada

7.5 Criterios de Entrada

7.6 Procedimientos

7.7 Criterios de Salida

7.8 Salida

7.9 Auditoria

8. El Proceso de Auditoria

8.1 Objetivos

8.2 Resumen

8.3 Responsabilidades Especiales

8.4 Entrada

8.5 Criterios de Entrada

8.6 Procedimientos

8.7 Criterios de Salida

8.8 Salida

8.9 Auditoria

## FIGURAS

Figura 1. Relación de Algunos Procesos de Aseguramiento de Calidad para Productos y Proyectos

## TABLAS

Tabla 1. Algunos Procesos Principales para Alcanzar Objetivos de la Calidad

## APÉNDICES

Apéndice A. Una Guía para Aplicación de Procesos de Software Crítico

Apéndice B. Una Guía para Aplicaciones de Revisión Específicas

Apéndice C. Una Guía para Aplicaciones de Auditorías Específicas

## APÉNDICE DE TABLAS

Tabla A1. Examen de Software Crítico

Tabla B1. Distinción de los Procesos de Revisión

## Norma IEEE para Revisión y Auditoría de Software

### 1. Alcance y Referencias

**1.1 Alcance.** El propósito de esta norma es proveer definiciones y requerimientos uniformes para los procesos de revisión y Auditoría. No establece la necesidad de conducir revisiones ó Auditorías específicas; esa necesidad es definida por política local. Cuando las revisiones y Auditorías específicas son requeridas, los procedimientos de las normas para su ejecución deben estar definidos.

Esta norma provee dichas definiciones para los procesos de revisión y Auditoría que son aplicables a productos y procesos durante el ciclo de vida del software. Cada organización debe especificar cuándo y dónde aplica esta norma y cualquier intento de desviación intencional de la misma.

**1.2 Referencias.** Se usará esta norma en conjunción con las publicaciones siguientes:

[1] Norma ANSI/IEEE 729-1983, Norma IEEE Glosario de Terminología de la Ingeniería del Software

[2] Norma ANSI/IEEE 829-1983, Norma IEEE para Documentación de Prueba del Software

## 2. Definiciones

**Auditoria.** Una evaluación independiente de productos o procesos del software para determinar cumplimiento, seguimiento, especificaciones y procedimientos de la norma basados en criterios objetivos que incluyan documentos que especifiquen:

- (1) La forma o contenido de los productos
- (2) El proceso por que se producirá.
- (3) Cómo cumplir las normas o seguimiento que se debe establecer

**Revisión.** Es una evaluación de elemento(s) del software o estado del proyecto para determinar discrepancias de los resultados planeados y las mejoras recomendadas. Esta evaluación sigue un proceso formal (por ejemplo, proceso de revisión gerencial, proceso de revisión técnica, proceso de inspección del software, o proceso de revisión)

**Elemento del software.** Un documento en proceso producido o adquirido durante el desarrollo o mantenimiento del software. Ejemplos específicos incluidos, pero no se limita a

- (1) Documentación de la Planificación del Proyecto (por ejemplo, plan de desarrollo del software, y verificación del software y plan de validación)
- (2) Requerimientos del Software y especificaciones de diseño.
- (3) Documentación de prueba de esfuerzo (por ejemplo, como los descritos en ANSI/IEEE 829-1983)
- (4) Documentación entregada al cliente
- (5) Código fuente del programa
- (6) Representación de soluciones del software implementados en firmware
- (7) Informes (por ejemplo, revisiones, Auditoria, estado del proyecto) y datos (por ejemplo, detección de errores, pruebas)

Para otras definiciones, incluyendo verificación y validación, ver Norma ANSI/IEEE 7291983 [1].

### 3. Introducción

Revisiones y Auditorias pueden ser usadas como soporte de objetivos asociados con el aseguramiento de la calidad del software, administración del proyecto, administración de la configuración, o función de control similar, como se muestra en la Tabla 1. Esos objetivos de la calidad, válido a través del desarrollo y mantenimiento del software, incluyen la necesidad de evaluación, verificación, validación y conformidad de la complacencia. Por causa de la competencia, las pruebas están definidas en la Tabla 1 como el proceso principal de validación. Las pruebas sin embargo, están fuera del alcance de esta norma. Los procesos definidos en esta norma incluye

- (1) Revisión Gerencial
- (2) Revisión Técnica
- (3) Inspección del Software
- (4) Revisión.
- (5) Auditoria.

Para soportar la conformidad de la confirmación, las actividades auditadas, deben incluir ejecución de revisiones y pruebas para determinar la consistencia del producto con respecto a las bases establecidas (Por ejemplo, Especificaciones de requerimiento del software o descripción del diseño) o dar testimonio de las correcciones en la implementación.

**Tabla 1**  
**Algunos Procesos Principales para Alcanzar Objetivos de Calidad**

<b>OBJETIVOS</b>	<b>PROCESOS PRINCIPALES QUE INCLUYE</b>
Evaluación	Verificación
Verificación	Inspección, Revisión
Validación	Pruebas
Conformidad	Auditoria

Además, el aseguramiento de la calidad de los productos y procesos, (Figura 1) provee una visión de cómo varios procesos de aseguramiento de calidad pueden ser trazados para examinar productos y proyectos

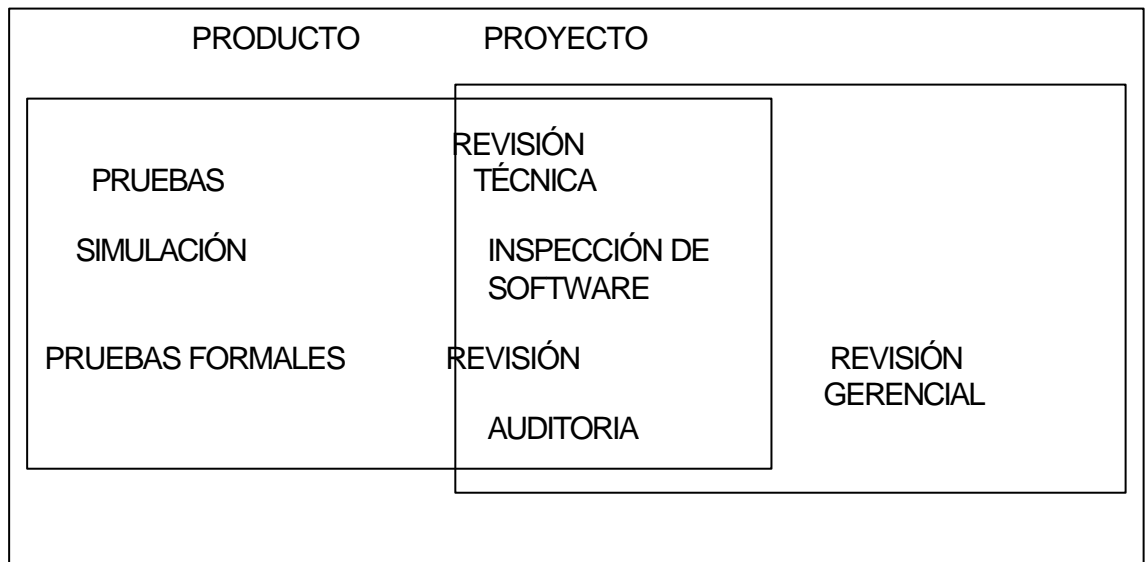
El examen de ediciones del proyecto (ambos técnico y administrativo) ocurre en varias fases durante el ciclo de vida del proyecto. Los resultados de dichos exámenes permiten mejorar el método de aseguramiento de la calidad del software y la habilidad para encontrar disminución de tiempo y costos. La evaluación de elementos del software ocurre durante la producción de los

elementos y su realización. Esto asegura que los elementos encarnen completa y correctamente sus especificaciones básicas.

Cualquier proceso normal tiene condiciones de requisitos previos; esas condiciones, no son suficientes para la realización del proceso. Para revisiones y Auditorias, se identifican estas condiciones en 3.1 y 3.2 respectivamente.

**Figura 1**

Relación de Unos Procesos de Aseguramiento de Calidad para Productos y Proyectos



La sección 3.3. presenta la plantilla usada para definir los procesos de revisión y Auditoria de software en las siguientes secciones.

**3.1 Requisitos previos al Proceso de Revisión.** El objetivo de revisar los elementos del software es evaluar el software o el estado del proyecto para identificar diferencias con los resultados planeados y hacer las recomendaciones donde sea apropiado.

Los siguientes cargos administrativos están para alcanzar estos objetivos:

**3.1.1. Administrador**

- (1) Responsable de las revisiones, y la reanudación anticipada del producto.
- (2) Responsable de ejecutar las revisiones y reportar el resultado de la revisión una vez ocurra el proyecto.

(3) Provee los recursos necesarios de tiempo, personal, presupuesto, y facilidades requeridas para planear, definir, ejecutar y manejar las revisiones.

### **3.1.2. Grupo de Desarrolladores**

(1) El grupo tiene un nivel de experiencia de desarrollo y suficiente conocimiento del producto para comprender el software durante la revisión.

(2) Instruyen y orientan en el uso de los procesos de revisión.

### **3.1.3 Desarrolladores del proceso de planificación**

(1) Definen para la fase de desarrollo los criterios de entrada y salida estándar.

(2) Los criterios del producto (tales como claridad y modularidad) están suficientemente definidos por las normas de desarrollo.

(3) El producto se divide en unidades manejables para la revisión.

(4) Los elementos del software son revisados y los procesos de revisión a ser empleados están identificados en la documentación de planificación del proyecto (por ejemplo, plan de aseguramiento de calidad del software, plan de revisión del software y plan de administración de proyecto software).

### **3.1.4. Revisores del proceso de planificación**

(4) Cada tipo de revisión que se emplea se especifica totalmente.

(2) Los procedimientos administrativos son definidos para habilitar la revisión que se va a iniciar, ejecutar, grabar y efectuar.

(3) Las responsabilidades son asignadas para la planificación, especificación, administración, y mantenimiento de los procesos de revisión.

**3.2 Requisitos previos al Proceso de Auditoria.** El objetivo de la Auditoria de software es proveer una evaluación objetiva de los productos y procesos que confirmen el cumplimiento de la norma, guías, especificaciones y procedimientos. Los siguientes requerimientos son requisitos previos para alcanzar el objetivo:

(1) Que exista un criterio de Auditoria objetivo (por ejemplo, contratos, requisitos, planes, especificaciones, normas) una vez los elementos y procesos del software puedan ser evaluados.

(2) Se selecciona el personal de la Auditoria para promover la objetividad del equipo. Usualmente hay independencia de cualquier responsabilidad directa de los productos y procesos examinados y deben ser de una organización externa.

(3) Al personal de Auditoria se le da autoridad suficiente para la administración apropiada en la realización de la Auditoria.

**3.3 Planilla de Descripción Procedimental.** Las siguientes descripciones establecen la plantilla mínima usada para planear, preparar, y ejecutar cualquier revisión o Auditoria. Los procesos de revisión y Auditoria de esta norma siguen esta plantilla común:

- (.1) Objetivo: Metas de los procesos.
- (.2) Resumen: Apreciación global del proceso.
- (.3) Responsabilidades Especiales: Papeles únicos para éste proceso específico.
- (.4) Entrada: Productos a que se aplica el proceso, e información soportada.
- (.5) Criterios de la Entrada: Condiciones que se debe satisfacer antes de que el proceso pueda empezar.
- (.6) Procedimientos: Pasos de la norma seguidos para ejecutar el proceso.
- (.7) Criterios de la Salida: Condiciones que deben ser satisfechas antes de que éste proceso sea considerado completo.
- (.8) Salida: Conjunto mínimo fijo de resultados entregados para la realización del proceso.
- (.9) Auditoria: Descripción de las evidencias necesarias para determinar la siguiente fecha para que el proceso continúe.

Pueden aumentarse secciones de la planilla adicionales a las nueve básicas cuando se requiera.

Las descripciones de los procedimientos en la sección “(.6)” deben ser presentadas como una inclusión, pero no se deben limitar a los siguientes pasos:

- (.6.1) Planificación
- (.6.2) Apreciación global
- (.6.3) Preparación
- (.6.4) Revisión

Subsecciones de los procedimientos pueden ser adicionadas cuando sea necesario.

## 4. El Proceso de Revisión Gerencial

**4.1 Objetivos.** El objetivo de una revisión gerencial es proveer recomendaciones para lo siguiente:

- (1) Hacer actividades progresivas de acuerdo al plan, basado en una evaluación del estado del desarrollo del producto;
- (5) Cambiar la dirección del proyecto o identificar la necesidad de planificación alternativa
- (6) Mantener control global del proyecto por asignación adecuada de recursos

El proceso de revisión gerencial se puede aplicar a desarrollos nuevos o actividades de mantenimiento.

**4.2 Resumen.** Una revisión gerencial es una evaluación formal del estado del plan del proyecto o del proyecto relativo al plan por un equipo de revisión designado.

Durante la reunión de la revisión el equipo de revisión entero examina los planes o el progreso aplicado a los planes, normas, y guías, o ambos. Cada área del problema identificado por el equipo de revisión es registrada. Cuando los datos e información crítico no pueden ser suministrados, entonces una reunión adicional es realizada para complementar el proceso de revisión gerencial.

**4.3 Responsabilidades Especiales** El jefe de la revisión es responsable de las tareas administrativas pertenecientes a la revisión, para asegurar que la revisión se conduce de una manera ordenada, y para asegurar el reporte de la revisión gerencial (ver 4.8).

Las responsabilidades individuales para reportes formales del estado del proyecto bajo revisión, debe asegurar que el estado del proyecto y toda la documentación de apoyo estén disponibles para su distribución.

Cada miembro del equipo de revisión es responsable por lo siguiente:

- (1) Estar preparado adecuadamente para las reuniones;
- (2) Asegurar que la revisión sea objetiva.

**4.4 Entrada.** Las mínimas entradas para el proceso de revisión gerencial son las siguientes:

- (1) Una declaración de objetivos para la revisión gerencial

- (2) Una lista de productos a ser direccionados
  - (3) Datos de horario y costo del proyecto presente.
  - (4) Reportes (por ejemplo, informes de revisiones de los directivos, reportes de revisiones técnicas y reportes de Auditoría) de otras revisiones o Auditorías, o ambos, ya completos
  - (5) Informes de recursos asignados al proyecto
  - (6) Datos sobre los elementos del software completos o en progreso
- Material de la referencia adicional aplicable puede ser suministrado por el director del proyecto o requerido por el jefe de la revisión.

## **4.5 Criterios de la Entrada**

**4.5.1 Autorización.** La necesidad de conducir revisiones gerenciales seguras se establece inicialmente en el documento apropiado de la planificación del proyecto (por ejemplo, Plan de Aseguramiento de Calidad del Software, Plan de Desarrollo del Software, o Plan de Verificación y Aprobación del Software). Bajo estos planes, la realización de un elemento del software específico (por ejemplo, un Documento de Planificación, una Especificación de los Requerimientos, Documento de Diseño, y una Especificación del Código), o la realización de una fase puede activar una revisión gerencial.

Además de estas revisiones gerenciales requeridas por un plan específico, otras revisiones gerenciales pueden ser formalmente anunciadas y tener los requerimientos de administración de la calidad del software, administración funcional, administración del proyecto o el cliente, de acuerdo a procedimientos locales.

**4.5.2 Comienzo del Evento.** Una revisión gerencial se conduce cuando un jefe de revisión hace lo siguiente:

- (1) Establecer o confirmar una declaración de objetivos para la reunión
- (2) Juzgar que los elementos del software o cualquier otra documentación o informes pertinentes para la revisión son suficientemente completos y se encuentra disponible como apoyo de los objetivos de la revisión.

## **4.6 Procedimientos.**

**4.6.1 Planificación.** Durante la planificación, el jefe de la revisión, en conjunto con el director del proyecto, hace lo siguiente:

- (1) Identifica el equipo de la revisión
- (2) Horario y lugar de reunión
- (3) Distribuye el material de entrada a los participantes, deja un tiempo adecuado para la preparación

**4.6.2 Apreciación global.** Una persona calificada del proyecto que se está examinando conduce una sesión de apreciación global para la revisión cuando lo requiere el jefe de la revisión. Esta apreciación global puede ocurrir como una parte de la reunión de examen (ver 4.6.4) o como una reunión separada.

**4.6.3 Preparación.** Cada persona del equipo de revisión estudia individualmente el material y lo prepara para la reunión de la revisión. Cualquier presentación requerida para el equipo de revisión se prepara.

**4.6.4 Examen.** Durante la revisión gerencial, el equipo de revisión solicita hacer una o más reuniones para lo siguiente:

(1) Examinar el estado del proyecto y determinar si este cumple con el estado esperado de acuerdo al plan predefinido.

(2) Examinar el estado del proyecto y determinar si está cubriendo factores externos e internos no considerados originalmente en el plan del proyecto;

(3) Registrar todas las desviaciones del estado esperado, acentuando riesgos;

(4) Generar una lista de ediciones y recomendaciones para ser direccionadas por un administrativo de nivel más alto.

(5) Generar una lista de ediciones y recomendaciones para ser direccionadas por otras responsabilidades individuales, u organizaciones que afectan el proyecto;

(6) Recomendar qué curso de acción se debe tomar sobre estos puntos;

(7) Recomendar autorización para revisiones o Auditorias adicionales;

(8) Identificar otras ediciones que puedan ser direccionadas.

**4.6.5 Reformas.** Cualquier plan de ajuste o reforma del producto como resultado de la revisión gerencial no es considerada como una parte del proceso de revisión, excepto cuando requiera como entrada adicional completar el proceso del examen.

**4.7 Criterios de Salida.** La revisión gerencial se considera completo cuando

(1) Todas las ediciones identificadas en la revisión de Declaración de Objetivos han sido direccionadas.

(2) El reporte de la revisión gerencial se ha emitido (ver 4.8).

**4.8 Salida.** La salida del proceso de revisión gerencial es un Reporte de la Revisión Gerencial que identifica lo siguiente:

- (1) El proyecto que está siendo revisado;
- (2) El equipo de revisión;
- (3) Entradas de la revisión;
- (4) Objetivos de la revisión;
- (5) Propiedad y estado del ítem en acción;
- (6) Una lista de ediciones y recomendaciones identificadas por el equipo de revisión que deben ser direccionadas para el proyecto satisfacer su requerimiento;
- (7) Recomendaciones con respecto a cualquier revisión y Auditoría futura, y una lista de información adicional y fecha en que se obtiene antes de que pueda ser ejecutada.

Aunque esta norma coloca requerimientos mínimos para el contenido del informe, se dan estas normas locales para prescribir cualquier informe de requerimiento de formato.

**4.9 Auditoría.** El reporte de revisión gerencial es un ítem de Auditoría. Se recomienda que este informe sea detectable para y desde la planificación de documentación del proyecto apropiado (por ejemplo, Plan de Desarrollo del Software, Plan de Aseguramiento de Calidad del Software, u otros planes aplicables, o una combinación de estos).

## 5. El Proceso de Revisión Técnica

**5.1. Objetivo.** El objetivo de la revisión técnica es evaluar un determinado elemento del software específico (s) y proveer administración con pruebas que:

- (1) El elemento del software (s) coincida con sus características técnicas;
- (2) El desarrollo (o mantenimiento) del elemento del software se hace acorde a diseños, normas, y pautas aplicables al proyecto.
- (3) Cambios a los elementos del software son implementados apropiadamente y afectan solo las áreas identificadas por la especificación del cambio.

Además, una confección más amplia de este proceso de revisión por un elemento del software individual, los objetivos específicos son identificados en una "Declaración de Objetivos" disponible antes de la reunión de revisión. Se puede aplicar el concepto de revisión técnica a desarrollos nuevos o a actividades del mantenimiento.

**5.2. Resumen.** Una revisión técnica es una evaluación formal de elementos del software. Identifica cualesquiera discrepancias de especificaciones y estándares o provee recomendaciones después de examinar alternativas, o ambos. Este examen requeriría más de una reunión.

**5.3. Responsabilidad Especial.** Los roles para la revisión técnica incluyen:

**5.3.1. Líder.** El jefe de la revisión es responsable por conducir una revisión específica. Esto incluye tareas administrativas pertinentes a la revisión y aseguramiento de que la revisión esta conducida de una manera ordenada. El jefe de la revisión es también responsable por emitir el informe de revisión.

**5.3.2. Registrador.** El registrador es responsable por documentar mantenimiento (por ej. Defectos, inconsistencias, omisiones y ambigüedades), decisiones y recomendaciones hechas por el equipo de revisión.

**5.3.3 Miembros del equipo.** Cada miembro del equipo es responsable de su propia preparación y por asegurar que la revisión cumpla sus objetivos. Simultáneamente, son responsables de la formulación de recomendaciones en tal modo que la gerencia pueda actuar de una manera oportuna.

**5.4 Entrada.** La entrada mínima al proceso de revisión técnica es como sigue:

- (1) Una definición de objetivos para la revisión técnica.

- (2) Los elementos software que está siendo examinados.
- (3) Especificaciones para los elementos software que está siendo examinados.
- (4) Todos los planes, estándares o pautas contra los cuales los elementos software están siendo examinados o reconocidos.

Adicionalmente se puede aplicar material de referencia utilizable para responsabilidades individuales de los elementos software, cuando son requeridos por el líder de la revisión.

## **5.5. Criterios de Entrada.**

**5.5.1. Autorización:** Lo necesario para conducir revisiones técnicas de elementos software específico esta definido por el documento de la planificación del proyecto.

Las revisiones técnicas no catalogadas pueden ser conducidas como requisito de administración funcional, administración del proyecto, administración de la calidad de software o ingeniería de software de acuerdo a procedimientos locales.

**5.5.2. Iniciando Eventos.** No se conduce una revisión técnica hasta que:

- (1) Se establece una declaración de objetivos para la revisión,
- (2) Las responsabilidades individuales para los elementos de software muestran facilidad para la revisión.
- (3) El líder de la revisión técnica esta satisfecho con que los elementos del software son suficientemente completos para una revisión técnica que valga la pena.

## **5.6. Procedimientos**

**5.6.1. Planificación:** El jefe de la revisión es asignado responsable de las siguientes actividades.

- (1) Identifica, con apropiado soporte la administración del equipo de revisión.
- (2) Programa y anuncia el lugar de reunión.
- (3) Distribuye los materiales de entrada para los participantes, permitiéndoles adecuar el tiempo para su preparación.

**5.6.2. Apreciación Global:** Una persona técnicamente calificada maneja una sesión de vistazo general ara el equipo de revisión cuando sea requerido

por el jefe de la revisión. Este vistazo general puede ocurrir como una parte de la reunión o de revisión (ver 5.6.4.) o como una reunión aislada

**5.6.3.Preparación** : Durante los pasos de preparación cada miembro del equipo de revisión individualmente examina los elementos del software y materiales relacionados preparándolos para la reunión de revisión.

**5.6.4.Examen** A la reunión de revisión el equipo completo revisa los elementos del software evaluando su condición relativa a aplicar pautas, especificaciones y estándares o evaluando soluciones alternativas del problema. Específicamente el equipo de revisión desarrolla las siguientes tareas:

- (1) Examina los elementos software y verifica que cumplan con las especificaciones y estándares para los cuales ellos deben ajustarse. Todas las desviaciones de las especificaciones y estándares son registradas.
- (2) Documenta productos técnicos, recomendaciones relacionadas y es responsable de hacer las ediciones.
- (3) Identifica otras ediciones que pueda ser catalogadas.

Después de que los elementos del software han sido revisados, un informe es generado para documentar la reunión, la lista de deficiencias encontrada en los elementos del software y describe las recomendaciones para la administración.

Cuando las deficiencias son lo suficientemente críticas o numerosas, el líder de la revisión recomendará que un proceso de revisión adicional será aplicado a los elementos de software después de que las deficiencias sean resueltas.

**5.7.Criterio de Salida.** Una revisión técnica esta completo cuando

- (1) Todos los sucesos identificados en el Informe de Objetivos de la revisión han sido registrados
- (2) El reporte de revisión técnica (ver 5.8) ha sido terminado.

**5.8.Salida.** El resultado del proceso de revisión técnica es un reporte de revisión técnica que identifica lo siguiente:

- (1) Los miembros del grupo de revisión
- (2) Los elementos de software revisados
- (3) Entradas específicas para la revisión
- (4) Una lista de deficiencias de los elementos del software, sin resolver
- (5) Una lista de ediciones de gerencia.
- (6) Ítem de acción de pertenencia y status

- (7) Cualquier recomendación hecha por el grupo de revisión sobre como poner fin a deficiencias.

Aunque estos grupos de estándares son los mínimos requeridos para la emisión del informe, es el punto de partida de normas locales para prescribir cualquier informe de requerimientos.

**5.8.Auditoria.** El reporte de revisión técnica es un ítem de auditoria. Se recomienda que este informe sea detectable para/desde los documentos de planificación del proyecto apropiados (por ej, plan de desarrollo del software, plan de aseguramiento de calidad del software).

## 6. Procesos de Inspección del Software

**6.1.Objetivo.** El objetivo de una inspección del software es descubrir e identificar los defectos de los elementos del software. Esta es un riguroso examen formal que hace lo siguiente:

- (1) Verifica que los elementos del software satisfacen las características técnicas.
- (2) Verifica que los elementos del software conforman normas aplicables
- (3) Identifica desviación de normas y especificaciones
- (4) Reúne datos de administración del software (por ej. defectos y datos de gestión)
- (5) No examina alternativas o ediciones estilizadas.

**6.2.Resumen.** Inspecciones de software son conducidas por parejas y típicamente comprende de 3 a 6 participantes. El proceso es guiado por un moderador imparcial de los elementos del software bajo examen. El moderador no es el autor. La resolución de defectos es obligatoria y la reanudación es formalmente verificada. Datos del defecto será sistemáticamente reunido y almacenado en una base de datos de inspección. Requisitos mínimos para la colección de los datos son presentados en 6.10. Cuando este dato es analizado, las etapas pueden ser tomadas para mejorar el producto y el desarrollo del proceso y evaluar la efectividad del proceso de inspección.

**6.3.Responsabilidades Especiales.** Todos los miembros del grupo, incluyendo esos con papeles especiales, son inspectores. Los papeles especiales requeridos por el proceso de inspección incluye:

**6.3.1.Moderador:** El moderador es el jefe que planea y dirige la reunión para el proceso de inspección y es responsable de producir el reporte de inspección (ver 6.8). El moderador puede también hacer el papel de registrador.

**6.3.2.Lector:** En las reuniones, el lector introduce al grupo de inspección a través de los elementos del software en una forma lógica y comprensiva, generalmente parafraseando secciones del trabajo y leyendo línea por línea cuando es requerido.

**6.3.3.Registrador:** El registrador es encargado de documentar los defectos detectados en las reuniones y de registrar los datos de inspección requeridos para el proceso de análisis.

**6.3.4. Inspector:** El papel del inspector es identificar y describir defectos en los elementos del software. Inspectores deben ser conocedores del proceso de inspección. Ellos son escogidos para representar puntos de vista diferentes en la reunión (por ej, requerimientos, plan, codificación, prueba, prueba independiente, administración del proyecto, administración de calidad). Solo los puntos de vista pertinentes a la inspección son presentados.

**6.3.5. Autor:** El autor es responsable de la reunión de los elementos del software según un criterio de entrada de inspección, para contribuir en la inspección basada en especial entendimiento de los elementos del software y para desarrollar cualquier reanudación requerida, para hacer del encuentro de los elementos del software, su inspección el criterio de éxito.

**6.4. Entrada.** Las entradas necesarias para la inspecciones consiste en las siguientes:

- (1) Los elementos de software a inspeccionar. Deben satisfacer los criterios de la entrada de la inspección.
- (2) La especificación del elemento de software aceptado.
- (3) La lista de inspección (ver apéndice B)
- (4) Cualquier estándar y pautas contra el cual el elemento de software esta siendo inspeccionado.
- (5) Toda forma necesaria para informar de la inspección.

Para la revisión, es necesaria la lista de defectos de la inspección previa.

## **6.5. Criterio de Entrada**

**6.5.1. Autorización.** Se planean inspecciones por y documentadas en los apropiados documentos de planificación de proyectos (por ej, plan del proyecto total, plan de aseguramiento de calidad de software o plan de validación y verificación de Software)

**6.5.2. Comienzo del Evento:** Los procesos de inspección de software pueden ser disparados por lo siguiente:

- (1) Disponibilidad de los elementos del Software
- (2) Acatamiento del diseño del proyecto
- (3) Complacencia del horario SQAP o SVVP
- (4) Revisión programada

- (5) En la necesidad de administración funcional, administración del proyecto, administración de la calidad del software o ingeniería del software.

### **6.5.3.Criterios de la entrada mínimos antes de planear la inspección:**

Antes de la planear la inspección, los siguientes criterios de entrada deberán satisfacerse

- (1) Los elementos del software conforme a las normas de contenido y formato del proyecto
- (2) Se satisface todo hito anterior, así como identificado en el documento de planificación apropiado.
- (3) Toda documentación de soporte requerida esta disponible
- (4) Para una revisión, todos los ítem anotados en la lista de defectos se debe satisfacer.

## **6.6.Procedimientos**

**6.6.1.Planificación** : Durante la planificación, el autor ensambla los materiales en el paquete de inspección para el moderador.

El moderador es responsable de asegurar que los materiales satisfagan el criterio de entrada de inspección. El moderador es también responsable de asegurar la selección del grupo de inspección y la asignación de sus papeles en la reunión de inspección, por programar los encuentro de inspección y por la distribución de los materiales de inspección.

**6.6.2.Apreciación global:** Si se programa una presentación de apreciación global de los elementos del software inspeccionados, es conducida por el moderador y el autor hace la presentación. Esta apreciación global es usado para educar a otros inspectores acerca de los elementos del software y puede asistir también otro personal del proyecto quienes pueden sacar provecho de la presentación.

**6.6.3.Preparación:** Es responsabilidad individual de todos los inspectores llegar a ser completamente familiar con el elemento del software, usando la lista de inspección, la información de la apreciación global y la especificación del software.

**6.6.4.Examen:** La reunión de inspección seguirá la siguiente agenda:

**6.6.4.1.Introducción** . Para iniciar el encuentro, el moderador presenta los participantes y describe sus roles. Las condiciones del moderador el propósito de la inspección y los inspectores centran sus esfuerzos hacia la detección de defectos. No caza-soluciones. El moderador recuerda a los inspectores dirigir sus comentarios al lector y comentar solo en el producto,

no el autor. También puede ser útil resolver cualquier inquietud de procedimiento especial elevado por los inspectores.

**6.6.4.2. Establecer preparativos.** El moderador pregunta por el total de registros y tiempo de preparación individual en el reporte de inspección. Es responsabilidad del moderador reprogramar la reunión si los inspectores no están adecuadamente preparados.

**6.6.4.3. Revisar la lista de Inspección.** El moderador revisa la lista de inspección con el equipo para asegurar que el producto ha sido adecuadamente estudiado antes de la reunión de inspección.

**6.6.4.4. Leer elementos del Software y registrar defectos.** El lector presenta el material al equipo de inspección. El equipo de inspección examina el software objetivamente y el moderador enfoca esta parte de la reunión en crear la lista de defectos. El registrador inserta cada defecto, localización, descripción y clasificación en la lista de defecto. Durante este tiempo, el autor responde cualquier pregunta específica y contribuye a la detección del defecto basado en su especial conocimiento del elemento del software.

**6.6.4.5. Revisar la lista de defectos.** Al final de la reunión de inspección, el moderador debe tener la lista de defecto revisada con el equipo para asegurar su completitud y exactitud.

**6.6.4.6. Tomar decisión de salida.** El propósito de la decisión de salida es brindar una clausura sin ambigüedades a la reunión de inspección. La decisión de salida determina si el material reúne el criterio de éxito de inspección y prescribe cualquier verificación de reanudación apropiada. Específicamente el equipo de inspección identifica la disposición de los elementos del software como una de las siguientes:

- (1) Aceptado : El elemento del software es aceptado como es o con solo una reanudación menor (por ej. que no requiere futura verificación)
- (2) Verifica reanudación: El elemento de software es aceptado después de que el moderador verifique la reanudación (corrección).
- (3) Revisión: Programa una revisión para verificar reanudación después de la revisión.

Como mínimo la revisión examina las áreas cambiadas para resolver defectos identificados en la ultima inspección.

**6.6.5.Reanudación.** Durante la reanudación, el autor revisa los materiales, dirigiendo todos los ítem en la lista de defecto de inspección.

**6.6.6.Continuar.** El proceso de la inspección del software provee continuación en dos niveles:

- (1) Verificando reanudación por disposición de la inspección.
- (2) Informando datos de inspección.

**6.7.Criterio de Salida.** Un criterio de salida aplicado a todas las inspecciones es que todo los defectos que han sido encontrados estén resueltos. Cada proyecto desarrollara sus propios criterios para encontrar las necesidades de sus productos específicos y ambiente de desarrollo.

El moderador asegurara que el material coincide con el criterio de salida y que los resultados de inspección han sido reportados antes certificando que la inspección es completada.

**6.8.Salida.** Para cada elemento de software los reportes producidos por la inspección son:

- (1) La lista de defecto, que contiene la localización, descripción y categorización del defecto.
- (2) El resumen del defecto de la inspección totalizando el numero de defectos por cada categoría.
- (3) El reporte de inspección, que contiene:
  - (a) El numero de participantes
  - (b) La duración de la reunión de inspección
  - (c) El tamaño de los materiales inspeccionados
  - (d) El total de tiempo de preparación del equipo de inspección
  - (e) La disposición del elemento de software
  - (f) Una estimación del esfuerzo de reanudación y la fecha de realización de la reanudación.

Los reportes de información son retenidos para análisis subsecuente como esta prescrito en los documentos de planificación del proyecto apropiados. Aunque este grupo de estándares son requisitos mínimos para el contenido del reporte, se deja a normas locales prescribir cualquier estructura del informe de requisitos.

**6.9.Auditoria.** Auditoria es proporcionada por lo siguiente:

- (1) Procedimientos de inspección documentados
- (2) Reportes de inspección retenidos
- (3) Dato de defecto de la inspección retenido

**6.10.Requisitos de colección de datos.** Las inspecciones de software suministran datos para el análisis de calidad de los elementos de software, la efectividad de los procedimientos del desarrollo y la eficiencia de los procesos de inspección. Este análisis permite, que los defectos que son identificados en una reunión de inspección estén agrupados por tipo de defecto, clase y gravedad.

6.10.1. El tipo de defecto identifica un atributo de los elementos del software y por ejemplo puede ser identificando como afectando los elementos del software respecto a

- (1) Acatamiento de estándares
- (2) Capacidad
- (3) Procedimiento
- (4) Interfaces
- (5) Descripción

6.10.2.La clase de defecto caracteriza evidencia de no concordancia y por ejemplo puede estar agrupado como

- (1) Perdido
- (2) Errado
- (3) Extra

6.10.3.Las clases de defectos adicionales usados cuando inspeccionan documentos pueden incluir

- (1) Ambigüedad
- (2) Inconsistencia

6.10.4.Defectos están agrupados por gravedad, por ejemplo, así

- (1) Mayor. Defectos que deben dar como resultado falla del elemento de software o un distanciamiento de la especificación
- (3) Menor. Defectos que afectan solo los aspectos no funcionales del elemento de software.

Además para contar los defectos por tipo, clase y gravedad, el dato de inspección guardado deberá contener la identificación del elemento del software, la fecha y la hora de la inspección, el moderador el tiempo de preparación e inspección, el volumen de los materiales inspeccionados, y la disposición del elemento de software inspeccionado. El registro de esta información es usado como guía local de esfuerzo y volumen de la inspección. En la practica se sugiere mas de 1.5 horas de preparación, mas de 2 horas en la reunión y cualesquiera que sea el volumen se maneja cómodamente en estos periodos de tiempo.

La administración de datos de la inspección requiere capacidades para el almacenamiento, entrada, acceso, actualización y resumen y reporte de los defectos agrupados. La frecuencia y tipos de reporte de análisis de inspección y su distribución, se dejan al usuario de este estándar.

## **7.Los procesos de revisión**

**7.1.Objetivo.** El objetivo de la revisión es evaluar un elemento de software. Aunque esta muy asociado con examinación del código, este proceso también es aplicable a otros elementos del software (por ejemplo, arquitectura del diseño, diseño detallado, prueba planes / procedimientos y procedimientos de control de cambio). Los mayores objetivos son encontrar defectos, omisiones y contradicciones; perfeccionar los elementos del software; y considerar implementaciones alternativas.

Otros objetivos importantes del proceso de revisión incluye el intercambio de variación de estilo y técnicas y educación de los participantes. Una revisión puede descubrir problemas de eficiencia y calidad en el código, problemas de modularidad en diseño o especificaciones de diseño sin probar.

La revisión puede ser usada para nuevos desarrollos o en actividades de mantenimiento

**7.1.Resumen.** Durante la reunión de revisión, el autor hace una presentación de los elementos del software bajo revisión. Este es seguido por una discusión general de los participantes después de los cuales el presentador “recorre” el elemento del software en detalle. Como progreso de revisión, errores, cambios sugeridos y mejoras son señaladas y escritas. Cuando la revisión es finalizada, las notas son consolidadas dentro de un informe, el cual es distribuido al autor y otro personal apropiado. Una copia del informe es puesta en el archivo del proyecto para auditoria.

**7.2.Responsabilidades Especiales.** Los siguientes roles son establecidos para la revisión:

**7.3.1.Moderador.** El moderador de la revisión es responsable de la conducción de una revisión específica, manejando las tareas administrativas pertinentes a la revisión, y asegurando que la revisión es conducida de manera ordenada. El moderador es también responsable de preparar los objetivos para guiar al equipo a través de la revisión.

**7.3.2.Registrador.** El registrador es responsable por escribir todos los comentarios hechos durante la revisión pertinentes a errores encontrados, preguntas de estilo, omisiones, contradicciones, sugerencias para improvisación, o propuestas alternativas

**7.3.3.Autor.** El autor es la persona responsable de que los elementos del software estén siendo examinados y presenta los materiales. Cada miembro

del equipo de revisión es responsable de revisar cualquier material de entrada previo a la revisión y participar durante la revisión para asegurar que este cumple los objetivos.

Los papeles pueden ser compartidos entre los miembros de la revisión. El autor y el moderador de la revisión pueden ser la misma persona.

**7.4.Entrada.** La entrada mínima para el proceso de revisión es la siguiente:

- (1) Una declaración de objetivos para la revisión.
- (2) El elemento software examinado
- (3) Estándares que estén en vigencia para el desarrollo del elemento software.
- (4) Especificaciones de los elementos software bajo revisión.

## **7.5.Criterio de Entrada**

**7.5.1.Autorización.** Lo necesario para conducir la revisión deberá estar establecido en los documentos de planificación apropiados, (por ejemplo, un plan de aseguramiento de calidad del software (SQAP) o un plan de desarrollo del software). Además, bajo cualquiera de estos planes, la terminación de un elemento de software específico impulsa la revisión para estos elementos. Adicionalmente las revisiones deberán estar conducidos durante el desarrollo de los elementos software a los requisitos de administración funcional, administración de calidad, o el autor, de acuerdo a procedimientos locales.

**7.5.2.Comienzo del Evento.** Una revisión es conducida cuando

- (1) El autor de los elementos software ha indicado que esta listo, y esta preparado para presentar los materiales.
- (2) El moderador esta equipado por practicas o estándares locales y ha concluido que el elemento software es suficientemente completo para conducir la revisión.

## **7.6.Procedimientos**

**7.6.1.Planificación.** Durante la fase de planificación, el moderador de revisión hace lo siguiente:

- (1) Identifica el equipo de revisión
- (2) Programa la reunión y selecciona el lugar de encuentro.
- (3) Distribuye todo el material de entrada necesario a los participantes, permitiendo el tiempo de preparación adecuado.

**7.6.2.Apreciación global.** Una presentación de respaldo es hecha por el autor como parte de la reunión de revisión. Antes de la reunión, sin embargo, la preparación individual esta aun requerida.

**7.6.3.Preparación.** Durante la fase de preparación los participantes revisan el material de entrada que les fue distribuido y prepara una lista de inquietudes y eventos ha ser discutidos durante la revisión.

**7.6.4.Examen** Durante la reunión de revisión

- (1) El presentador hace una presentación rápida de los elementos de software examinado.
- (2) El autor camina a través de los elementos de software para que los miembros del equipo de revisión puedan hacer preguntas o crear discusiones relacionado con el elemento de software o documentar su importancia, o ambos.
- (3) El registrador escribe los comentarios y decisiones para incluirlas en el informe de revisión.

Al completarse la revisión, el equipo de revisión puede recomendar una revisión de refuerzo. Este refuerzo deberá seguir los estándares para revisiones y deberá como mínimo cubrir áreas cambiadas por el autor.

Dentro de un tiempo razonable después de la revisión, el líder de la revisión producirá un reporte de revisión detallando los descubrimientos de la revisión. Como mínimo contiene los requerimientos que están en 7.8

**7.7.Criterio de Salida.** El proceso de revisión esta completo cuando

- (1) El total de elementos de software ha sido repasado en detalle
- (2) Todas las deficiencias, omisiones, y sugerencias para mejoramiento han sido anotadas.
- (3) El informe de revisión ha sido producido.

**7.8.Salida.** El resultado del proceso de revisión es un informe de revisión listando sus deficiencias, omisiones, beneficios y sugerencia para mejorar que fueron registradas durante la revisión.

El informe de revisión contiene lo siguiente:

- (1) Identificación de equipo de revisión
- (2) Identificación de los elementos de software examinados.
- (3) La declaración de objetivos que fueron seguidos durante la reunión de revisión
- (4) Una lista de deficiencias anotadas, omisiones, contradicciones y sugerencias para mejorar

- (5) Cualquier recomendación hecha por el equipo de revisión en como acabar con las deficiencias y problemas no resueltas.

Si son sugeridas revisiones de refuerzo, esta sugerencia deberá también mencionarse en el reporte.

Aunque este grupo de estándares son los mínimos requeridos para el contenido del informe, se deja estándares locales para prescribir cualquier formato de informe de requerimientos.

**7.9.Auditoria.** El reporte de revisión, minutos de reunión y otros materiales en los cuales las conclusiones están basadas deben incluirse en los archivos de documentación del proyecto. El moderador de la revisión archiva estos documentos

## **8.El proceso de la Auditoria**

**8.1.Objetivo.** El objetivo de auditar el software es proveer una conformidad objetiva de la complacencia de productos y procesos para certificar adhesión a estándares, pautas, especificaciones y procedimientos.

**8.2.Resumen** Las auditorias son desarrolladas en acuerdo con planes y procedimientos documentados. El plan de auditoria establece un procedimiento para conducir la auditoria y para reforzar la acción en los descubrimientos de auditoria.

En la ejecución de la auditoria, el personal de auditoria evalúa los elementos de software y los procesos para producir su criterio de auditoria objetivo comparado con requerimientos, planes especificaciones o procedimientos, guías y estándares.

Los resultados de la auditoria son documentados y son propuestos a la administración de la organización auditada, al ente que inicia auditoria y a cualquier organización externa identificada en el plan de auditoria. El reporte incluye una lista de los ítem que no cumplen u otros productos para revisión y acción subsiguiente. Cuando se estipula por el plan de auditoria, las recomendaciones son reportadas además de los resultados de auditoria.

**8.3.Responsabilidades especiales** La responsabilidad del jefe del grupo de auditoria es organizar y dirigir la auditoria y coordinar la preparación y edición del informe de auditoria. El jefe de equipo de auditoria asegurara que el equipo de auditoria esta preparado para conducir la auditoria y que los procedimientos de auditoria son desarrollados y producen informes en concordancia con el alcance de la auditoria.

**8.4.Entrada.** Para asegurar el éxito de la auditoria se requiere:

- (1) El propósito y alcance de la auditoria
- (2) Criterio de auditoria objetiva, tales como compendiar requerimientos, planes, especificaciones, procedimientos, pautas y normas
- (3) Los elementos y procesos de software a ser auditados y cualquier historia pertinente.
- (4) Información de antecedentes respecto a responsable de la organización de los productos y procesos auditados (por ejemplo, mapas de la organización)

**8.5.Criterio de Entrada:** La necesidad de una auditoria esta establecida por uno de los siguientes eventos:

- (1) Un proyecto pilar especial ha sido rechazado. La auditoria es iniciada por planes mas tempranos (por ejemplo, plan de aseguramiento de calidad del software, plan de desarrollo de software)
- (2) Partes externas (por ejemplo, agencias reguladoras, o usuarios finales) solicitan una auditoria en una fecha calendario especifico o proyecto pilar. Este estaría en cumplimiento de un requisito del contrato, o como un requisito previo al acuerdo contractual.
- (3) Un elemento de organización local (por ejemplo, administración del proyecto, administración funcional, ingeniería de sistemas, control/aseguramiento de calidad interna) ha requerido la auditoria, estableciendo una necesidad clara y específica.
- (4) Un proyecto pilar especial, fecha calendario, u otro criterio que ha sido encontrado como parte de la carta constitucional de la organización intervenida, en respuesta se inicia la auditoria

## **8.6.Procedimientos**

**8.6.1.Planificación.** La organización a auditar deberá desarrollar y documentar un plan de auditoria para cada auditoria. Este plan, adicional al alcance de la auditoria, identificara lo siguiente:

- (1) Procesos del proyecto a ser examinados (dados como entrada) y el tiempo de ajuste para observación del equipo de auditoria.
- (2) Software requerido a ser examinado (dado como entrada) y su disponibilidad. Cuando se usa una muestra, un método de muestreo valido estadísticamente será usado para establecer el criterio de selección y el tamaño de la muestra.
- (3) Las máscaras de los reportes son identificados (esto es, reporte de resultados, y opcionalmente las recomendaciones sobre los reportes y la definición del formato general). Si se requieren recomendaciones o excluir máscaras, se deben mostrarse explícitamente.
- (4) Distribución de reportes.
- (5) Flujo de actividades requeridas.
- (6) Requerimientos. Actividades necesarias, elementos y procedimientos para encontrar el objetivo de la Auditoria.
- (7) Criterio objetivo de Auditoria: provee la base para determinar complacencia (como entrada).
- (8) Lista de chequeo y procedimientos de la Auditoria.

- (9) Personal de la Auditoria: número requerido, habilidades, experiencia, y responsabilidades.
- (10) Organizaciones involucradas en la Auditoria (por ejemplo, organizaciones cuyos productos y procesos se intervienen)
- (11) Fecha, tiempo, lugar, agenda, y público intencional de sesión de apreciación global (opcional).

El jefe del equipo de Auditoria debe asegurar que el equipo de Auditoria esté preparado e incluir miembros con experiencia y especialización necesaria.

La notificación de la Auditoria se debe proveer a las organizaciones involucradas en una cantidad de tiempo razonable antes de que la Auditoria se ejecute, excepto para Auditorias no anunciadas. La notificación debe ser escrita e incluida en los alcances e identificación de procesos y productos involucrados.

**8.8.2 Apreciación global.** Una reunión opcional de apreciación global en la organización auditada es recomendada para la fase *de partida* de examen de la Auditoria. La reunión de apreciación global, guiada por el jefe de la Auditoria, provee lo siguiente:

- (1) Apreciación global de los acuerdos existentes (por ejemplo, alcance de la Auditoria, plan, contratos relacionados);
- (2) Apreciación global de productos y procesos que intervienen;
- (3) Apreciación global de procesos de Auditoria, sus objetivos, y salidas;
- (4) Contribuciones esperadas de la organización que interviene en el proceso de Auditoria (esto es, número personas a ser entrevistadas, facilidades para la reunión, etc.);
- (5) Especificar horario de la Auditoria.

### **8.6.3.Preparación.**

8.6.3.1. Las siguientes preparaciones se requieren por el equipo de Auditoria:

- (1) Entender la organización: Esto es esencial para identificar funciona y actividades ejecutadas por quién realizó la Auditoria de la organización.
- (2) Entender los productos y procesos: es un requisito previo para que el jefe del equipo entienda acerca de los productos y procesos que están siendo auditados a través de lecturas y sesiones de información.
- (3) Entender el criterio de Auditoria objetiva: es importante que el equipo de Auditoria llegue a familiarizarse con un criterio de Auditoria objetivo usado en la Auditoria.
- (4) Preparación para el reporte de Auditoria: esto es importante escoger los mecanismos de los reportes administrativos que será usado a través de

la Auditoria para desarrollar el informe que sigue el esquema identificado en la planificación de la Auditoria.

(5) Detalles del plan de Auditoria: encontrar métodos apropiados en cada paso en el programa de la Auditoria.

8.6.3.2 Además, el jefe del equipo de la Auditoria debe hacer los arreglos necesarios para

(1) Orientación e instrucción del equipo como se requiera

(2) Facilitar las entrevistas de la Auditoria.

(3) Materiales, documentos y herramientas requeridas para los procedimientos de la Auditoria

(4) Elementos del software para la Auditoria (por ejemplo, documentos, archivos de computador, personal para entrevistar).

(5) Programar las entrevistas.

**8.6.4.Examen.** Los elementos que han sido seleccionados para la Auditoria deben ser evaluados nuevamente con un criterio de Auditoria objetivo. La evidencia debe ser evaluada con profundidad para determinar si estos elementos cumplen con los criterios especificados.

La Auditoria debe:

(1) Revisar procedimientos e instrucciones

(2) Examinar la estructura de interrupción del trabajo

(3) Examina evidencia de implementación y controles balanceados

(4) El personal entrevistado determina el estado y funcionamiento de los procesos y de los productos

(5) Examinar documentos de los elementos.

(6) Probar los elementos.

**8.6.5 Informes.** Después del examen de la Auditoria, el equipo de la Auditoria debe emitir un reporte para la organización revisar y comentar.

El equipo de la Auditoria entrega un resultado anterior antes de los resultados formales. Esto es ejecutado en acuerdo con el reporte del proyecto de revisión, resuelve cualquier inquietud o ambigüedad mientras mantiene objetividad y correcciones. Esto también sirve para asegurar reportes usualmente para agregar consistencia al nivel de reporte detallado y para agregar cualquier información recientemente verificada. Las recomendaciones prácticas son involucrar representantes de quién hizo la Auditoria para revisar resultados, como

(1) Una continuidad, actividad periódica incluida dentro del horario de examen general, o

(2) Como un paso de la clausura del período de examen

Involucrar al auditor de la organización, contribuye para reportar calidad a través de la interacción llevada a cabo y el rescate de cualquier futura.

El equipo de Auditoría debe conducir a una posterior conferencia de Auditoría para revisar con el auditor de auditor de la organización las deficiencias, hallazgos, y (si aplica) recomendaciones. Comentarios y ediciones deben ser resultas por el auditor de la organización.

El informe final de la Auditoría debe entonces ser preparado, aprobado y emitido por el jefe del equipo de la Auditoría a las organizaciones especificadas en el plan de la Auditoría.

**8.7. Criterios de Salida.** Se considerará una Auditoría completa cuando

- (1) Cada elemento (s) dentro de la mira de la Auditoría se ha examinado,
- (2) Los hallazgos han sido presentados al auditor de la organización;
- (3) Respuesta del bosquejo de los hallazgos recibidos y evaluados;
- (4) Hallazgos finales que han sido presentados formalmente al auditor de la organización y a la entidad;
- (5) Los informes de Auditoría han sido preparados y sometidos a destinatarios designados en el plan de la Auditoría;
- (6) El informe de recomendaciones, si es requerido por el plan, ha sido preparado y sometido a los designados en el plan de la Auditoría,
- (7) Todo el flujo de las acciones de la organización incluida en la meta (o contrato) de la Auditoría que ha sido ejecutada.

**8.8 Salida.** Como un esquema estándar para informes de la Auditoría, el proyecto y reportes de Auditoría final deben, como mínimo, contener lo siguiente:

- (1) Identificación de la Auditoría. Título del informe, auditor de la organización, organización auditada, fecha de la Auditoría.
- (2) Alcance. El alcance de la Auditoría, incluye una enumeración de los estándares, especificaciones, prácticas, y procedimientos que constituyen el criterio objetivo de Auditoría, contra los cuales se condujo la Auditoría de los elementos del software y procesos auditados.
- (3) Conclusiones. Un resumen e interpretación de los hallazgos de la Auditoría, incluyendo los ítem no conformes.
- (4) Sinopsis. Una lista de todos los elementos del software auditados, procesos, y hallazgos asociados,
- (5) Flujo. El tipo y tiempo de las actividades de la Auditoría.

Adicionalmente, cuando se estipula en el plan de la Auditoria, se deben proveer recomendaciones al auditor de la organización, o la entidad que comenzó la Auditoria. Las recomendaciones deben ser reportadas por separado de los resultados.

**8.9 Auditoria.** El material de documentación del proceso de Auditoria debe ser mantenido por la organización para un período estipulado de tiempo subsiguiente a la Auditoria, e incluye lo siguiente:

- (1) Todos los programas de trabajo, listas de chequeo, etc., son completamente anotadas.
- (2) Equipo de trabajo
- (3) Notas de interventoría, notas de observaciones.
- (4) Completas pruebas de evidencia
- (5) Copia de ítem examinados, anotados
- (6) Informes del proyecto con respuesta del auditor de la organización
- (7) Memorando del flujo como sea necesarios

## **Apéndices**

(Estos apéndices no son parte de la norma IEEE I028-I988, para Revisión y Auditoria del Software, pero se incluyen sólo por información.)

### **Apéndice A**

#### **Guía para Aplicación de Procesos del Software Crítico**

##### **A1.**

La norma ANSI/IEEE 730-1984[A1] ha identificado un conjunto mínimo de procesos de revisión y Auditoria aplicado para software crítico (por ejemplo, cuando una falla puede ser impacto de seguridad, o causa aumento financiero o impacto social).

- (1) Revisión de requerimientos del Software(SRR)
- (2) Revisión del diseño preliminar (PDR)
- (3) Revisión del diseño crítico (CDR)
- (4) Revisión del plan de verificación y validación del Software (SVVPR)
- (5) Auditoria Funcional (FA)
- (6) Auditoria Física (PA)
- (7) Auditoria en proceso (IPA)
- (8) Revisión de Dirección (MR)

##### **A2**

A2.1 La meta de este Apéndice es proveer un posible mapa de aplicación de procesos específicos que se requiere identificar en esta norma, para procesos candidatos para la reunión que lo necesite. La Selección de un proceso específico debe estar como local, contractual, o condiciones específicas del proyecto dictadas.

Por ejemplo:

- (1) El proceso de la inspección del software debe ser usado para encontrar la aplicación de revisión del plan de verificación y validación del software necesario.
- (2) El proceso de revisión gerencial debe ser usado para encontrar la aplicación de revisión de dirección necesaria.

##### **Tabla A1**

Requerimientos Norma ANSI/IEEE 730-1.984	SRR	PDR	CDR	SVVPR	FCA	PCA	IPA	MR
Revisión gerencial			X					X
Revisión técnica	X	X	X	X				
Inspección de software	X	X	X	X				
Revisión	*	*	*	*				
Auditoria					X	X	X	**

\*Como práctica común en muchas compañías, la revisión es permitida cuando la inspección del software pudo ser aplicado. Esto surgió sin embargo para que el software crítico fuera presentado en la norma ANSI/IEEE 730.1984, sólo técnicas más formales y rigurosas son usadas. Una diferencia significativa esta desarrollada por el autor en la presentación del trabajo.

\*\*Los procesos de Auditoria deben ser adaptados en esta aplicación o administración para encontrar las necesidades de una revisión de los directivos.

Un proceso particular no es necesario que esté descrito por todas las aplicaciones. Por ejemplo, usando el mismo proceso para todos los SRR, PDR, y CDR puede no tener un costo efectivo o puede basado en características técnicas o económicas.

A2.2. Similarmente, una combinación de procesos distintos puede ser aplicado para encontrar las necesidades de una aplicación específica. Un ejemplo lo provee el examen adecuado de un SRR por un sistema muy grande y complejo. SRR podría estar subdividido en tres categorías.

- (1) Satisfacción del cliente con funcionalidad y ejecución con consideración por viabilidad del negocio y técnica.
- (2) Complementos técnicos y correctivos de requerimientos
- (3) Aptitud para utilizar el documento de especificaciones como los vistos en el desarrollo, las pruebas, y la organización del mantenimiento

En este ejemplo, sería apropiado usar una revisión gerencial para examinar la primer categoría, una revisión técnica para la segunda categoría, y una inspección del software para verificar la aptitud para utilizar el mismo documento

La tabla A1 muestra cartografías posibles de revisión y Auditoria de procesos para las aplicaciones referenciadas por la norma ANSI/IEEE 730-1984 (A1).

### **A3.**

Las referencias siguientes aplican a los Apéndices del A al C:

[A1] Norma ANSI/IEEE 730-1984, Norma IEEE para el Plan de Aseguramiento de la Calidad del Software.

[A2] Norma ANSI/IEEE 1016-1987, IEEE Recomendaciones Prácticas para Descripciones de Diseño de Software.

## **APÉNDICE B**

### **GUÍA PARA APLICACIÓN DE REVISIONES ESPECÍFICAS**

Este Apéndice ha sido adicionado para guiar en la aplicación de procesos de revisión y para delinear algunas aplicaciones específicas.

A través de los procesos de revisión definidos por esta norma se tienen distintos objetivos, la Tabla B1 está provista para acentuar ampliamente la distinción de procesos.

Además, las siguientes aplicaciones de procesos de revisión son presentadas

- (1) Revisión de requerimientos de software (SRR)
- (2) Revisión de diseño preliminar (PDR)
- (3) Revisión de diseño crítico (CDR)
- (4) Revisión del plan de validación y verificación del software (SVVPR)
- (5) Revisión gerencial (MR)

#### **B1 Revisión de Requerimientos del Software (SRR)**

La SRR está hecha para asegurar la suficiencia de los requisitos indicados en las especificaciones de requerimientos de software. Ver Norma ANSI/IEEE 730.1984 (A1)

La SRR es una evaluación de las especificaciones de los requerimientos del software (SRS) para asegurar suficiencia, viabilidad técnica, y complacencia. Por otra parte la SRR se sostiene con la participación del usuario para examinar la SRS para verificar que no es ambigua, comprobable, consistente, modificable, detectable y usable durante la fase de operación y mantenimiento. Todos los elementos de la organización (Por ejemplo diseño de software, pruebas del software, aseguramiento de calidad del software, ingeniería de sistemas, comercialización, manufactura y servicio de campo) que son directamente afectadas por los requerimientos que se deben contemplar.

La lista de revisión para la revisión de requerimientos del software debería incluir, pero no está limitada a lo siguiente:

- (1) Detección y complacencia de los requerimientos del siguiente nivel de especificaciones (Tales como especificaciones del sistema o especificaciones de los requerimientos del usuario).
- (2) Suficiencia en análisis racional para cualquier requisito generado.
- (3) Compatibilidad de interfases externas (hardware y software)
- (4) Suficiencia de los requisitos para la interfase hombre-máquina
- (5) Disponibilidad de constantes y tablas para cálculos
- (6) Prueba de la capacidad de los requisitos funcionales

- (7) Suficiencia y complacencia de la verificación y aceptación de requerimientos
- (8) Conformidad con la especificación de requerimientos estándar
- (9) Suficiencia y viabilidad de funcionamiento de los requerimientos
- (10) Suficiencia de consideración de factores humanos

## B2. Revisión de diseño preliminar (PDR)

B2.1 La PDR esta hecha para evaluar la suficiencia técnica del diseño preliminar del software como representación en una versión preliminar de la Descripción del Diseño del Software. Ver norma ANSI/IEEE 730-1984 (A1)

Específicamente, la necesidad es para aplicar un proceso de revisión que asegure el progreso, la consistencia, y la suficiencia técnica de acercamiento al diseño seleccionado.

Todos los elementos organizacionales (Por ejemplo prueba de software, aseguramiento de calidad del software, ingeniería de sistemas, comercialización, manufactura y servicio de campo) que son directamente afectados por la descripción del diseño del software debería participar.

Los siguientes puntos deberían ser incluidos en la lista de revisión para la revisión del diseño preliminar (PDR):

- (1) Compatibilidad de las interfases entre software, hardware y usuario final
- (2) Cumplimiento de los requerimientos contenidos en la SRS
- (3) Conformidad documentación de diseño estándar

B2.2. Consideraciones que deben ser tratadas por la lista de chequeo para la revisión del diseño preliminar

Tabla B1

Categoría y Atributos	Revisión Gerencial	Revisión Técnica	Inspección del Software	Revisión
Objetivo	Asegurar el progreso. Recomendar acciones correctivas Asegurar una apropiada distribución	Evaluar la conformidad con las especificaciones y el plan. Asegurar la integridad de los cambios.	Detectar e identificar defectos. Verificar resolución	Detectar defectos. Examinar alternativas. Foro para aprender

	de recursos			
<b>Controles Delegados</b>				
Toma de decisiones	Equipo gerencial planea conducta. Las decisiones son tomadas en la reunión o como resultado de las recomendaciones.	Peticiones del equipo de revisión. Administración o dirección técnica para actuar sobre las recomendaciones	El equipo elige las disposiciones de los productos predefinidos. Los defectos deben ser eliminados	Todas las decisiones tomadas por el productor. Los cambios son prerrogativa del productor.
Verificación de cambios	La verificación de cambios antes de otros controles del proyecto.	El líder verifica como parte del reporte de revisión	El moderador verifica la reanudación	La verificación de cambios antes de otros controles del proyecto
<b>Dinámica del grupo</b>				
Tamaño recomendado	Dos ó más personas	Tres o más personas	De tres a seis personas	De dos a siete personas
Atención	Administración, dirección técnica, y combinación de las dos	Dirección técnica y combinación de las dos	Agrupar grupos de dos con atención documentada	Dirección técnica combinación de las dos
Dirección	Usualmente el encargado responsable	Usualmente el ingeniero líder	Asesor entrenado	Usualmente el productor
<b>Procedimientos</b>				
Volumen de material	Media a alto, dependiendo de la declaración específica de los	Media a alto, dependiendo de la declaración específica de los objetivos	Relativamente baja	Relativamente baja

	objetivos para la reunión	para la reunión		
Presentador	Representante del proyecto	Representante del elemento del software	Presentado por el lector con excepción del productor	Usualmente el productor
Colección de datos	Según los requeridos por las políticas estándar o planes aplicables	No es un requerimiento formal del proyecto. Puede ser hecho localmente	Formalmente requerido	No es un requerimiento formal del proyecto. Puede ser hecho localmente
Reportes de salida	Reporte de la revisión gerencial	Reportes de la revisión técnica	Lista de defectos y resumen. Reporte de inspección	Reporte de revisión
Entradas de la base de datos	Cualquier cambio de horario debe ser entrado en la base de datos de seguimiento del proyecto	No es un requerimiento formal la base de datos	Guardar número de defectos, características, severidad y cualidades de la reunión	No es un requerimiento formal la base de datos

(PDR) incluye (pero no se limita a):

- (1) Interfases del software
- (2) Interfases del hardware
- (3) Fondo para la configuración
- (4) Descripción funcional
- (5) Análisis racional de la selección alternativa de diseño
- (6) Ambiente operativo estándar
- (7) Requerimientos de factor humano
- (8) Requerimientos de documentación
- (9) Consideraciones de prueba
- (10) Manufactura
- (11) Mantenimiento
- (12) Asunciones

(13) Riesgos

Mas información está disponible en la norma ANSI/IEEE 1016-1987 (A2)

**B3 Revisión del diseño crítico (CDR)**

La CDR esta hecha para determinar la aceptabilidad del diseño del software detallado según lo representado en la descripción del diseño del software que satisfacen los requerimientos de las especificaciones de los requerimientos de software. Ver norma ANSI/IEEE 730-1984 (A1)

La CDR es una evaluación de la descripción del diseño del software (SDD) para evaluar su suficiencia técnica, completo y correcto. Otras consideraciones que incluye es su compatibilidad con otro software y hardware (donde el producto es requerido para interactuar), y la técnica, costo, y riesgos del horario del diseño del producto.

Todos los otros elementos organizacionales que impongan requerimientos o que sean afectados por el diseño que debe ser participativo (por ejemplo diseño del software, pruebas de software, e ingeniería de sistemas).

Los siguientes puntos deben ser incluidos en la lista de chequeo de revisión para revisión de diseño crítico

- (1) La compatibilidad del diseño detallado con la SRS
- (2) Disponibilidad de datos en forma de diagramas lógicos, algoritmos, cartas de asignación de almacenamiento, y representación del diseño detallado (por ejemplo, organigrama, lenguaje de diseño del programa) para establecer integridad del diseño.
- (3) Compatibilidad y completos los requerimientos de diseño de la interfase.
- (4) Suficiencia y completos los algoritmos y ecuaciones
- (5) Correcciones y conveniencia de descripciones lógicas que deben ser autorizadas
- (6) Exactitud técnica y actualización de toda la documentación de pruebas disponible y su compatibilidad con los requerimientos de pruebas de la SRS
- (7) Los requerimientos para soportar y probar el software y hardware para ser usados en el desarrollo del producto
- (8) La inclusión en el diseño final de flujo funcional, tiempo, tamaño, requerimientos de almacenamiento, mapas de memoria, base de datos y otros factores de funcionamiento.

**B4. Verificación y Validación del Software (SVVPR)**

La revisión del plan de verificación y validación del software esta hecho para evaluar la suficiencia y completa la verificación y validación de los métodos descritos en la SVVP. Ver norma ANSI/IEEE 730-1984 (A1).

La SVVP es una evaluación del plan de verificación y validación del software completo (SVVP). Desde que la SVVP pueda ser desarrollada incrementalmente, pueden ser requeridas múltiples revisiones. Estas revisiones son hechas para asegurar que la verificación y la validación de los métodos descritos en la SVVP son adecuados y proveerán datos de evaluación completos.

Todos los elementos organizacionales que imponen requerimientos o son afectados por la SVVP deben participar. Este grupo debe incluir ingeniería de sistemas, comercialización de software, diseños de software, pruebas de software, aseguramiento de calidad de software, clientes, y usuarios.

Los siguientes puntos deben ser incluidos como requerimientos de revisión para la revisión del plan de verificación y validación del software:

- (1) Todos los métodos de verificación y validación junto con un criterio completo para asegurar seguimiento para, y compatibilidad con, la funcionalidad y requerimientos de funcionamiento expresados en la SRS.
- (2) Los métodos a ser usados para verificar y validar que el producto satisfaga los requerimientos de la SRS; esto debe incluir aplicación de procesos de revisión (por ejemplo revisión técnica, inspección o revisión), demostración, análisis y pruebas.
- (3) Reportes que documenten adecuadamente los resultados de todas las revisiones, Auditorias, y pruebas basados en los requerimientos dados en la SVVP.
- (4) Descripción adecuada de la configuración del software a ser probado incluyendo pruebas soporte de software y hardware
- (5) Plan de pruebas, procedimientos de pruebas, casos de pruebas, y datos de prueba para asegurar que todos los requerimientos especificados están probados y que las instrucciones de las pruebas son claras y concisas.
- (6) Procedimientos de prueba y casos de prueba para asegurar que las entradas de prueba y criterios de salida están definidos adecuadamente.
- (7) Cronograma de plan de pruebas y casos de pruebas identificando pruebas a ser hechas, cuando y por quién.
- (8) Conformidad con los estándares de documentación de pruebas

#### B5. Revisión gerencial

Estas revisiones son hechas periódicamente para determinar la ejecución del plan (SQA). Estas revisiones deberían ser hechas por elementos

organizacionales independientes de la unidad que esta siendo auditada, o por terceros calificados. Ver Norma ANSI/IEEE 730-1984 (A1)

La frecuencia y estructura previstas de la (Plan SQL) revisión gerencial debería estar indicada en el plan SQL. Ellos deberían conducir la dirección del programa gerencial.

Aunque la aplicación del proceso de revisión gerencial reúne estos requerimientos de revisión, una alternativa es ejecutar la Auditoria de los sistemas de calidad (Como se presenta en el Apéndice C) para extender un grado de independencia y autoridad.

## **Apéndice C**

### **Una guía para uso de Auditoria específica**

Dirigir en el uso de procesos de Auditoria, y delinear algunas aplicaciones específicas, ha sido incluido este apéndice. Las siguientes aplicaciones de procesos de auditoria son presentados:

- (1) Auditoria funcional (FA)
- (2) Auditoria física (PA)
- (3) Auditoria en procesos (IPA)
- (4) Auditoria de sistemas de calidad (QSA)

Donde un auto examen (esto es, uso del proceso de Auditoria de un elemento(s) independiente no organizacional del área auditada) es requerido, el proceso de Auditoria general es aplicado, con el requerimiento de independencia liberado.

#### **C1 La Auditoria Funcional (FA)**

El objetivo de la Auditoria funcional (FA) es proveer una evaluación independiente de los productos software, verificando que la configuración de la funcionalidad y funcionamiento actual de los ítem sea consistente con la especificación de requerimientos. Específicamente, esta Auditoria es realizada antes de la entrega del software para verificar que todos los requerimientos especificados en la especificación de requerimientos del software han sido satisfechos. Ver norma ANSI/IEEE-730-1984 (A1).

C1.1. Central para la FA es la identificación de los elementos del software a ser auditados por

- 1) Nomenclatura
- 2) Número de identificación de la especificación
- 3) Número de ítem de la configuración

C1.2. Además, materiales proporcionados por la organización auditada incluyen:

- (1) Especificación de requerimientos del software
- (2) Copia del objeto del código
- (3) Lista actual de la configuración de los ítem específicos liberados
- (4) Cuando este disponible, el estado de los programas de prueba para probar los ítem configurados con el equipo de prueba automático.
- (5) Reporte de verificación y validación del software
- (6) Todos los reportes de la Auditoria en proceso de los ítem bajo Auditoria.

- (7) Cualquier otra documentación (por ejemplo, planes, especificaciones, procedimientos y reportes) para los ítem configurados
- (8) Lista de todas las pruebas funcionales satisfactorias
- (9) Lista de todo lo planeado, lo no ejecutado y las pruebas funcionales

C1.3. Las siguientes actividades son requeridas para conducir una Auditoria funcional:

- (1) Una Auditoria de la documentación de pruebas formales debe ser ejecutada con datos de prueba. Los resultados deben ser chequeados por completo y todas las deficiencias documentadas.
- (2) Una Auditoria del reporte de verificación y validación del software debe ser hecho para validar que el reporte está completo y describe totalmente el esfuerzo de la prueba. Todos los cambios aprobados (acciones correctivas y reporte de problemas) deben ser revisados para asegurar que ellos han sido técnicamente incorporados y verificados.
- (3) Las actualizaciones de los documentos previamente entregados deben ser revisados para asegurar consistencia.
- (4) Las salidas de la revisión del diseño preliminar y del diseño crítica deben ser ejemplo para asegurar que todos los resultados han sido incorporados y terminados.

El equipo de Auditoria debe comparar el código con la documentación de los requerimientos del software como el estado en el documento de especificaciones actual, para determinar que el código contiene todo (y únicamente) los requerimientos documentados.

C1.4. Esta Auditoria está relacionado no solamente con funcionalidad, pero también con funcionamiento como sigue: Acompañamiento en las pruebas con documentación de pruebas apropiada y datos validados debe establecer suficientemente la funcionalidad de los ítem configurados. Para los parámetros de funcionalidad que pueden no estar verificados completamente durante las pruebas, simulaciones u otros análisis debe ser acompañados para establecer confiabilidad que el ítem de configuración encuentra criterio de funcionalidad cuando entra en operación como indica en sus especificaciones.

El informe de FA debe incluir una evaluación de cada elemento del software examinado, y proporcionar esta evaluación dentro del marco estándar del informe de Auditoria. La evaluación debe tomar la forma de:

- (1) Aprobación. Si los elementos del software han sido observados como teniendo todos las funciones requeridas y mostrar evidencia cumplimiento de los estándares mínimos de funcionalidad.

- (2) Aprobación de contingencia. Si la evaluación de los elementos del software muestra como resultados la aprobación completa de las especificaciones, acción correctiva bien definida (observable)
- (3) Desaprobada. Si los elementos del software fueron observados para ser seriamente inadecuados.

## **C2. La Auditoria Física (PA)**

El objetivo de la Auditoria física es proveer una evaluación independiente de la configuración de los ítem de un producto software para confirmar que componentes en la versión como construir el mapa para sus especificaciones. Específicamente esta Auditoria está hecha para verificar que el software y su documentación son internamente consistentes y están listas para la entrega. Ver norma ANSI/IEEE 730-1984 (A1).

C2.1. Central para la PA es la identificación de los elementos software a ser auditados por

- (1) Nomenclatura
- (2) Número de identificación de la especificación
- (3) Número del ítem de configuración.

C2.2 Además, los materiales proporcionados por la organización auditada incluyen:

- (1) Lista actual de la configuración de los ítem específicos deliberados.
- (2) Requerimientos y arquitectura y especificaciones del diseño detallado
- (3) Lista de cambios aprobados de la configuración con los cambios de las salidas identificados
- (4) Documentación de la prueba de aceptación (por ejemplo, planes, especificaciones, procedimientos y reportes) para el ítem de la configuración.
- (5) Apropiada documentación entregada al cliente (por ejemplo, manuales operativos, y manuales de mantenimiento).
- (6) Nomenclatura apropiada, marcas variables
- (7) Documento con la descripción de la versión del software.
- (8) Reportes de FA

Además, la organización auditada debe identificar cualquier diferencia entre la configuración del sistema en ambiente de desarrollo y la configuración específica del sistema bajo examinación. Por otra parte, ellos deben certificar o de otra manera demostrar que no hay diferencia que degraden las características funcionales o de funcionamiento del sistema.

C2.3 Para proporcionar evidencia del control adecuado del contenido y la consistencia del sistema, el equipo de Auditoría debe como mínimo examinar:

- (1) Los documentos de las especificaciones del sistema para formatos y complementos
- (2) Reportes funcionales (FA) por discrepancias y acciones tomadas
- (3) Descripción del diseño (por lo menos sobre una base ejemplo) para símbolos apropiados etiquetas, referencias, y descripción de datos.
- (4) Descripción de los niveles de abstracción para la consistencia
- (5) Manuales (por ejemplo manual del usuario, manual del programador y manual de diseño) para completar los formatos y la conformidad con la descripción de los datos. Verificación formal o aceptación de los manuales para ser enunciados mientras el resto de las pruebas del sistema se completan para asegurar que el contenido de los procedimental está correcto.
- (6) Los medios de entrega de elementos del software (y controles de los medios) para asegurar transferencia o transmisión apropiada..

El equipo de auditoría debe examinar el sistema para proveer vendedor y desarrolladores de elementos de software, y dejar evidencia que los elementos del software fueron producidos bajo controles de calidad razonables y prudentes.

C2.4 El equipo de Auditoría debe examinar la revisión de ingeniería y cambiar las funciones de control (y asociarla a la documentación) para evidenciar que puede

- (1) Identificar la composición de cualquier configuración en términos de sus unidades subordinadas
- (2) Determinar, dando un identificador de unidad, el elemento padre
- (3) Describir la composición del ítem de configuración o su identificador con respecto a otro ítem de configuración o identificadores
- (4) Identificar las especificaciones relacionadas con un ítem de configuración dado
- (5) Identificar cambios excepcionales una vez se tenga la configuración del ítem
- (6) Identificar cambios y conservar registros de las configuraciones mas recientes

C2.5 El informe del PA debe ser incluido en la evaluación de todos los elementos del software en su totalidad y proporcionar esta información dentro del marco estándar del informe de intervención. La evaluación debe tener

- (1) Aprobación. Cuando el elemento del software del sistema ha sido observado teniendo todos los componentes entregados de acuerdo con su uso compatible

- (2) Aprobación contingente. Cuando la aprobación de los elementos del software del sistema dan lugar a la terminación correcta dada la aprobación de las especificaciones bien definidas.
- (3) Desaprobación. Cuando los elementos del software del sistema han sido observados por ser seriamente inadecuados.

C2.6 El equipo de Auditoria debe examinar la revisión de ingeniería y cambiar las funciones de control (y asociar la documentación) para evidenciar que puede

- (1) Identificar la composición de cualquier configuración en los ítem de sus unidades subordinadas
- (2) Se determina el elemento del padre dando un identificador de la unidad.
- (3) Describir la composición de un ítem de configuración o su identificación con respecto a otros identificadores de ítem de configuración.
- (4) Identificar las especificaciones relacionadas a un ítem de configuración dado.

### **C3. Auditoria en procesos**

Las Auditorias en procesos se mantiene durante las fases de diseño y desarrollo antes de la auditoria funcional para verificar la consistencia del diseño.

El objetivo es verificar la consistencia de los productos como su evolución a través de los procesos de desarrollo determinando sobre una base de muestreo, que

- (1) Las interfases del hardware y software cn consistentes con los requerimientos del diseño en la especificación de requerimientos del software (SRS)
- (2) El código es completamente probado por el SWP para asegurar que los requerimientos funcionales del SWP son satisfechos
- (3) El diseño del producto, como la descripción del diseño del software (SDD) es desarrollada, satisfaciendo los requerimientos funcionales del SRS.
- (4) El código es consistente con el SDD

Sin embargo, las auditorias en procesos pueden ser aplicadas para determinar conformidad con el estándar del proceso aplicable (por ejemplo, revisión del proceso de inspección, sobre un base del muestreo para determinar si las reglas de los procesos de inspección se están siguiendo).

El material de entrada debe ser incluido en la descripción de los procesos formales y una muestra del producto que evidencie la ejecución de los procesos.

Las Auditorias en procesos son únicas para el proceso y producto examinado. La especificación de procedimientos deben establecerse localmente, usando los pasos generales encontrados en la sección 8.

El resultado de todas las Auditorias en procesos debe ser documentado en reportes de Auditorias en procesos, los cuales especifican todas las discrepancias encontradas y proveen una opinión total. La opinión debe tomar la forma de aprobación, aprobación contingente o desaprobación. Las desviaciones significativas de conformidad se observan y se debe seguir hasta asegurar su corrección o renuncia. Esto ocurre durante o antes de la auditoria funcional.

#### **C4. La Auditoria de Sistemas de Calidad (QSA)**

El propósito de las auditorias de sistemas de calidad o evaluación de programas de calidad es proporcionar una gravamen independiente de la conformidad de un plan de aseguramiento de calidad del software.

C4.1 Específicamente, el objetivo de QSA es determinar basado en la evidencia (comprobable), que

- (1) La documentación del programa de calidad del software establecida por la organización desarrolladora, direcciona como mínimo los elementos básicos de ANSI/IEEE Std 730-1984 [A1] u otro estándar apropiado.
- (2) La organización desarrolladora de software sigue la documentación del programa de calidad del software.

El plan SQL debe incorporar todos los objetivos con criterio funcional de la organización; estándares y procedimientos internos; requerimientos de los procesos mandados por la ley; el contrato, u otras políticas; y conformar para el ANSI/IEEE 730-1984[A1] u otro estándar apropiado para aseguramiento de calidad del software.

C4.2 Central para QSA es la identificación de la documentación del programa de calidad, y su disponibilidad para el equipo de auditoria.

Las siguientes actividades son requeridas:

- (1) Una conferencia de preauditoria con personal de auditoria de la organización se requiere para familiarizarse con las metas de la auditoria y procedimientos.
- (2) Examinar los documentos del programa de calidad e ítem relacionados para cumplimiento, seguimiento y consistencia.
- (3) Realizar pruebas de cumplimiento selectivo para comprobar objetivamente que los procedimientos SQ y estándares se han implementado correctamente.
- (4) Entrevistas al personal

- (5) Ejecutar auditorias apropiadas en procesos (Por ejemplo, observando inspecciones y sus evidencias documentadas)
- (6) Examinar informes de cualquier auditoria funcional y física reciente de la configuración.

El reporte SQA debe incluir una evaluación de la calidad del programa en su totalidad, y debe proporcionar esta evaluación dentro del informe de auditoria estándar.

La evaluación ( sigue el formato general de la sección 8) deberá tomar la forma de aprobación, aprobación contingente o desaprobación para cada elemento importante del programa identificado en el plan de auditoria.

## **ANEXO B: Caso de aplicación de la Norma IEEE 1028 - 1988**

### **Evaluación de la Descripción del Diseño**

#### **1. Introducción**

Este documento sigue parcialmente el estándar IEEE 1028. Según el estándar este es requerido para incluir la porción de otros requisitos tales como el proceso de revisión gerencial, el proceso de revisión técnica, el proceso de inspección del software, proceso de verificación y procesos de auditoría. El objetivo principal del documento es verificar, probar si el producto resuelve todos los requerimientos necesarios. Los criterios principales que incluyen en éste documento son: traceabilidad, cohesión y acoplamiento

##### **1.1 Propósito del documento**

El propósito del documento es proporcionar la manera de tener acceso a la base de datos ORACLE del PERL usando ORAPERL

##### **1.2 Alcance**

El alcance del producto ha sido claramente definido. El alcance del documento es tener una manera tener acceso a una base de datos ORACLE de PERL usando ORACLE. Este documento evalúa a descripción del diseño parcialmente siguiendo el estándar IEEE-1028

##### **1.3 Definiciones, siglas y abreviaturas**

Hay solamente información mínima proporcionada en el documento. No se consideran definiciones, siglas o abreviaturas a través del documento.

##### **1.4 Referencias**

<http://www.perl.com/CPAN/modules/by-module/DBD/DBD-Oracle-1.12.readme>

<http://www.cs.purdue.edu/homes/cs290w/perlLecs/PerlOracleExample.html>

<http://www.princeton.edu/~storacle/perldb.html>

<http://www.npac.syr.edu/users/gcf/cps616oracleaccess97/foilsepimagedir/038IMAGE.html>

[http://www.csee.umbc.edu/~jklabrou/courses/fall1999\\_691f/week5/suttonDBi nterfaces/slide8.html](http://www.csee.umbc.edu/~jklabrou/courses/fall1999_691f/week5/suttonDBi nterfaces/slide8.html)

<http://www.eecs.umich.edu/dco/services/software/perl/perlman.pdf>

<http://theoryx5.uwinnipeg.ca/CPAN/by-module/DBI.html>

[http://theoryx5.uwinnipeg.ca/mod\\_perl/cpan-search?](http://theoryx5.uwinnipeg.ca/mod_perl/cpan-search?)

## **2 Ambiente de Ejecución y Desarrollo**

Esta sección contiene información sobre el ambiente en el cual el producto ha sido desarrollado y además acerca del ambiente en el cual el producto ha sido ejecutado.

### **2.1 Ambiente de desarrollo**

PERL 5.004 es el ambiente de desarrollo para desarrollar los programas perl los cuales contienen comandos sql. Las tablas y otra información necesaria en Oracle 8i han sido creadas usando SQLplus. Perl y Oracle 8i están funcionando en máquinas Unix/Solaris. Todas las herramientas son accesibles de geneva.engr.ucf.edu bajo el path /usr/local/apps. Únicamente Oracle 8i es accesible a través de oraperl

El path para fijar cuenta oracle: adicionar el siguiente comando al archivo cshrc e instalar el archivo del cshrc del ambiente o copiar el archivo cshrc desde bruce.engr.ucf.edu/wan

```
set PATH = (setenv PATH /usr/local/apps/oracle8i/OraHome1/bin
/usr/local/apps/insure++/bin.solaris $PATH)
setenv ORACLE_HOME /usr/local/apps/oracle8i/OraHome1
setenv ORACLE_SID SEECS
```

### **2.2 Ambiente de ejecución**

Sun Solaris.

## **3 DISEÑO DEL SOFTWARE**

### **3.1 Resumen**

El archivo prueba (test.pl) y cuatro tablas (dbi:Oracle:EXTPROC\_CONNECTION\_DATA, dbi:Oracle:HQ, dbi:Oracle:INST1\_HTTP, dbi:Oracle:SEECs) que vienen con el módulo DBD son usados más bien que crear nuestro propio producto.

### 3.2 Arquitectura

El programa se conecta como 'scott/tiger' (desde la línea de comando, use ORACLE\_SID o TWO\_TASK) (variable del medio ORACLE\_USERID o el defecto – recomiendan [name/passwd@dbname](#)) (ORACLE\_SID='SEECs', TWO\_TASK=), Probando internal fila traída de arriba, trayendo lista de tablas, probando ora\_do con comando inofensivo non-select (fije transacción de sólo lectura), probando repetitivamente conectar/abrir/cerrar/desconectar, probando interacción explícita cerrar/salir e implícita DESTRUYE, Probando fila cache.

La siguiente es la salida prevista:

```
Campos: 6
Nombres:      'NUM_T'      'DATE_T'      'CHAR_T'      'ROWID_T'
'RAW_T' 'NULL_T'
Longitudesws: 134 76   31   21   3   1
OraTypes: 2  12  1   104  23  1
SQLTypes: 8  9   12  -9104 -2  12
Scale:  0  0   0   0   0   0
Precision: 126 75  30  20  2   0
Nullable: 1  1   1   1   1   1
Ancho de la fila: 46
```

### 3.3 Diseño de Constraints

No hay

### 3.4 Objetos de datos globales

No hay objetos de datos encontrados pero tenemos que colocar las variables del ambiente para acceder oracle y oraperl

### 3.5 Manejo de Errores

El manejo de excepciones de Perl está incluido en algunos ejemplos:

```
# get error codes
require('oraperl.ph');
# Arrange to use curses functions if they're available.
# (This is just showing off)
if (defined (&initscr) && &initscr ())
{
    _eval <<'____END_OF_CURSES_STUFF';
```

```

$curses = 1;
# functions used by the list function
sub before
{
  &erase();
  &standout();
  &addstr("Num Name      Ext\n\n");
  &standend();
  $lineno = 1;
}
sub during
{
  &addstr(sprintf("%2d %-15s%3s\n", $lineno++, $name, $ext));
}
sub after
{
  &standout();
  &move($LINES - 1, 0);
  &addstr("Press RETURN to continue.");
  &standend();
  &refresh();
  &getstr($dummy);
  &move($LINES - 1, 0);
  &addstr("      ");
  &move($LINES - 1, 0);
  &refresh();
}
__END_OF_CURSES_STUFF

```

### 3.6 Lenguaje de Desarrollo

El lenguaje de desarrollo es PERL 5.004. el ambiente de desarrollo para crear y modificar la base de datos de Oracle<sup>®</sup> es SQLplus

### 3.7 Precondiciones y Poscondiciones

Antes de ejecutar el software habíamos hecho pruebas y el sistema reunió con todos los requerimientos en la documentación.

El producto implica tener acceso a la base de datos mientras se ejecuta. Si cualquier función o módulo establece una conexión a la base de datos entonces debe cerrar la conexión antes de salir. Cuando el software termina, todos los recursos deben ser liberados, incluyendo la memoria y recursos DLL

### **3.8 Interfases del Usuario**

La interfaz del usuario es un shell C.

## **4 Requerimientos de traceabilidad**

La definición de requerimientos de traceabilidad es amplia: la traceabilidad de los requisitos es la capacidad de describir y seguir la vida de un requisito, en ambas direcciones, hacia adelante y hacia atrás, sin el contexto de tres compuestos, interrelacionados y capas paralelas: organización / reglas del ambiente, gerencia y desarrollo.

## **5 Cohesión**

Cohesión es una medida de cómo está bien un componente ' caben juntos', un componente debe implementar una sola entidad o función lógica. Cohesión es un atributo del componente del diseño deseable como cuando un cambio tiene que ser realizado, esto es localizado en un único componente cohesivo. Cohesión es una extensión para el cual el módulo tiene una única responsabilidad. En un módulo verdaderamente cohesivo, todas las instrucciones en el módulo pertenecen a realizar una única tarea unificada. En un módulo máximamente cohesivo también tienden a ser juntados lo más libremente posible, esto es la realización de cohesión de altos niveles en su diseño del sistema también ayuda a minimizar acoplamiento. Si un módulo es diseñado para ejecutar una y sólo una función entonces no tiene necesidad de saber sobre los ejecutamientos interiores de otros módulos. El módulo cohesivo sólo necesita tomar los datos para pasarlos, actuar sobre ellos, y pasarlos a la salida encendidos a su estupendo módulo ordinario. Siete tipos de cohesión han sido identificados, en orden del mas al menos funcional, secuencial, comunicacional, procedimental, temporal, lógico y coincidencial.

### **Cohesión Funcional**

El nombre del módulo indica su función

Todas las declaraciones dentro de un módulo se base en una función.

Esta es la mejor cohesión como el módulo ejecuta una función especifica simple.

### **Cohesión Secuencial**

Las instrucciones dentro de un módulo están relacionadas unas con otras a través de un dato de entrada.

La primera instrucción actúa sobre los datos que son pasados dentro del módulo, la segunda instrucción usa la salida de la primera instrucción como entrada y así sucesivamente.

La secuencia de eventos es muy importante

### **Cohesión Comunicacional**

Las actividades son relacionadas unas con otras por los datos que los módulos utilizan.

Cada instrucción actúa sobre los mismos datos de entrada o se trata con los mismos datos de salida

La secuencia no es importante

### **Cohesión Procedimental**

Las instrucciones en un módulo están relacionadas una con otra a través de un flujo de control.

Las instrucciones se agrupan juntas debido a un orden procedimental particular

La secuencia es importante

Las instrucciones están más relacionadas con otros módulos que son el uno al otro.

### **Cohesión Temporal**

Las instrucciones en un módulo están relacionadas una a la otra a través de un flujo de control.

La instrucción está agrupada

Las instrucciones están agrupadas juntas debido a que ocurre casi al mismo tiempo.

Estas instrucciones ejecutan más de una función

Por ejemplo el módulo de iniciación al principio del programa, el cual contiene el paso de iniciación sin afectar otros módulos.

### **Cohesión Lógica**

Las instrucciones están relacionadas unas con otras en todo

Las instrucciones están agrupadas juntas para asegurar clases o actividades

Una bandera que es pasada de afuera determinará cuando fijar la instrucción para ser ejecutada

### **Cohesión Coincidental**

Las instrucciones no están relacionadas unas con otras en todo, ellas apenas coinciden en el mismo módulo.

Este es el peor tipo de cohesión

## 6 Acoplador

El acoplador es una medida de aseguramiento de las interconexiones entre sistemas

El acoplador es una extensión para lo cual los módulos dependen uno del otro, el acoplador puede estar formalmente y el acoplador es bueno.

El acoplador es un indicador de inter-dependencia de los módulos. Generalmente, los módulos menos dependientes están uno sobre otro, el software mas estable y mejor diseñado es percibido para ser.

Hay cinco categorías de acopladores.

- 1.) Acoplador de datos
- 2.) Acoplador de estampilla
- 3.) Acoplador de control
- 4.) Acoplador común
- 5.) Acoplador de contenido

## 7 Conclusiones

Este proyecto ha tenido mucho valor. Este muestra al evaluador los pasos y los detalles que tienen que considerar para determinar una evaluación profesional de una herramienta de software. Los estándares de la IEEE usados durante este proyecto son la guía principal hacia el desarrollo de la evaluación.

Oraperl es una herramienta poderosa que puede ser usada para acceder Oracle remotamente o localmente.

Oraperl is a powerful tool that can be used to access Oracle either remotely or locally. No había mucho tiempo para evaluar todos los ejemplos y características de Oraperl, las partes que fueron evaluadas, mostraron que la herramienta funciona bien, y que la documentación es clara y conforme a los estándares IEEE

Algunos problemas fueron encontrados a los largo del proyecto. Al comienzo algunos problemas administrativos para conseguir registrar este curso, los problemas con el engr.engr.ucf.edu debido a la inexistencia de Oraperl ni a un perl en bruce.engr.ucf.edu, finalmente que algunos archivos principales de Oracle han sido instalados en la misma máquina donde oraperl es instalado, si uno quiere acceder Oracle remotamente. Debido a la presión del tiempo, el proyecto se hace solamente en Ginebra en vez de acceder remotamente Oracle.

No hay mucha información, el entrenamiento y maestría de la herramienta Oraperl debe ser habilidad para indicar posibles extensiones de ésta herramienta.

**ANEXO C: Plantilla Norma IEEE 1028 - 1988**

PLANTILLA NORMA IEEE - 1028-1988 REVISIÓN Y AUDITORIA DE SOFTWARE		FASE I: REVISIÓN GERENCIAL
I. EVALUACION DE LA PLANIFICACION, RECURSOS Y PROGRAMACION		
<b>1.1 OBJETIVOS</b>		
DETERMINA		S / N
1.1.1	<b>METAS DE LA REVISIÓN GERENCIAL</b>	
	Hacer actividades progresivas de acuerdo al plan basado en evaluación del estado	
	Cambiar la dirección del proyecto o identificar necesidad de planificación	
	Mantener control global del proyecto por asignación adecuada de recursos	
<b>1.2 RESPONSABILIDADES</b>		
DETERMINA		S / N
1.2.1	<b>JEFE DEL PROYECTO</b>	
	Asegurar que la revisión sea ordenada	
	Asegurar el reporte administrativo de la revisión	
		S / N
1.2.2	<b>MIEMBROS DEL EQUIPO DE REVISIÓN</b>	
	Estar preparado para las reuniones	
	Asegurar que la reunión sea objetiva	
<b>1.3 ENTRADAS</b>		
DEFINE:		S / N
1.3.1	<b>PRODUCTOS A QUE SE APLICA EL PROCESO E INFORMACION SOPORTADA</b>	
	Objetivos para la revisión gerencial	
	Lista de usos a ser direccionados	
	Datos de horario y costos del proyecto	
	Reportes de otras revisiones o auditorias terminadas	
	Informe de recursos asignados al proyecto	

	Datos sobre elementos de Software completos o en progreso	
	Material adicional suministrado por el director	
<b>1.4 CRITERIOS DE ENTRADA</b>		
		S / N
<b>1.4.1</b>	<b>AUTORIZACIÓN</b>	
	Refiere:	
	Revisiones requeridas por un plan específico	
	Revisiones formalmente anunciadas de acuerdo a procedimientos locales	S / N
<b>1.4.2</b>	<b>COMIENZO DE LA REVISIÓN GERENCIAL</b>	
	El jefe de Revisión Confirma:	
	Definición de Objetivos	
	Compleitud de los elementos de software y documentos	
<b>1.5. PROCEDIMIENTOS</b>		
		S / N
<b>1.5.1</b>	<b>PLANIFICACION</b>	
	El jefe de Revisión y el Director del Proyecto identifican:	
	El equipo de Revisión	
	Horario y Lugar de la Revisión	
	Distribuyen:	
	Material de entrada a participantes para preparación	
		S / N
<b>1.5.2</b>	<b>APRECIACION GLOBAL</b>	
	Persona calificada conduce:	
	Sesión apreciación global del proyecto	
		S / N
<b>1.5.3</b>	<b>PREPARACIÓN</b>	
	Asegura:	
	Cada persona del equipo estudia el material y lo prepara	
		S / N
<b>1.5.4</b>	<b>REVISIÓN</b>	
	Examina:	
	El estado del proyecto	
	Determina:	

	Cumple con el estado esperado de acuerdo al plan predefinido	
	Cubre factores externos e internos no considerados originalmente en el plan	
Genera:		
	Lista de emisiones y recomendaciones para niveles administrativos mas altos	
	Lista de emisiones y recomendaciones a otras organizaciones que afecten el proyecto	
Recomienda:		
	Cursos de acción	
	Autorización para revisiones o auditorias adicionales	
		S / N
<b>1.5.5 REFORMAS</b>		
<b>1.6.CRITERIOS DE SALIDA</b>		
	ESPECIFICA:	S / N
<b>1.6.1 CONDICIONES QUE DEBEN SER SATISFECHAS ANTES DE TERMINAR</b>		
Obtiene:		
	Reporte de revisión gerencial	
Verifica:		
	Cumplimiento de los objetivos de la revisión	
<b>1.7.SALIDA</b>		
	DETERMINA:	S / N
<b>1.7.1. REPORTE DE REVISISON GERENCIAL</b>		
Identifica:		
	El proyecto que se esta revisando	
	Equipo de revisión	
	Entradas de la revisión	
	Objetivos de la revisión	
	Propiedades y estado del ítem en acción	
	Emisiones y recomendaciones para que el proyecto encuentre su pilar	

	Recomendaciones para revisión y auditoria	
<b>1.8.AUDITORIA</b>		
	DESCRIBE:	S / N
<b>1.8.1. EVIDENCIAS PARA DETERMINAR SIGUIENTE FECHA PARA EL PROCESO</b>		
	Analiza:	
	Reporte de revisión gerencial	
PLANTILLA NORMA IEEE - 1028-1988 REVISIÓN Y AUDITORIA DE SOFTWARE		FASE II: REVISIÓN TECNICA
II. EVALUACION Y ADMINISTRACION DE LOS ELEMENTOS SOFTWARE		
<b>2.1 OBJETIVOS</b>		
	DETERMINA	S / N
<b>2.1.1 METAS DE LA REVISIÓN TECNICA</b>		
	Evalúa:	
	El elemento software coincide con sus características técnicas	
	El desarrollo o mantenimiento del elemento software se hace acorde a diseños, normas y pautas aplicables al proyecto	
	Cambios a los elementos software son implementados apropiadamente y afectan solo las áreas identificadas por la especificación del cambio	
<b>2.2 RESPONSABILIDADES</b>		
	DETERMINA	S / N
<b>2.2.1 JEFE DE LA REVISIÓN - LIDER</b>		
	Conducir la revisión de una manera ordenada	
	Responsable de tareas administrativas de la revisión	
	Asegurar la emisión del reporte de la revisión	
S / N		
<b>2.2.2 REGISTRADOR</b>		
	Documentar mantenimiento	
	Documentar decisiones	
	Documentar recomendaciones	

			S / N
<b>2.2.3 MIEMBROS DEL EQUIPO</b>			
	Propia preparación		
	Asegurar que la revisión cumpla sus objetivos		
	Formular recomendaciones para la gerencia		
<b>2.3 ENTRADAS</b>			
DEFINE:			S / N
<b>2.3.1 PRODUCTOS A QUE SE APLICA EL PROCESO E INFORMACION SOPORTADA</b>			
	Objetivos para la revisión técnica		
	Elementos de Software		
	Especificaciones de los Elementos de Software		
	Planes, estándares o pautas de los Elementos Software		
<b>2.4 CRITERIOS DE ENTRADA</b>			
			S / N
<b>2.4.1 AUTORIZACION</b>			
Refiere:			
	Documento de planificación del proyecto		
	Procedimientos locales para revisiones técnicas no catalogadas		
			S / N
<b>2.4.2 COMIENZO DE LA REVISIÓN TECNICA</b>			
El jefe de Revisión Confirma:			
Definición de Objetivos			
Los Elementos Software muestran facilidad para la revisión			
Compleitud de los elementos de software y documentos			
<b>2.5 PROCEDIMIENTOS</b>			
			S / N
<b>2.5.1 PLANIFICACION</b>			
El jefe de Revisión identifica:			
	El equipo de Revisión		
	Horario y Lugar de la Revisión		
Distribuyen:			
	Material de entrada a participantes para preparación		

				S / N
2.5.2	APRECIACION GLOBAL			
	Persona calificada conduce:			
	Sesión apreciación global del proyecto			
				S / N
2.5.3	PREPARACIÓN			
	Asegura:			
	Cada persona del equipo estudia el material y lo prepara			
				S / N
2.5.4	REVISIÓN			
	Examina:			
	Los elementos software			
	Verifica:			
	Elementos Software cumplan con especificaciones y estándares propios			
	Documenta:			
	Productos técnicos, recomendaciones y responsable de ediciones			
	Identifica:			
	Lista de emisiones y recomendaciones a otras organizaciones que afecten el proyecto			
	Genera:			
	Informe de la reunión, lista de deficiencias y recomendaciones			
	Recomienda:			
	Verificación adicional a deficiencias críticas o numerosas			
<b>2.6.CRITERIOS DE SALIDA</b>				
ESPECIFICA:				S / N
2.6.1	CONDICIONES QUE DEBEN SER SATISFECHAS ANTES DE TERMINAR			
	Obtiene:			
	Reporte de revisión técnica			
	Verifica:			
	Cumplimiento de los objetivos de la revisión			
<b>2.7.SALIDA</b>				
ESPECIFICA:				S / N
2.7.1	REPORTE DE REVISION TECNICA			
	Identifica:			
	Miembros del equipo de revisión			
	Elementos de software revisados			

	Entradas específicas para la revisión	
	Lista de deficiencias de los elementos software	
	Lista de administración de emisiones	
	Ítem de acción de pertenencia y estatus	
	Recomendaciones del equipo de revisión para resolver deficiencias	
<b>2.8. AUDITORIA</b>		
	DESCRIBE:	S / N
2.8.1	EVIDENCIAS PARA DETERMINAR SIGUIENTE FECHA PARA EL PROCESO	
	Analiza:	
	Reporte de Revisión Técnica	
	PLANTILLA NORMA IEEE - 1028-1988 REVISIÓN Y AUDITORIA DE SOFTWARE	FASE III: INSPECCIÓN DEL SOFTWARE
	III. DESCUBRIR E IDENTIFICAR LOS DEFECTOS DE LOS ELEMENTOS SOFTWARE	
<b>3.1 OBJETIVOS</b>		
	DETERMINA	S / N
3.1.1	METAS DE LA INSPECCION DE SOFTWARE	
	Verifica:	
	Los elementos software satisfacen las características técnicas	
	Los elementos software conforman normas aplicables	
	Identifica desviación de normas y especificaciones	
	Reúne Datos de administración del software	
<b>3.2 RESPONSABILIDADES</b>		
	DETERMINA	S / N
3.2.1	MODERADOR	
	Planea y dirige la reunión para el proceso de inspección	
	Asegurar la emisión del reporte de inspección	
		S / N
3.2.2	LECTOR	
	Introducir al grupo de inspección a través de los elementos del software en forma lógica y comprensiva	

		S / N
3.2.3	REGISTRADOR	
	Documentar defectos detectados en las reuniones	
	Registrar los datos de inspección requeridos para el proceso de análisis	
3.2.4	INSPECTOR	S / N
	Identificar y describir defectos en los elementos del software	
3.2.5	AUTOR	S / N
	Reunir los elementos de software según criterio de entrada de inspección	
<b>3.3 ENTRADAS</b>		
	DEFINE:	S / N
3.3.1	PRODUCTOS A QUE SE APLICA EL PROCESO E INFORMACION SOPORTADA	
	Elementos de software que satisfagan criterio de inspección	
	Especificación del elemento software aceptado	
	Lista de Inspección	
	Planes, estándares o pautas de los Elementos Software contra los cuales están siendo inspeccionados	
	Toda forma necesaria para informar de la inspección	
<b>3.4 CRITERIOS DE ENTRADA</b>		
3.4.1	AUTORIZACION	S / N
	Refiere:	
	Documento de planificación del proyecto donde se planea la inspección	
3.4.2	COMIENZO DE LA INSPECCION	S / N
	Requiere:	
	Disponibilidad de los elementos del software	
	Acatamiento del diseño del proyecto	
	Aprobación del horario SQA o SVVP	
	Revisión Programada	
	Solicitud desde administración funcional, administración del proyecto	

	administración de la calidad del software o ingeniería del software.	
		S / N
<b>3.4.3 CRITERIO DE ENTRADA MINIMOS ANTES DE PLANEAR LA INSPECCION</b>		
Satisfacer:		
	Elementos de Software conforme a las normas de contenido y formato del proyecto	
	Todo hito anterior e identificado en el documento de planificación apropiado	
	Toda documentación de soporte requerida este disponible	
	Todos los ítem anotados en la lista de defectos, para la revisión	
<b>3.5.PROCEDIMIENTOS</b>		
		S / N
<b>3.5.1 PLANIFICACION</b>		
Autor :		
	Ensambla los materiales en el paquete de inspección para el moderador	
Moderador:		
	Asegura que los materiales satisfagan el criterio de entrada de inspección	
	Seleccionar el grupo de inspección y asignar sus papeles	
	Programar las reuniones de inspección	
	Distribuir el material para la inspección	
		S / N
<b>3.5.2 APRECIACION GLOBAL</b>		
Moderador conduce y autor presenta:		
	Presentación global, a los inspectores, de los elementos de software inspeccionados	
		S / N
<b>3.5.3 PREPARACION</b>		
Asegura:		
	Todos los inspectores conocen los elementos software	
Utiliza:		
	Listas de Inspección	
	Información de la apreciación global	
	Especificación del software	
		S / N
<b>3.5.4 REVISIÓN</b>		
CUMPLE:		

3.5.4.1 INTRODUCCIÓN		
Describe:		
	Condiciones del moderador	
	Participantes y sus roles	
	Propósito de la Inspección	
3.5.4.2 ESTABLECER PREPARATIVOS		
Verificar:		
	Por el moderador el total de registros y tiempo de preparación individual de los inspectores	
Definir:		
	La inspección de acuerdo a la verificación anterior o reprogramar la reunión.	
3.5.4.3 REVISAR LA LISTA DE INSPECCION		
Asegura:		
	Que el producto ha sido adecuadamente estudiado	
3.5.4.4 LEER ELEMENTOS DE SOFTWARE Y REGISTRAR DEFECTOS:		
Examina:		
	El Software Objetivamente	
Produce:		
	Lista de defectos (localización, descripción y clasificación)	
3.5.4.5 REVISAR LA LISTA DE DEFECTOS		
Verificar:		
	La completitud y exactitud de la lista de defectos	
3.5.4.6 TOMAR DECISION DE SALIDA		
Determina:		
	Material reúne el criterio de éxito de inspección	
Prescribe:		
	Verificación de trabajo apropiado	
Identifica:		
	Disposición de los elementos de software según clasificación	
		S / N
3.5.5 RETRABAJO		
<b>3.6.CRITERIOS DE SALIDA</b>		
ESPECIFICA:		S / N
3.6.1 CONDICIONES QUE DEBEN SER SATISFECHAS ANTES DE TERMINAR		

Verifica:	
Que todos los defectos encontrados estén resueltos	
Resultados de inspección han sido reportados	
<b>3.7.SALIDA</b>	
OBTIENE:	S / N
<b>3.7.1 LISTA DE DEFECTOS</b>	
Identifica:	
Localización, descripción y categorización del defecto	
S / N	
<b>3.7.2 RESUMEN DEL DEFECTO</b>	
Identifica:	
Total de numero de defectos por cada categoría	
S / N	
<b>3.7.3 REPORTE DE INSPECCION</b>	
Identifica:	
Numero de participantes	
Duración de la reunión de inspección	
Tamaño de los materiales inspeccionados	
Total del tiempo de preparación del equipo de inspección	
Disposición del elemento software	
Estimación del tamaño del retrabajo y la fecha de realización	
<b>3.8.AUDITORIA</b>	
UTILIZA:	S / N
<b>3.8.1 EVIDENCIAS PARA DETERMINAR SIGUIENTE FECHA PARA EL PROCESO</b>	
Analiza:	
Procedimientos de inspección documentados	
Reportes de inspección retenidos	
Dato de defecto de la inspección retenida	
PLANTILLA NORMA IEEE - 1028-1988 REVISIÓN Y AUDITORIA DE SOFTWARE	FASE IV: REVISIÓN

IV. PERFECCIONAR LOS ELEMENTOS SOFTWARE	
<b>4.1 OBJETIVOS</b>	
DETERMINA	S / N
<b>4.1.1. METAS DE LA REVISIÓN</b>	
Encontrar defectos, omisiones y contradicciones del elemento software	
Perfeccionar los elementos del software	
Considerar implementaciones alternativas	
Intercambio de variación de estilo y técnicas. Educación de los participantes.	
<b>4.2 RESPONSABILIDADES</b>	
DEFINE:	S / N
<b>4.2.1 MODERARDOR</b>	
Conducir la revisión de una manera ordenada	
Responsable de preparar objetivos para guiar al equipo	
S / N	
<b>4.2.2. REGISTRADOR</b>	
Documentar errores, omisiones, contradicciones	
Documentar sugerencias para improvisación o propuestas alternativas	
S / N	
<b>4.2.3 AUTOR</b>	
Presentar material para examinar elementos de software	
<b>4.3 ENTRADAS</b>	
DEFINE:	S / N
<b>4.3.1 PRODUCTOS A QUE SE APLICA EL PROCESO E INFORMACION SOPORTADA</b>	
Objetivos para la revisión	
Elementos de Software bajo reexaminación	
Estándares que estén en vigencia para el desarrollo del elemento software	
Especificaciones de los elementos software bajo revisión	
<b>4.4 CRITERIOS DE ENTRADA</b>	
S / N	
<b>4.4.1 AUTORIZACIÓN</b>	

Refiere:		
	En documento de planificación del proyecto	
	Procedimientos locales para revisiones	
		S / N
<b>4.4.2 COMIENZO DE LA REVISIÓN</b>		
Requiere:		
	El autor indica que esta listo y preparado para presentar los materiales	
	El moderador esta preparado con estándares locales y el elemento software esta completo para conducir la revisión	
<b>4.5 PROCEDIMIENTOS</b>		
		S / N
<b>4.5.1 PLANIFICACION</b>		
El moderador identifica:		
	El equipo de Revisión	
	Horario y Lugar de la Revisión	
Distribuyen:		
	Material de entrada a participantes para preparación	
		S / N
<b>4.5.2 APRECIACION GLOBAL</b>		
Persona calificada conduce:		
	Sesión apreciación global del proyecto	
		S / N
<b>4.5.3 PREPARACION</b>		
Asegura:		
	Cada persona del equipo estudia el material y lo prepara	
Produce:		
	Lista de inquietudes y eventos para discutir	
		S / N
<b>4.5.4 REVISIÓN</b>		
	Presentación de los elementos del software bajo reexaminación	
	Repasa elementos software para hacer preguntas o crear discusiones	
Genera:		
	Reporte de revisión detallando descubrimientos de revisión	
<b>4.6 CRITERIOS DE SALIDA</b>		

ESPECIFICA:		S / N
<b>4.6.1 CONDICIONES QUE DEBEN SER SATISFECHAS ANTES DE TERMINAR</b>		
Obtiene:		
Informe de revisión		
Verifica:		
El total de elementos software han sido repasados en detalle		
Todas las deficiencias, omisiones y sugerencias han sido anotadas		
<b>4.7 .SALIDA</b>		
ESPECIFICA:		S / N
<b>4.7.1 REPORTE DE REVISIÓN</b>		
Identifica:		
Equipo de revisión		
Elementos de Software examinados		
Objetivos seguidos durante la reunión de revisión		
Lista de deficiencias notadas, omisiones, contradicciones y sugerencias		
Recomendación del equipo de revisión		
<b>4.8 AUDITORIA</b>		
DESCRIBE:		S / N
<b>4.8.1 EVIDENCIAS PARA DETERMINAR SIGUIENTE FECHA PARA EL PROCESO</b>		
Analiza:		
Reporte de Revisión		
PLANTILLA NORMA IEEE - 1028-1988 REVISIÓN Y AUDITORIA DE SOFTWARE		FASE V: AUDITORIA
V. AUDITORIA DEL SOFTWARE		
<b>5.1 OBJETIVOS</b>		
DETERMINA		S / N
<b>5.1.1 METAS DE LA AUDITORIA</b>		
Confirmación objetiva de productos y procesos		
Certificar adhesión a estándares, pautas, especificaciones y procedimientos		
<b>5.2 RESPONSABILIDADES</b>		

	DETERMINA	S / N
5.2.1	JEFE DE LA AUDITORIA	
	Organizar y dirigir la auditoria	
	Coordinar la preparación y edición del informe de auditoria	
	Equipo de auditoria este preparado	
	Procedimientos de auditoria son desarrollados y producen informes en concordancia con el alcance de la auditoria	
<b>5.3 ENTRADAS</b>		
	DEFINE:	S / N
5.3.1	PRODUCTOS A QUE SE APLICA EL PROCESO E INFORMACION SOPORTADA	
	Propósito y alcance de la auditoria	
	Compendiar requerimientos, planes, especificaciones, procedimientos, pautas y normas	
	Elementos y procesos de software a ser auditados	
	Antecedentes respecto a responsable de la organización de los productos y procesos auditados	
<b>5.4 CRITERIOS DE ENTRADA</b>		
5.4.1.	AUTORIZACION	S / N
	Requiere:	
	Un proyecto pilar especial ha sido rechazado	
	Partes externas solicitan una auditoria	
	Elemento de organización local solicita una auditoria	
<b>5.5. PROCEDIMIENTOS</b>		
5.5.1	PLANIFICACION	S / N
	Obtiene:	
	Plan de Auditoria	
	Identifica:	
	Procesos del proyecto a ser examinado y el tiempo de ajuste	
	Software requerido a ser examinado y su disponibilidad	

	Mascara de los reportes				
	Distribución de reportes				
	Flujo de actividades requeridas				
	Requerimientos, actividades, elementos y procedimientos para encontrar el objetivo de la auditoria				
	Criterio objetivo de auditoria				
	Lista de chequeo y procedimientos de la auditoria				
	Datos del personal de auditoria: numero, habilidades, experiencia y responsabilidades				
	Organizaciones involucradas en la auditoria				
	Fecha, tiempo, lugar, agenda y publico de la apreciación global				
					S / N
<b>5.5.2 APRECIACION GLOBAL</b>					
	Jefe de auditoria informa:				
	Acuerdos existentes				
	Productos y procesos que intervienen				
	Criterio de auditoria objetivo				
	Preparación para el reporte de auditoria				
	Detalles del plan de auditoria				
					S / N
<b>5.5.3 PREPARACIÓN</b>					
<b>5.5.3.1 Equipo de auditoria:</b>					
	Cumple:				
	Entender la organización				
	Entender los productos y procesos				
	Entender el criterio de auditoria objetiva				
	Preparación para el reporte de auditoria				
	Detalles del plan de auditoria				
<b>5.5.3.2 El jefe del equipo:</b>					
	Cumple:				
	Orientación e instrucción del equipo como se requiera				

	Facilitar las entrevistas de la auditoria	
	Hacer arreglos par materiales, documentos y herramientas requeridas para los procedimientos de auditoria	
	Elementos del software para la auditoria	
	Programar entrevistas	
		S / N
<b>5.4.4 REVISIÓN</b>		
Debe:		
	Revisar procedimientos e instrucciones	
	Examinar la estructura breakdown del trabajo	
	Examinar evidencia de implementación y controles balanceados	
	Examinar documentos de los elementos	
	Probar los elementos	
		S / N
<b>5.5.5 INFORMES</b>		
<b>5.6. CRITERIOS DE SALIDA</b>		
ESPECIFICA:		S / N
<b>5.6.1 CONDICIONES QUE DEBEN SER SATISFECHAS ANTES DE TERMINAR</b>		
Obtiene:		
	Informe de auditoria	
Verifica:		
	Cada elemento dentro de la mira de la auditoria se ha examinado	
	Hallazgos han sido presentado al auditor de la organización	
	El informe de recomendaciones ha sido preparado y sometido a los designados en el plan de la auditoria	
<b>5.7. SALIDA</b>		
DETERMINA:		S / N
<b>5.7.1 INFORME DE AUDITORIA</b>		

Contiene		
	Identificación de la auditoria, titulo, auditor, organización auditada, fecha	
	Alcance	
	Conclusiones	
	Sinopsis	
	Flujo	
<b>5.8.AUDITORIA</b>		
DESCRIBE:		S / N
5.8.1	EVIDENCIAS PARA DETERMINAR SIGUIENTE FECHA PARA EL PROCESO	
Incluye:		
	Programas de trabajo, listas de chequeo	
	Equipo de trabajo	
	Notas de interventoria	
	Pruebas de evidencia completas	
	Copia de ítem examinados	
	Memorando del flujo	
	Informes del proyecto con respuesta del auditor de la organización	

**ANEXO D: Norma IEEE 1028 - 1988**

# IEEE Standard for Software Reviews and Audits

*Circuits and Devices*

*Communications Technology*

## **Computer**

Sponsored by the  
Software Engineering  
Technical Committee of the  
IEEE Computer Society

*Electromagnetics and  
Radiation*

*Energy and Power*

*Industrial Applications*

*Signals and  
Applications*

*Standards  
Coordinating  
Committees*

IEEE Std 1028-1988



Published by the Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017, USA.

June 30, 1989

SH12625



# **IEEE Standard for Software Reviews and Audits**

**Corrected Edition**  
*June 30, 1989*

Corrections have been made on pages 5, 8, 10, 11, 12, 25, 26, 27, 29, and 35. A black bar has been added opposite each correction to aid in identifying the changes made to this printing.

Sponsor

**Software Engineering Technical Committee of the  
IEEE Computer Society**

Approved June 9, 1988

**IEEE Standards Board**

© Copyright 1989 by

**The Institute of Electrical and Electronics Engineers, Inc**  
**345 East 47th Street, New York, NY 10017 USA**

*No part of this publication may be reproduced in any form,  
in an electronic retrieval system or otherwise,  
without the prior written permission of the publisher.*

**IEEE Standards** documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE which have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old, and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board  
345 East 47th Street  
New York, NY 10017  
USA

<p>IEEE Standards documents are adopted by the Institute of Electrical and Electronics Engineers without regard to whether their adoption may involve patents on articles, materials, or processes. Such adoption does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards documents.</p>
--

## Foreword

(This Foreword is not a part of IEEE Std 1028-1988, IEEE Standard for Software Reviews and Audits.)

This Foreword provides the user with the rationale and background of the review and audit procedures outlined in this standard, and their relation to other IEEE standards.

*Purpose.* This standard defines the review and audit processes applicable to critical and noncritical software, and the specific procedures required for the execution of the reviews and audits described. Also, this standard supports ANSI/IEEE Std 730-1984 by defining review and audit procedures.

This standard does not establish measures or mechanisms for judging individual contributions to product development. Conformance to this standard may be claimed where reviews and audits are performed using procedures defined by this standard.

Additional documents that should be referenced for specific application of this standard are as follows:

- (1) ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.
- (2) ANSI/IEEE Std 730-1984, IEEE Standard for Software Quality Assurance Plans.
- (3) ANSI/IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans.
- (4) ANSI/IEEE Std 983-1986, IEEE Guide for Software Quality Assurance.
- (5) ANSI/IEEE Std 1012-1986 IEEE Standard for Software Verification and Validation Plans.

These standards address specific applications to this standard. Although only IEEE standards are referenced in this standard, this standard can be applied in concert with local standards requiring reviews and audits.

Moreover, to support the use of this standard by ensuring uniform classification and codes, use of IEEE Std 1044-1988, IEEE Standard Classification for Software Errors, Faults, and Failures, is suggested.

*General Application Intent.* This standard applies to all phases of typical software life cycles and provides a standard against which review and audit plans can be prepared and assessed. Maximum benefit can be derived from this standard by planning for its application early in the project life cycle.

This standard can be used with different life cycles provided its intent is followed. ANSI/IEEE Std 730-1984 identifies specific (and minimum) review and audit applications for critical software, while this standard defines the specific processes available to meet these application needs. Planning for reviews and audits, as part of an integrated verification and validation approach, can be aided by following ANSI/IEEE Std 1012-1986.

This standard for software reviews and audits was written in consideration of both the software and its system operating environment. It can be used where software is the total system entity or where it is part of a larger system. Care should be taken to integrate software review and audit activities into any total system development planning; they should exist in concert with hardware system reviews and audits to the benefits of the entire system. To broaden the application possibilities of this standard, simply replace *software element* with *system element* when reading.

This standard was written with the assumption that organizations are void of corruption. This may become an issue where, for example, during the course of an audit, theft or embezzlement is discovered. In such a case, although the audit may proceed, the applicability of this standard is removed.

*Appendixes.* Although not considered part of this standard, appendixes are provided to assist the reader in understanding:

- (1) Specific process applications required by ANSI/IEEE Std 730-1984.
- (2) How processes defined in this standard might be applied to meet those application needs.

*Audience.* This standard was written for those who are responsible for defining, planning, implementing, or supporting software reviews and audits.

*Conformance.* Conformance to this standard can be claimed when review and audit processes, wherever performed, are performed as defined in this standard.

The following persons were on the balloting committee that approved this document for submission to the IEEE Standards Board.

A. Frank Ackerman	Jean A. Gilmore	Thomas Steven Radi
Richard L. Aurbach	Shirley A. Gloss-Soler	Salim Ramji
Motoei Azuma	J. G. Glynn	Jean-Claude Rault
H. Jack Barnard	Andrej Grebenc	Meir Razy
James B. Behm	Benjamin W. Green	Donald J. Reifer
Leo Beltracchi	Victor M. Guarnera	R. San Roman
Michael A. Blackledge	Lawrence M. Gunther	John C. Rowe
Gilles R. Bracon	David A. Gustafson	Margaret Rumley
Kathleen L. Briggs	Clark Hay	Julio Gonzalez Sanz
A. Winsor Brown	David A. Hill	Stephen R. Schach
William Bryan	William A. Hoberg	Hans Schaefer
F. Buckley	Charles P. Hollocker	Norman Schneidewind
Lorie J. Call	John W. Horch	Wolf A. Schnoege
Ronald L. Cariola	Laurel V. Kaleda	Robert G. Schueppert
Harry Carl	Harry Kalmbach	David J. Schultz
John Center	Myron S. Karasik	Gregory D. Schumacher
T. S. Chow	Timothy C. Kasse	Leonard W. Seagren
J. K. Chung	George A. Klammer	Tony Sgarlatti
Won L. Chung	George Konstantinow	Isaac-Shahab Shadman
Antonio Cicu	Joseph A. Krpinsh	Gerald P. Share
Corey Clinger	Joan Kundig	Robert W. Shillato
Peter Coach	Tom Kurihara	David M. Siefert
Francois Coallier	Lak-Ming Lam	William G. Singer
Carolyn Collins	John B. Lane	Jacob Slonim
Chris Cooke	Robert A. Lane	G. Wayne Staley
Rudolph P. Corbett	Gregorie N. Larsen	Nick Stewart
Stewart Crawford	F. C. Lim	W. G. Sutcliffe
Taz Daughtrey	Bertil Lindbert	Richard H. Thayer
Peter A. Denny	Ben Livson	Bob Thibodeau
Fred M. Discenzo	Austin J. Maher	Paul U. Thompson
David A. Dobratz	Stanley A. Marash	George D. Tice
Michael P. Dotson	Paulo Cesar Marcondes	Terrence L. Tillmanns
David C. Doty	Philip C. Marriott	Valentin W. Tirman
Einar Dragstedt	Nicholas L. Marselos	L. F. Tracey
William P. Dupras	Roger J. Martin	Glenn R. Trebble
Michael Dutton	Paul A. Mauro	Henry J. Trochesset
Mary L. Eads	J. A. McCall	C. L. Troyanowski
Robert G. Ebenau	Russell McDowell	William Stephen Turner, III
L. G. Egan	Stanley E. McQueen	Robert B. Urling
W. D. Ehrenberger	Manijeh Moghis	David Usechak
Walter Ellis	Charles S. Mooney	Tom Vollman
Caroline L. Evans	Gary D. Moorhead	Dolores R. Wallace
David W. Favor	D. D. Morton	John P. Walter
John Fendrich	G. T. Morun	John W. Walz
Glenn S. Fields	Hirouobu Nagano	Charles J. Wertz
Violet Foldes	Geraldine Neidhart	G. Allan Whittaker
Thom Foote-Lennox	Dennis E. Nickle	Patrick J. Wilson
Joel Forman	E. J. Oertel	David L. Winningham
Julian Forster	Wilma Osborne	W. M. Wong
Richrad C. Fries	M. T. Perkins	Dennis L. Wood
Ismael Fuentes	William E. Perry	Marian A. Wysocki
F. Karl Gardner	John Petraglia	Natalie C. Yopconka
Leonard B. Gardner	Ronald T. Pfaff	Donald Zeleny
David Gelperin	Donald J. Pfeiffer	Kenneth M. Zemrowski
Anne K. Geraci	I. C. Pyle	Peter F. Zoll

At the time this standard was approved, the Software Reviews and Audits Working Group of the Software Engineering Technical Committee of the IEEE Computer Society had the following members:

**Charles P. Hollocker, Chairman**

**Tim Kasse, Cochairman**

**William A. Hoberg, Secretary**

A. Frank Ackerman  
Jack Barnard  
William Bryan  
Ronald L. Cariola  
Francois Coallier  
Christopher M. Cooke  
Rudy Corbett  
Stewart Crawford  
Vince Dowd  
Robert G. Ebenau  
Michael Fagan

Violet Foldes  
Gordon Grunewald  
Clark Hay  
Peter Hung  
David R. Janson  
Harry Kalmbach  
Myron Karasik  
Adi Kasad  
Robert A. Kessler  
Tom Kurihara  
Philip C. Marriot

Stan McQueen  
Belden Menkus  
William Perry  
William N. Sabor  
Isaac Shadman  
David J. Schultz  
Richard H. Thayer  
William S. Turner, III  
Marian Wysocki  
Natalie C. Yopconka

When the IEEE Standards Board approved this standard on June 9, 1988, it had the following membership:

**Donald C. Fleckenstein, Chairman**

**Marco W. Migliaro, Vice Chairman**

**Andrew G. Salem, Secretary**

Arthur A. Blaisdell  
Fletcher J. Buckley  
James M. Daly  
Stephen R. Dillon  
Eugene P. Fogarty  
Jay Forster\*  
Thomas L. Hannan  
Kenneth D. Hendrix  
Theodore W. Hissey, Jr

John W. Horch  
Jack M. Kinn  
Frank D. Kirschner  
Frank C. Kitzantides  
Joseph L. Koepfinger\*  
Irving Kolodny  
Edward Lohse  
John E. May, Jr  
Lawrence V. McCall

L. Bruce McClung  
Donald T. Michael\*  
Richard E. Mosher  
L. John Rankine  
Gary S. Robinson  
Frank L. Rose  
Helen M. Wood  
Karl H. Zaininger  
Donald W. Zipse

\*Members Emeriti



## Contents

SECTION	PAGE
1. Scope and References .....	9
1.1 Scope.....	9
1.2 References.....	9
2. Definitions.....	9
3. Introduction .....	10
3.1 Review Process Prerequisites.....	11
3.2 Audit Process Prerequisites.....	11
3.3 Procedural Description Template .....	11
4. The Management Review Process .....	12
4.1 Objective.....	12
4.2 Abstract .....	12
4.3 Special Responsibilities.....	12
4.4 Input.....	12
4.5 Entry Criteria .....	12
4.6 Procedures.....	13
4.7 Exit Criteria.....	13
4.8 Output.....	13
4.9 Auditability.....	13
5. The Technical Review Process.....	13
5.1 Objective.....	13
5.2 Abstract .....	14
5.3 Special Responsibilities.....	14
5.4 Input.....	14
5.5 Entry Criteria .....	14
5.6 Procedures.....	14
5.7 Exit Criteria .....	15
5.8 Output.....	15
5.9 Auditability.....	15
6. The Software Inspection Process.....	15
6.1 Objective.....	15
6.2 Abstract.....	15
6.3 Special Responsibilities .....	15
6.4 Input.....	16
6.5 Entry Criteria.....	16
6.6 Procedures .....	16
6.7 Exit Criteria.....	17
6.8 Output .....	17
6.9 Auditability.....	17
6.10 Data Collection Requirements.....	17
7. The Walkthrough Process .....	18
7.1 Objective.....	18
7.2 Abstract .....	18
7.3 Special Responsibilities.....	18
7.4 Input.....	19

SECTION	PAGE
7.5 Entry Criteria.....	19
7.6 Procedures.....	19
7.7 Exit Criteria.....	19
7.8 Output.....	19
7.9 Auditability.....	20
8. The Audit Process .....	20
8.1 Objective.....	20
8.2 Abstract .....	20
8.3 Special Responsibilities.....	20
8.4 Input.....	20
8.5 Entry Criteria.....	20
8.6 Procedures.....	21
8.7 Exit Criteria.....	22
8.8 Output.....	22
8.9 Auditability.....	23

FIGURE

■	Fig 1 Relation of Some Quality Assurance Processes to Products and Projects.....	10
---	--	----

TABLE

■	Table 1 Some Principal Processes for Achieving Quality Objectives .....	10
---	---	----

APPENDIXES

Appendix A	A Guide to Process Applications for Critical Software.....	25
Appendix B	A Guide to Specific Review Applications .....	27
Appendix C	A Guide to Specific Audit Applications .....	31

APPENDIX TABLES

■	Table A1 Critical Software Examination .....	25
■	Table B1 Review Process Distinctions .....	28

# IEEE Standard for Software Reviews and Audits

## 1. Scope and References

**1.1 Scope.** The purpose of this standard is to provide definitions and uniform requirements for review and audit processes. It does not establish the need to conduct specific reviews or audits; that need is defined by local policy. Where specific reviews and audits are required, standard procedures for their execution must be defined.

This standard provides such definition for review and audit processes that are applicable to products and processes throughout the software life cycle. Each organization shall specify where and when this standard applies and any intended deviations from this standard.

**1.2 References.** This standard shall be used in conjunction with the following publications:

[1] ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.<sup>1</sup>

[2] ANSI/IEEE Std 829-1983, IEEE Standard for Software Test Documentation.

## 2. Definitions

**audit.** An independent evaluation of software products or processes to ascertain compliance to standards, guidelines, specifications, and procedures based on objective criteria that include documents that specify

<sup>1</sup>ANSI/IEEE publications are available from IEEE Service Center, 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331, or from the Sales Department, American National Standards Institute, 1430 Broadway, New York, NY 10018.

- (1) The form or content of the products to be produced
- (2) The process by which the products shall be produced
- (3) How compliance to standards or guidelines shall be measured

**review.** An evaluation of software element(s) or project status to ascertain discrepancies from planned results and to recommend improvement. This evaluation follows a formal process (for example, management review process, technical review process, software inspection process, or walkthrough process).

**software element.** A deliverable or in-process document produced or acquired during software development or maintenance. Specific examples include but are not limited to

- (1) Project planning documents (for example, software development plans, and software verification and validation plans)
- (2) Software requirements and design specifications
- (3) Test effort documentation (ie, as described in ANSI/IEEE Std 829-1983 [2])<sup>2</sup>
- (4) Customer-deliverable documentation
- (5) Program source code
- (6) Representation of software solutions implemented in firmware
- (7) Reports (for example, review, audit, project status) and data (for example, defect detection, test)

For other definitions, including verification and validation, see ANSI/IEEE Std 729-1983 [1].

<sup>2</sup>The numbers in brackets correspond to those of the references in 1.2.

### 3. Introduction

Reviews and audits can be used in support of objectives associated with software quality assurance, project management, and configuration management, or similar control functions, as shown in Table 1. Those quality objectives, valid throughout software development and maintenance, include the need for evaluation, verification, validation, and compliance confirmation. For the sake of completeness, Test is identified in Table 1 as the principal process of validation. Testing, however, is outside the scope of this standard. Processes defined in this standard include

- (1) Management review
- (2) Technical review
- (3) Software inspection
- (4) Walkthrough
- (5) Audit

To support compliance confirmation, auditing activities may include performing reviews and tests to determine product consistency with respect to established baselines (ie, software requirements specifications or design descriptions) or to witness the correctness of implementation.

Moreover, since quality assurance deals with products and processes, Fig 1 provides a view of how various quality assurance pro-

**Table 1**  
**Some Principal Processes for Achieving Quality Objectives**

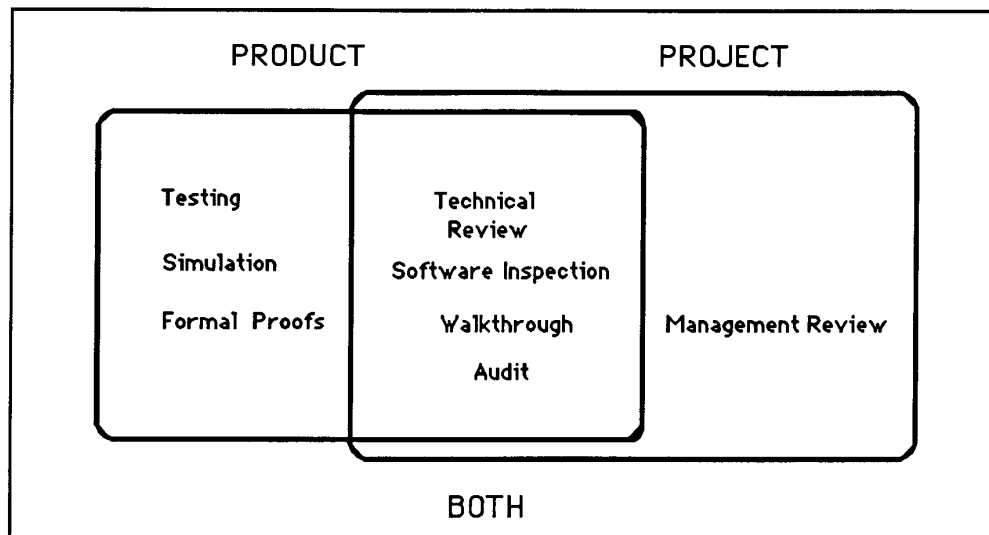
Objectives	Principal Processes Include
Evaluation	Management review, Technical review
Verification	Inspection, Walkthrough
Validation	Test
Compliance, Confirmation	Audit

cesses can be mapped to examine product and project issues.

The examination of project issues (both technical and managerial) occurs at various phases during the project life cycle. The results of such examinations are meant to permit improvement of the methods of ensuring software quality and the ability to meet time and cost constraints. The evaluation of software elements occurs during the production of the element(s) and on its completion. This ensures that the element(s) completely and correctly embodies its baseline specifications.

Any standard process has prerequisite conditions; those conditions necessary, though not in themselves sufficient for the process to reach completion. For reviews and audits, these conditions are identified in 3.1 and 3.2 respectively.

**Fig 1**  
**Relation of Some Quality Assurance Processes to Products and Projects**



Section 3.3 presents the template used to define review and audit processes in the following sections.

**3.1 Review Process Prerequisites.** The objective of reviewing software elements is to evaluate software or project status to identify discrepancies from planned results and to recommend improvement where appropriate.

The following managerial offices are to achieve that objective:

#### 3.1.1 Management

- (1) Responsible for reviews, and the anticipated product rework
- (2) Responsible for performance of reviews and reporting of review results against project milestone events
- (3) Provides the necessary resources of time, personnel, budget, and facilities required to plan, define, execute, and manage the reviews

#### 3.1.2 Development Staff

- (1) The staff has a level of development expertise and product knowledge sufficient to comprehend the software under review.
- (2) Training and orientation in the use of the review processes are provided to the staff.

#### 3.1.3 Development Process Planning

- (1) Standard entry and exit criteria have been defined for development phases.
- (2) The product criteria (such as readability and modularity) are sufficiently defined by development standards.
- (3) The product is partitioned into manageable, reviewable units.
- (4) The software elements to be reviewed and the review process to be employed are identified in the project planning documents (for example, Software Quality Assurance Plan, Software Verification Plan, and Software Project Management Plan).

#### 3.1.4 Review Process Planning

- (1) Each type of review to be employed is fully specified.
- (2) Administrative procedures are defined to enable the review to be initiated, executed, recorded, and acted on.
- (3) The responsibility is assigned for the planning, specification, administration, and maintenance of the review process.

**3.2 Audit Process Prerequisites.** The objective of software auditing is to provide an objective evaluation of products and processes to confirm compliance to standards, guidelines, specifications, and procedures. The following requirements are prerequisite to achieve that objective:

- (1) Objective audit criteria exist (for example, contracts, requirements, plans, specifications, standards) against which software elements and processes can be evaluated.
- (2) Audit personnel are selected to promote team objectivity. They are usually independent of any direct responsibility for the products and processes examined and may be from an external organization.
- (3) Audit personnel are given sufficient authority by appropriate management to perform the audit.

**3.3 Procedural Description Template.** The following descriptions establish the minimum template used to plan, prepare, and execute any review or audit. The review and audit process descriptions in this standard follow this common template:

- (.1) *Objective:* Process goals.
- (.2) *Abstract:* Process overview.
- (.3) *Special Responsibilities:* Roles unique to this specific process.
- (.4) *Input:* Products to which the process is applied, and supporting information.
- (.5) *Entry Criteria:* Conditions that must be satisfied before the process can begin.
- (.6) *Procedures:* Standard steps followed in performing the process.
- (.7) *Exit Criteria:* Conditions that must be satisfied before this process is considered complete.
- (.8) *Output:* The minimum set of deliverables resulting from process completion.
- (.9) *Auditability:* Description of evidence needed to determine at a later date that the process was followed.

Additional template sections have been added to these basic nine sections when needed.

Procedure descriptions in section "(.6)" will be presented as to include, but not be limited to, the following steps:

- (.6.1 Planning      (.6.3 Preparation
- (.6.2 Overview      (.6.4 Examination

Additional procedure subsections have been added where needed.

#### 4. The Management Review Process

**4.1 Objective.** The objective of a management review is to provide recommendations for the following:

- (1) Making activities progress according to plan, based on an evaluation of product development status;
- (2) Changing project direction or to identify the need for alternative planning;
- (3) Maintaining global control of the project through adequate allocation of resources.

The management review process can be applied to new development or to maintenance activities.

**4.2 Abstract.** A management review is a formal evaluation of a project level plan or project status relative to that plan by a designated review team.

During the review meeting the entire review team examines plans or progress against applicable plans, standards, and guidelines, or both. Each problem area identified by the review team is recorded. When critical data and information cannot be supplied, then an additional meeting shall be scheduled to complete the management review process.

**4.3 Special Responsibilities.** The review leader is responsible for the administrative tasks pertaining to the review, for assuring that the review is conducted in an orderly manner, and for issuing the management review report (see 4.8).

The individual(s) responsible for formally reporting the status of the project under review will ensure that the project status and all supporting documentation are available for distribution.

Each member of the review team is responsible for the following:

- (1) Being adequately prepared for the meeting;
- (2) Ensuring that the review meets its objectives.

**4.4 Input.** The minimum input to the management review process is as follows:

- (1) A statement of objectives for the management review
- (2) A list of issues to be addressed
- (3) Current project schedule and cost data
- (4) Reports (for example, managerial review reports, technical review reports, audit reports) from other reviews or audits, or both, already completed
- (5) Reports of resources assigned to the project
- (6) Data on the software elements completed or in progress

Additional applicable reference material can be supplied by project management or requested by the review leader.

#### 4.5 Entry Criteria

**4.5.1 Authorization.** The need for conducting certain management reviews is initially established in the appropriate project planning documents (for example, Software Quality Assurance Plan, Software Development Plan, or Software Verification and Validation Plan). Under these plans, completion of a specific software element (for example, a Planning Document, a Requirements Specification, a Design Document, and Code Specification), or completion of a phase can trigger a management review.

In addition to those management reviews required by a specific plan, other management reviews may be formally announced and held at the request of software quality management, functional management, project management or the customer, according to local procedures.

**4.5.2 Initiating Event.** A management review is conducted when a selected review leader does the following:

- (1) Establishes or confirms a statement of objectives for the meeting;
- (2) Judges that the software element(s) and any other documentation or reports pertinent to the review are sufficiently complete and their status available to support the review objectives.

**4.6 Procedures**

**4.6.1 Planning.** During the planning step, the review leader, in conjunction with project management, does the following:

- (1) Identifies the review team
- (2) Schedules the meeting and identifies a meeting place
- (3) Distributes input materials to the participants, allowing adequate time for preparation

**4.6.2 Overview.** A qualified person from the project under examination shall conduct an overview session for the review team when requested by the review leader. This overview can occur as a part of the examination meeting (see 4.6.4) or as a separate meeting.

**4.6.3 Preparation.** Each person on the review team individually studies the material and prepares for the review meeting. Any required presentations to the review team are prepared.

**4.6.4 Examination.** During the management review the review team holds one or more meetings to do the following:

- (1) Examine project status and determine if it complies with the expected status according to a predefined plan;
- (2) Examine project status and determine if it is overly constrained by external and internal factors not originally considered in the project plan;
- (3) Record all deviations from the expected status accenting risks;
- (4) Generate a list of issues and recommendations to be addressed by higher level management;
- (5) Generate a list of issues and recommendations to be addressed by other responsible individuals, or organizations who affect the project;
- (6) Recommend what course of action should be taken from this point on;
- (7) Recommend authorization for additional reviews or audits;
- (8) Identify other issues that must be addressed.

**4.6.5 Rework.** Any plan adjustments or product rework resulting from the management review is not considered as part of the management review process, except where needed as additional input to complete the examination process.

**4.7 Exit Criteria.** The management review is considered complete when

- (1) All issues identified in the review Statement of Objectives have been addressed.
- (2) The management review report has been issued (see 4.8).

**4.8 Output.** The output from the management review process is a Management Review Report that identifies the following:

- (1) The project being reviewed;
- (2) The review team;
- (3) Inputs to the review;
- (4) Review objectives;
- (5) Action item ownership and status;
- (6) A list of issues and recommendations identified by the review team that must be addressed for the project to meet its milestone;
- (7) Recommendations regarding any further reviews and audits, and a list of additional information and data that must be obtained before they can be executed.

Although this standard sets minimum requirements for report content, it is left to local standards to prescribe any report format requirements.

**4.9 Auditability.** The management review report is an auditable item. It is recommended that this report be traceable to and from the appropriate project planning documents (for example, Software Development Plan, Software Quality Assurance Plan, or other applicable plans, or a combination of these).

**5. The Technical Review Process**

**5.1 Objective.** The objective of a technical review is to evaluate a specific software element(s) and provide management with evidence that

- (1) The software element(s) conform to its specifications;
- (2) The development (or maintenance) of the software element(s) is being done according to plans, standards, and guidelines applicable for the project;
- (3) Changes to the software element(s) are properly implemented, and affect only those system areas identified by the change specification.

Moreover, to further tailor this review process for an individual software element, specific objectives are to be identified in a Statement of Objectives made available before the review meeting. The technical review concept can be applied to new development or to maintenance activities.

**5.2 Abstract.** A technical review is a formal team evaluation of a software element(s). It identifies any discrepancies from specifications and standards or provides recommendations after the examination of alternatives, or both. This examination may require more than one meeting.

**5.3 Special Responsibilities.** Roles for the technical review include:

**5.3.1 Leader.** The review leader is responsible for conducting a specific review. This includes administrative tasks pertaining to the review and ensuring that the review is conducted in an orderly manner. The review leader is also responsible for issuing the review report.

**5.3.2 Recorder.** The recorder is responsible for documenting findings (for example, defects, inconsistencies, omissions, and ambiguities), decisions, and recommendations made by the review team.

**5.3.3 Team Member.** Each team's members is responsible for their own preparation, and for ensuring that the review meets its objectives. Together, they are responsible for formulating recommendations in such a way that management can act on them promptly.

Management is responsible for acting on the review team recommendations in a timely manner.

**5.4 Input.** The minimum input to the technical review process is as follows:

- (1) A statement of objectives for the technical review
- (2) The software element(s) being examined
- (3) Specifications for the software element(s) being examined
- (4) Any plans, standards, or guidelines against which the software element(s) are to be examined

Additional applicable reference material can be made available by the individual(s)

responsible for the software element, when requested by the review leader.

### 5.5 Entry Criteria

**5.5.1 Authorization.** The need for conducting technical reviews of specific software elements is defined by project planning documents.

Unscheduled technical reviews may be conducted at the request of functional management, project management, software quality management, or software engineers, according to local procedures.

**5.5.2 Initiating Event.** A technical review may not be conducted until

- (1) A statement of objectives for the review is established;
- (2) The responsible individual(s) for the software element(s) indicate readiness for review;
- (3) The technical review leader is satisfied that the software element(s) are sufficiently complete for a technical review to be worthwhile

### 5.6 Procedures

**5.6.1 Planning.** A review leader is assigned responsibility for the following planning activities:

- (1) Identify, with appropriate management support, the review team
- (2) Schedule and announce the meeting place
- (3) Distribute input materials to participants, allowing adequate time for their preparation

**5.6.2 Overview.** A technically qualified person shall conduct an overview session for the review team when requested by the review leader. This overview can occur as a part of the review meeting (see 5.6.4) or as a separate meeting.

**5.6.3 Preparation.** During the preparation step, each review team member individually examines the software element(s) and related materials in preparation for the review meeting.

**5.6.4 Examination.** At the review meeting the entire team reviews the software element, evaluating its condition relative to applicable guidelines, specifications and standards, or evaluating alternative problem solutions. Specifically, the review team performs the following tasks:

- (1) Examines the software element(s) under review and verifies that it complies with the specifications and standards to which it must adhere. All deviations from the specifications and standards are recorded.
- (2) Documents technical issues, related recommendations, and the individual responsible for getting the issues resolved.
- (3) Identifies other issues that must be addressed.

After the software element(s) have been reviewed, a report is generated to document the meeting, list deficiencies found in the software element, and describe any recommendations for management.

When deficiencies are sufficiently critical or numerous, the review leader must recommend that an additional review process (that is, management review, technical review, or walkthrough) be applied to the reworked software element(s) after the deficiencies have been resolved. This, at a minimum, covers areas changed to resolve deficiencies.

**5.7 Exit Criteria.** A Technical Review is complete when

- (1) All issues identified in the review Statement of Objectives have been addressed
- (2) The technical review report (see 5.8) has been issued

**5.8 Output.** The output from the technical review process is a technical review report that identifies the following:

- (1) The review team members
- (2) The software element(s) reviewed
- (3) Specific inputs to the review
- (4) A list of unresolved software element(s) deficiencies
- (5) A list of management issues
- (6) Action item ownership and status
- (7) Any recommendations made by the review team on how to dispose of unresolved issues and deficiencies

Although this standard sets minimum requirements for report content, it is left to local standards to prescribe any report format requirements.

**5.9 Auditability.** The technical review report is an auditable item. It is recommended that

this report be traceable to/from the appropriate project planning documents (for example, Software Development Plan, Software Quality Assurance Plan).

## 6. The Software Inspection Process

**6.1 Objective.** The objective of a software inspection is to detect and identify software element defects. This is a rigorous, formal peer examination that does the following:

- (1) Verifies that the software element(s) satisfy its specifications
- (2) Verifies that the software element(s) conform to applicable standards
- (3) Identifies deviation from standards and specifications
- (4) Collects software engineering data (for example, defect and effort data)
- (5) Does not examine alternatives or stylistic issues

**6.2 Abstract.** Software inspections are conducted by peers, and typically comprise three to six participants. The process is led by a moderator impartial to the software element(s) under examination. The moderator is not the author. Defect resolution is mandatory, and rework is formally verified.

Defect data shall be systematically collected and stored in an inspection data base. Minimum requirements for data collection are presented in 6.10. When this data is analyzed, steps can be taken to improve both the product and the development process, and to assess the effectiveness of the inspection process.

**6.3 Special Responsibilities.** All team members, including those with special roles, are inspectors. Special roles required for the inspection process include

**6.3.1 Moderator.** The moderator is the chief planner and meeting manager for the inspection process, and is responsible for issuing the inspection reports (see 6.8). The moderator may also perform the role of recorder.

**6.3.2 Reader.** At the meeting(s), the reader leads the inspection team through the software element(s) in a comprehensive and logical fashion, generally paraphrasing sections of

the work and by reading line-by-line where required by local standards.

**6.3.3 Recorder.** The recorder is charged with documenting defects detected at the meeting(s) and recording inspection data required for process analysis.

**6.3.4 Inspector.** The role of inspector is to identify and describe defects in the software element. Inspectors must be knowledgeable of the inspection process. They are chosen to represent different viewpoints at the meeting (for example, requirements, design, code, test, independent test, project management, quality management). Only those viewpoints pertinent to the inspection of the element are present.

**6.3.5 Author.** The author is responsible for the software element(s) meeting its inspection entry criteria, for contributing to the inspection based on special understanding of the software element, and for performing any rework required to make the software element(s) meet its inspection exit criteria. The author is not allowed to perform any other (that is, moderator, reader, or recorder) role.

**6.4 Input.** The necessary inputs to the inspection consist of the following:

- (1) The software element(s) to be inspected; They must satisfy the inspection entry criteria;
- (2) The approved software element(s) specification;
- (3) The inspection checklist. See Appendix B for sample checklists;
- (4) Any standards and guidelines against which the software element is to be inspected;
- (5) All necessary inspection reporting forms.

For a reinspection, the previous inspection defect list is also required.

## 6.5 Entry Criteria

**6.5.1 Authorization.** Inspections are planned for, and documented in the appropriate project planning documents (for example, the overall project plan, software quality assurance plan, or software verification and validation plan).

**6.5.2 Initiating Event.** The software inspection process can be triggered by the following:

- (1) Software element(s) availability
- (2) Project plan compliance

- (3) SQAP or SVVP schedule compliance
- (4) Scheduled reinspection
- (5) At the request of functional management, project management, software quality management, or software engineering

**6.5.3 Minimum Entry Criteria Before Planning Inspection.** Before planning the inspection, the following minimum entry criteria must be met:

- (1) The software element(s) conform to project standards of content, and format
- (2) All prior milestones are satisfied, as identified in the the appropriate planning documents
- (3) All required supporting documentation is available
- (4) For a reinspection, all items noted on the defect list must be satisfied

## 6.6 Procedures

**6.6.1 Planning.** During the planning step, the author assembles the inspection package materials for the moderator.

The moderator is responsible for assuring that the materials meet the inspection entry criteria. The moderator is also responsible for assuring the selection of the inspection team and the assignment of their inspection meeting roles, for scheduling the inspection meetings, and for the distribution of the inspection materials.

**6.6.2 Overview.** If scheduled, an overview presentation of the software element(s) to be inspected is conducted by the moderator, and the author makes the presentation. This overview is used to educate the other inspectors concerning the software element(s) and may also be attended by other project personnel who could profit from the presentation.

**6.6.3 Preparation.** It is the individual responsibility of all inspectors to become thoroughly familiar with the software element, using the inspection checklist, the information provided in the overview, and the software specification.

**6.6.4 Examination.** The inspection meeting shall follow this agenda:

**6.6.4.1 Introduce Meeting.** To open the meeting, the moderator introduces the participants, and describes their roles. The moderator states the purpose of the inspection and directs the inspectors to focus their efforts toward defect detection, not solution-hunting.

The moderator reminds the inspectors to direct their remarks to the reader and to comment only on the product, not the author. It may also be useful to resolve any special procedural questions raised by the inspectors.

**6.6.4.2 Establish Preparedness.** The moderator asks for individual preparation times and records the total on the inspection report. It is the moderator's responsibility to reschedule the meeting if the inspectors are not adequately prepared.

**6.6.4.3 Review the Inspection Checklist.** The moderator reviews the inspection checklist with the team to ensure that the product has been adequately studied before the inspection meeting.

**6.6.4.4 Read Software Elements and Record Defects.** The reader presents the materials to the inspection team. The inspection team examines the software objectively, and the moderator focuses this part of the meeting on creating the defect list. The recorder enters each defect, location, description, and classification on the defect list. During this time, the author answers any specific questions and contributes to defect detection based on his/her special understanding of the software element.

**6.6.4.5 Review the Defect List.** At the end of the inspection meeting, the moderator must have the defect list reviewed with the team to ensure its completeness and accuracy.

**6.6.4.6 Make Exit Decision.** The purpose of the exit decision is to bring an unambiguous closure to the inspection meeting. The exit decision determines if the materials meet the inspection exit criteria and prescribes any appropriate rework verification. Specifically, the inspection team identifies the software element(s) disposition as one of the following:

- (1) *Accept.* The software element is accepted *as is* or with only minor rework (for example, that would require no further verification)
- (2) *Verify Rework.* The software element is to be accepted after the moderator verifies rework.
- (3) *Reinspect.* Schedule a reinspection to verify rework after revision.

At a minimum, a reinspection examines the product areas changed to resolve defects identified in the last inspection.

**6.6.5 Rework.** During rework, the author revises the materials, addressing all items on

the inspection defect list.

**6.6.6 Follow-Up.** The software inspection process provides follow-up on two levels:

- (1) Verifying rework per inspection disposition
- (2) Reporting inspection data

**6.7 Exit Criteria.** A single exit criterion applied to all inspections is that all of the defects that have been detected are resolved. Each project shall develop its own criteria to meet the needs of its specific products and development environment.

The moderator will ensure that the materials comply with the exit criteria and that the inspection results have been reported before certifying that the inspection is complete.

**6.8 Output.** For each software element(s), the reports produced by the inspection are:

- (1) The defect list, containing the defect location, description, and categorization
- (2) The inspection defect summary summarizing the number of defects identified by each defect category
- (3) The inspection report, containing
  - (a) The number of participants
  - (b) The inspection meeting duration
  - (c) The size of the materials inspected
  - (d) The total preparation time of the inspection team
  - (e) The disposition of the software element
  - (f) An estimate of the rework effort and rework completion date

Information reports are retained for subsequent analysis as prescribed in the appropriate project planning documents. Although this standard sets minimum requirements for report content, it is left to local standards to prescribe any report format requirements.

**6.9 Auditability.** Auditability is provided by the following:

- (1) Documented inspection procedures
- (2) Retained inspection reports
- (3) Retained inspection defect data

**6.10 Data Collection Requirements.** Software inspections provide data for the analysis of the quality of the software elements, the effectiveness of the development procedures, and the

efficiency of the inspection process. To enable these analyses, defects that are identified at an inspection meeting are categorized by defect type, class, and severity.

**6.10.1** The defect type identifies software element(s) attribute, and for example may be identified as affecting the software element's conformance to

- (1) Standards compliance
- (2) Capability
- (3) Procedure
- (4) Interface
- (5) Description

**6.10.2** Defect class characterizes evidence of nonconformance, and for example, may be categorized as

- (1) Missing
- (2) Wrong
- (3) Extra

**6.10.3** Additional defect classes used when inspecting documents could include

- (1) Ambiguous
- (2) Inconsistent

**6.10.4** Defects are then ranked by severity, for example, as

- (1) *Major*. Defects that would result in failure of the software element(s) or an observable departure from specification
- (2) *Minor*. Defects that would affect only the nonfunctional aspects of the software element

In addition to counts of defects by type, class, and severity, retained inspection data shall contain the identification of the software element, the date and time of the inspection, the moderator, the preparation and inspection times, the volume of the materials inspected, and the disposition of the inspected software element. The recording of this information is required in setting local guidance for inspection effort and volume. Common practice would suggest up to 1.5 h preparation, up to 2 h meeting time, and whatever volume might be comfortably handled in that period of time.

The management of inspection data requires capabilities for the storage, entry, access, updating, and summarization and reporting of the categorized defects. The frequency and types of the inspection analysis reports, and their distribution, are left to the users of this standard.

## 7. The Walkthrough Process

**7.1 Objective.** The objective of a walkthrough is to evaluate a software element. Although long associated with code examinations, this process is also applicable to other software elements (for example, architectural design, detailed design, test plans/procedures, and change control procedures). The major objectives are to find defects, omissions, and contradictions; to improve the software element(s); and to consider alternative implementations.

Other important objectives of the walkthrough process include exchange of techniques and style variations, and education of the participants. A walkthrough may point out efficiency and readability problems in the code, modularity problems in design or untestable design specifications.

The walkthrough may be used in either new development or maintenance activities.

**7.2 Abstract.** During the walkthrough meeting, the author makes an overview presentation of the software element(s) under review. This is followed by a general discussion from the participants after which the presenter "*walks through*" the software element in detail. As the walkthrough progresses, errors, suggested changes, and improvements are noted and written. When the walkthrough is finished, the notes are consolidated into one report, which is distributed to the author and other appropriate personnel. A copy of the report is placed in the project file for auditability.

**7.3 Special Responsibilities.** The following roles are established for the walkthrough:

**7.3.1 Moderator.** The walkthrough moderator is responsible for conducting a specific walkthrough, handling the administrative tasks pertaining to the walkthrough, and ensuring that the walkthrough is conducted in an orderly manner. The moderator is also responsible for preparing the statement of objectives to guide the team through the walkthrough.

**7.3.2 Recorder.** The recorder is responsible for writing all comments made during the walkthrough that pertain to errors found,

questions of style, omissions, contradictions, suggestions for improvement, or alternative approaches.

**7.3.3 Author.** The author is the person responsible for the software element(s) being examined, and presents the materials.

Each member of the walkthrough team is responsible for reviewing any input material prior to the walkthrough, and participating during the walkthrough to ensure that it meets its objective.

Roles may be shared among the walkthrough members. The author and walkthrough moderator may be the same person.

**7.4 Input.** The minimum input to the walkthrough process is as follows:

- (1) A statement of objectives for the walkthrough
- (2) The software element under examination
- (3) Standards that are in effect for the development of the software element
- (4) Specifications for the software element(s) under review

#### 7.5 Entry Criteria

**7.5.1 Authorization.** The need for conducting walkthroughs shall be established in the appropriate planning documents, (for example, a software quality assurance plan (SQAP) or a software development plan). In addition, under either of these plans, completion of a specific software element(s) trigger the walkthrough for that element. Additional walkthroughs should be conducted during development of the software element(s) at the request of functional management, quality management, or the author, according to local procedures.

**7.5.2 Initiating Event.** A walkthrough is conducted when

- (1) The author of the software element(s) to be reviewed indicates it is ready, and is prepared to present the materials.
- (2) The moderator is appointed per local standards or practices and has concluded that the software element is sufficiently complete to conduct a walkthrough.

#### 7.6 Procedures

**7.6.1 Planning.** During the planning phase, the walkthrough moderator does the following:

- (1) Identifies the walkthrough team
- (2) Schedules the meeting and selects the meeting place
- (3) Distributes all necessary input materials to the participants, allowing for adequate preparation time

**7.6.2 Overview.** An overview presentation is made by the author as part of the walkthrough meeting. Before that meeting, however, individual preparation is still required.

**7.6.3 Preparation.** During the preparation phase participants shall review the input material that was distributed to them and prepare a list of questions and issues to be discussed during the walkthrough.

**7.6.4 Examination.** During the walkthrough meeting

- (1) The presenter makes an overview presentation of the software element(s) under examination.
- (2) The author walks through the specific software element(s) so that members of the walkthrough team may ask questions or raise issues regarding the software element, or document their concerns, or both.
- (3) The recorder writes comments and decisions for inclusion in the walkthrough report.

At the completion of the walkthrough, the walkthrough team may recommend a follow-up walkthrough. This follow-up would follow the standard for walkthroughs, and would, at a minimum, cover areas changed by the author.

Within a reasonable time after the walkthrough, the walkthrough leader shall issue a walkthrough report detailing the walkthrough findings. Minimum content requirements are stated in 7.8.

**7.7 Exit Criteria.** The walkthrough process is complete when

- (1) The entire software element(s) have been "walked through" in detail
- (2) All deficiencies, omissions, efficiency issues, and suggestions for improvement have been noted
- (3) The walkthrough report has been issued

**7.8 Output.** The output of the walkthrough process is a walkthrough report listing those deficiencies, omissions, efficiency issues, and suggestions for improvement that were recorded during the walkthrough.

The walkthrough report contains the following:

- (1) Identification of the walkthrough team
- (2) Identification of the software element(s) being examined
- (3) The statement of objectives that were to be handled during this walkthrough meeting
- (4) A list of the noted deficiencies, omissions, contradictions, and suggestions for improvement
- (5) Any recommendations made by the walkthrough team on how to dispose of deficiencies and unresolved issues

If follow-up walkthroughs are suggested, this suggestion shall also be mentioned in the report.

Although this standard sets minimum requirements for report content, it is left to local standards to prescribe any report format requirements.

**7.9 Auditability.** The walkthrough report, meeting minutes, and other materials upon which conclusions are based shall be included in project documentation files. The walkthrough moderator files these documents.

## 8. The Audit Process

**8.1 Objective.** The objective of software auditing is to provide an objective compliance confirmation of products and processes to certify adherence to standards, guidelines, specifications, and procedures.

**8.2 Abstract.** Audits are performed in accordance with documented plans and procedures. The audit plan establishes a procedure to conduct the audit and for follow-up action on the audit findings.

In performing the audit, audit personnel evaluate software elements and the processes for producing them against objective audit criteria, such as contracts, requirements, plans, specifications, or procedures, guidelines, and standards.

The results of the audit are documented and are submitted to the management of the audited organization, to the entity initiating

the audit, and to any external organizations identified in the audit plan. The report includes a list of the items in noncompliance or other issues for subsequent review and action. When stipulated by the audit plan, recommendations are reported in addition to the audit results.

**8.3 Special Responsibilities.** It is the responsibility of the audit team leader to organize and direct the audit and to coordinate the preparation and issuance of the audit report. The audit team leader shall ensure that the audit team is prepared to conduct the audit, and that audit procedures are performed and reports (see 8.8) issued in accordance with audit scope.

The entity initiating the audit is responsible for authorizing the audit. Management of the auditing organization assumes responsibility for the audit, and the allocation of the necessary resources to perform the audit.

Those whose products and processes are being audited provide all relevant materials and resources and correct or resolve deficiencies cited by the audit team.

**8.4 Input.** The following inputs are required to ensure the success of the audit:

- (1) The purpose and scope of the audit.
- (2) Objective audit criteria, such as contracts requirements, plans, specifications, procedures, guidelines, and standards.
- (3) The software elements and processes to be audited and any pertinent histories.
- (4) Background information regarding the organization responsible for the products and processes being audited (for example, organization charts).

**8.5 Entry Criteria.** The need for an audit is established by one of the following events:

- (1) A special project milestone has been reached. The audit is initiated per earlier plans (for example, the software quality assurance plan, software development plan).
- (2) External parties (for example, regulatory agencies, or end users) demand an audit at a specific calendar date or project milestone. This may be in fulfillment of a contract requirement, or as a prerequisite to contractual agreement.

- (3) A local organizational element(s) (for example, project management, functional management, systems engineering, internal quality assurance/control) has requested the audit, establishing a clear and specific need.
- (4) A special project milestone, calendar date, or other criterion has been met and, as part of the auditing organization's charter, it is to respond by initiating an audit.

### 8.6 Procedures

**8.6.1 Planning.** The auditing organization shall develop and document an audit plan for each audit. This plan shall, in addition to restating the audit scope, identify the following:

- (1) Project processes to be examined (provided as input) and the time frame for audit team observation.
- (2) Software required to be examined (provided as input) and their availability. Where sampling is used, a statistically valid sampling methodology shall be used to establish selection criteria and sample size.
- (3) Reports shall be identified (that is, results report, and optionally, the recommendations report and their general format defined). Whether recommendations are required or excluded shall be explicitly stated.
- (4) Report distribution
- (5) Required follow-up activities
- (6) Requirements: necessary activities, elements and procedures to meet the scope of the audit
- (7) Objective audit criteria: provides the basis for determining compliance (provided as input)
- (8) Audit procedures and checklists
- (9) Audit personnel: required number, skills, experience, and responsibilities
- (10) Organizations involved in the audit (for example, the organization whose products and processes are being audited)
- (11) Date, time, place, agenda, and intended audience of *overview* session (optional)

The audit team leader shall ensure that the audit team is prepared and includes members with the necessary experience and expertise.

Notification of the audit shall be provided to the involved organizations at a reasonable amount of time before the audit is performed, except for unannounced audits. The notification shall be written and shall include the scope and the identification of processes and products to be audited, and to identify the auditors.

**8.6.2 Overview.** An optional overview meeting with the audited organization is recommended to *kick-off* the examination phase of the audit. The overview meeting, led by the audit team leader, provides the following:

- (1) Overview of existing agreements (for example, audit scope, plan, related contracts);
- (2) Overview of production and processes being audited;
- (3) Overview of the audit process, its objectives, and outputs;
- (4) Expected contributions of the audited organization to the audit process (that is, the number of people to be interviewed, meeting facilities, etc);
- (5) Specific audit schedule.

### 8.6.3 Preparation

**8.6.3.1** The following preparations are required by the audit team:

- (1) Understand the organization: it is essential to identify functions and activities performed by the audited organization, and to identify functional responsibility.
- (2) Understand the products and processes: it is a prerequisite for the team to learn about the products and processes being audited through readings and briefings.
- (3) Understand the objective audit criteria: it is important that the audit team become familiar with the objective audit criteria to be used in the audit.
- (4) Prepare for the audit report: it is important to choose the administrative reporting mechanism that will be used throughout the audit to develop the report that follows the layout identified in the audit plan.
- (5) Detail the audit plan: choose appropriate methods for each step in the audit program.

**8.6.3.2** In addition, the audit team leader shall make the necessary arrangements for

- (1) Team orientation and training as needed

- (2) Facilities for audit interviews
- (3) Materials, documents, and tools required by the audit procedures
- (4) The software elements to be audited (for example, documents, computer files, personnel to be interviewed)
- (5) Scheduling interviews

**8.6.4 Examination.** Elements that have been selected for audit shall be evaluated against the objective audit criteria. Evidence shall be examined to the depth necessary to determine if these elements comply with specified criteria.

The audit shall, as appropriate to its scope:

- (1) Review procedures and instructions
- (2) Examine the work breakdown structures
- (3) Examine evidence of implementation and balanced controls
- (4) Interview personnel to ascertain the status and functioning of the processes and the status of the products
- (5) Examine element documents
- (6) Test the element(s)

**8.6.5 Reporting.** Subsequent to the audit examination, the audit team will issue a draft audit report to the audited organization for review and comments.

Audit team rework of the audit report occurs before formal results reporting. This rework is performed, in concert with the draft report review, resolve any misunderstandings or ambiguities while maintaining objectivity and correctness. It also serves to ensure report usability by adding consistency to the level of report details and by adding any freshly verified information. The recommended practice is to involve representatives of the audited organization in reviewing audit results, as

- (1) An ongoing, periodic activity integrated into the overall examination schedule, or
- (2) As a closure step to the examination period

Involving the audited organization contributes to report quality through interaction and the possible delivery of any further evidence.

The audit team shall conduct a post audit conference to review with audited organization staff the deficiencies, findings, and (if applicable) recommendations. Comments and issues raised by the audited organization must be resolved.

The final audit report shall then be prepared, approved, and issued by the audit team leader to the organizations specified by the audit plan.

**8.7 Exit Criteria.** An audit shall be considered complete when

- (1) Each element(s) within the scope of the audit has been examined;
- (2) Findings have been presented to the audited organization;
- (3) Response to draft findings have been received and evaluated;
- (4) Final findings have been formally presented to the audited organization and initiating entity;
- (5) The audit report has been prepared and submitted to recipients designated in the audit plan;
- (6) The recommendation report, if required by plan, has been prepared and submitted to recipients designated in the audit plan;
- (7) All of the auditing organization's follow-up actions included in the scope (or contract) of the audit have been performed.

**8.8 Output.** As a standard framework for audit reports, the draft and final audit reports shall, as a minimum, contain the following:

- (1) *Audit identification.* Report title, audited organization, auditing organization, date of the audit.
- (2) *Scope.* Scope of the audit, including an enumeration of the standards, specifications, practices, and procedures constituting the objective audit criteria against which the audit was conducted of the software elements and processes audited.
- (3) *Conclusions.* A summary and interpretation of the audit findings including the key items of nonconformance.
- (4) *Synopsis.* A listing of all the audited software elements, processes, and associated findings.
- (5) *Follow-up.* The type and timing of audit follow-up activities.

Additionally, when stipulated by the audit plan, recommendations shall be provided to the audited organization, or the entity that initiated the audit. Recommendations shall be reported separately from results.

**8.9 Auditability.** The materials documenting the audit process must be maintained by the audit organization for a stipulated period of time subsequent to the audit and include the following:

- (1) All work programs, checklists, etc are fully annotated
- (2) Team staffing
- (3) Interview notes, observation notes
- (4) Compliance test evidence
- (5) Copies of examined items, annotated
- (6) Report draft(s) with response by audited organization
- (7) Follow-up memo(s) as necessary



## Appendixes

(These Appendixes are not a part of IEEE Std 1028-1988, IEEE Standard for Software Reviews and Audits, but are included here for information only.)

### Appendix A

#### Guide to Process Applications for Critical Software

##### A1.

ANSI/IEEE Std 730-1984 [A1] has identified a minimum set of review and audit process applications for critical software (for example, where failure could impact safety, or cause large financial or social losses).

- (1) Software requirements review (SRR)
- (2) Preliminary design review (PDR)
- (3) Critical design review (CDR)
- (4) Software verification and validation plan review (SVVPR)
- (5) Functional audit (FA)
- (6) Physical audit (PA)
- (7) In-process audit (IPA)
- (8) Managerial review (MR)

##### A2.

**A2.1** The goal of this Appendix is to provide one possible mapping of specific process application needs identified in that standard, to candidate processes for meeting those needs. Selection of a specific process should be made as local, contractual, or project specific conditions dictate.

For example:

- (1) The software inspection process may be used to meet the software verification and validation plan review application need.
- (2) The management review process may be used to meet the managerial review application need.

**Table A1  
Critical Software Examination**

ANSI/IEEE Std 730-1984 Requirements	SRR	PDR	CDR	SVVPR	FCA	PCA	IPA	MR
Management review			X					X
Technical review	X	X	X	X				
Software inspection	X	X	X	X				
Walkthrough	*	*	*	*				
Audit					X	X	X	**

\* As common practice in many companies, walkthroughs are allowed wherever software inspections might be applied. It is suggested however, that for critical software as presented in ANSI/IEEE Std 730-1984, only more formal/rigorous techniques be used. One significant difference is disallowing the author from presenting the work.

\*\* The audit process may be tailored in its application or administration to meet the need for a managerial review.

A particular process need not be prescribed for all applications. For example, using the same process for all the SRR, PDR, and CDR might not be cost effective or might unwisely be biased toward either technical or business concerns.

**A2.2** Similarly, a combination of distinct processes may be applied to meet the needs of a specific application. An example is providing adequate examination of an SRR for a very large and complex system. SRR concerns could be subdivided into three categories:

- (1) Customer satisfaction with functionality and performance with consideration for technical and business feasibility
- (2) Technical completeness and correctness of requirements
- (3) Fitness-for-use of the specifications document(s) as viewed by development, test, and maintenance organizations

In this example, it may be appropriate to use a management review to examine the first category, a technical review for the second category, and a software inspection to verify fitness-for-use of the document itself.

Table A1 shows possible mappings of review and audit processes to the applications referenced in ANSI/IEEE Std 730-1984 [A1].

### **A3.**

The following references apply to Appendixes A through C:

[A1] ANSI/IEEE Std 730-1984, IEEE Standard for Software Quality Assurance Plans.

[A2] ANSI/IEEE Std 1016-1987, IEEE Recommended Practice for Software Design Descriptions.

## Appendix B

### Guide to Specific Review Applications

This Appendix is added to guide in the application of review processes, and to delineate some specific applications.

Although review processes defined by this standard have distinct objectives, Table B1 is provided to further accent process distinctions.

In addition, the following review process applications are presented:

- |  |   |
|--|---|
| <ol style="list-style-type: none"> <li>(1) Software requirements review (SRR)</li> <li>(2) Preliminary design review (PDR)</li> <li>(3) Critical design review (CDR)</li> <li>(4) Software verification and validation plan review (SVVPR)</li> <li>(5) Managerial reviews (MR)</li> </ol> | <ol style="list-style-type: none"> <li>(4) Adequacy of requirements for the man-machine interface</li> <li>(5) Availability of constants and tables for calculations</li> <li>(6) Testability of the functional requirements</li> <li>(7) Adequacy and completeness of the verification and acceptance requirements</li> <li>(8) Conformance to requirements specification standards</li> <li>(9) Adequacy and feasibility of performance requirements</li> <li>(10) Adequacy of consideration for human factors</li> </ol> |
|--|---|

#### B1. Software Requirements Review (SRR)

The SRR is held to ensure the adequacy of the requirements stated in the software requirements specification. See ANSI/IEEE Std 730-1984 [A1].

The SRR is an evaluation of the software requirements specification (SRS) to ensure adequacy, technical feasibility, and completeness. Moreover, the SRR is held with user participation to examine the SRS to verify that it is unambiguous, verifiable, consistent, modifiable, traceable, and usable during the operation and maintenance phase. All organizational elements (for example, software design, software test, software quality assurance, systems engineering, marketing, manufacturing, and field service) that are directly affected by the requirements should participate.

The review checklist for the software requirements review may include but is not limited to the following:

- (1) Traceability and completeness of the requirement from the next higher level specification (such as a system specification or user requirements specification)
- (2) Adequacy of rationale for any derived requirements
- (3) Compatibility of external (hardware and software) interfaces

#### B2. Preliminary Design Review (PDR)

**B2.1** The PDR is held to evaluate the technical adequacy of the preliminary design of the software as depicted in a preliminary version of the Software Design Description. See ANSI/IEEE Std 730-1984 [A1].

Specifically, the need is to apply a review process to assess the progress, consistency, and technical adequacy of the selected design approach.

All organizational elements (for example, software test, software quality assurance, systems engineering, marketing, manufacturing, and field service) that are directly affected by the software design description should participate.

The following items may be included in the review checklist for the preliminary design review (PDR):

- (1) Compatibility of the interfaces between the software, hardware, and end users.
- (2) Fulfillment of the requirements contained in the SRS.
- (3) Conformance to design documentation standards.

**B2.2** Concerns that may be addressed by the checklist for preliminary design review

Table B1

Category and Attributes	Management Review	Technical Review	Software Inspection	Walkthrough
<b>Objective</b>	Ensure progress. Recommend corrective action. Ensure proper allocation of resources.	Evaluate conformance to specifications and plans. Ensure change integrity.	Detect and identify defects. Verify resolution.	Detect defects. Examine alternatives. Forum for learning.
<b>Delegated Controls</b>				
Decision making	Management team charts course of action. Decisions are made at the meeting or as a result of recommendations.	Review team petitions. Management or technical leadership to act on recommendations.	Team chooses from predefined product dispositions. Defects must be removed.	All decisions made by producer. Change is the prerogative of the producer.
Change verification	Change verification left to other project controls.	Leader verifies as part of review report.	Moderator verifies rework.	Change verification left to other project controls.
<b>Group Dynamics</b>				
Recommended size	Two or more persons.	Three or more persons.	Three to six persons.	Two to seven persons.
Attendance	Management, technical leadership, and peer mix.	Technical leadership and peer mix.	College of peers meet with documented attendance.	Technical leadership and peer mix.
Leadership	Usually the responsible manager.	Usually the lead engineer.	Trained moderator.	Usually producer.
<b>Procedures</b>				
Material volume	Moderate to high, depending on the specific <i>statement of objectives</i> for the meeting.	Moderate to high, depending on the specific <i>statement of objectives</i> for the meeting.	Relatively low.	Relatively low.
Presenter	Project representative.	Software element representative.	Presentation by <i>reader</i> other than producer.	Usually the producer.
Data collection	As required by applicable policies, standards, or plans.	Not a formal project requirement. May be done locally.	Formally required.	Not a formal project requirement. May be done locally.
<b>Outputs</b>				
Reports	Management review report.	Technical review reports.	Defect list and summary. Inspection report.	Walkthrough report.
Data base entries	Any schedule changes must be entered into the project tracking database.	No formal data base required.	Defect counts, characteristics, severity, and meeting attributes are kept.	No formal data base requirement.

(PDR) include (but are not limited to):

- (1) Software interfaces
- (2) Hardware interfaces
- (3) Baseline for configuration
- (4) Functional overview
- (5) Design alternative selection rationale
- (6) Standard operating environments
- (7) Human factors requirements
- (8) Documentation requirements
- (9) Testing considerations
- (10) Manufacturability
- (11) Maintainability
- (12) Assumptions
- (13) Risks

More information is available in ANSI/IEEE Std 1016-1987 [A2].

### B3. Critical Design Review (CDR)

The CDR is held to determine the acceptability of the detailed software designs as depicted in the software design description in satisfying the requirements of the software requirements specification. See ANSI/IEEE Std 730-1984 [A1].

The CDR is an evaluation of the completed software design description (SDD) to evaluate its technical adequacy, completeness, and correctness. Other concerns include its compatibility with the other software and hardware (where the product is required to interact), and the technical, cost, and schedule risks of the product design.

All other organizational elements that impose requirements or are affected by the design should participate (for example, software design, software test, and systems engineering).

The following items may be included in the review checklist for the critical design review:

- (1) The compatibility of the detailed design with the SRS
- (2) Available data in the form of logic diagrams, algorithms, storage allocation charts, and detailed design representations (for example, flow chart, program design language) to establish design integrity
- (3) Compatibility and completeness of interface design requirements
- (4) Adequacy and completeness of algorithms and equations

- (5) Correctness and suitability of logic descriptions that may be warranted
- (6) Technical accuracy and currency of all available test documentation and its compatibility with the test requirements of the SRS
- (7) The requirements for the support and test software and hardware to be used in the development of the product
- (8) The inclusion in the final design of function flow, timing, sizing, storage requirements, memory maps, data base, and other performance factors

### B4. Software Verification and Validation Plan Review (SVVPR)

The software verification and validation plan review is held to evaluate the adequacy and completeness of the verification and validation methods described in the SVVP. See ANSI/IEEE Std 730-1984 [A1].

The SVVPR is an evaluation of the completed software verification and validation plan (SVVP). Since the SVVP may be developed incrementally, multiple reviews may be required. These reviews are held to ensure that the verification and validation methods described in the SVVP are adequate and will provide complete evaluation data.

All organizational elements that impose requirements or are impacted by the SVVP should participate. These groups may include system engineering, software marketing, software design, software test, software quality assurance, customers, and users.

The following items may be included as review requirements for the software verification and validation plan review:

- (1) All verification and validation methods, along with completion criteria to assure traceability to, and compatibility with, the functional and performance requirements expressed in the SRS.
- (2) The methods to be used to verify and validate that the product satisfies the requirements of the SRS; these should include review process applications (for example, technical review, inspection, or walkthrough), demonstration, analysis, and test.

### **B5. Managerial Reviews**

- (3) Reports that adequately document the results of all reviews, audits, and tests based on the requirements listed in the SVVP.
- (4) Adequate descriptions of the software configuration to be tested including test support software and hardware.
- (5) Test plans, test procedures, test cases, and test data to ensure that all specification requirements are tested and that test instructions are clear and concise.
- (6) Test procedures and test cases to ensure that test inputs and success criteria are adequately defined.
- (7) A test plan and test case schedule identifying tests to be done, when, and by whom.
- (8) Conformance to test documentation standards.

These reviews are held periodically to assess the execution of (the SQA) plan. These reviews shall be held by an organizational element(s) independent of the unit being audited, or by a qualified third party. See ANSI/IEEE Std 730-1984 [A1].

The planned frequency and structure of the (SQA Plan) managerial reviews should be stated in the SQA Plan. They should be conducted at the direction of the program manager.

Although the application of the *Management Review Process* meets this review requirement, an alternative is to perform the *Quality Systems Audit* (as presented in Appendix C) to the extent possible given the particular degree of independence and authority.

**Appendix C****A Guide to Specific Audit Applications**

To guide in the application of the audit process, and to delineate some specific applications, this Appendix has been added. The following audit process applications are presented:

- (1) Functional audit (FA )
- (2) Physical audit (PA)
- (3) In-process audit (IPA)
- (4) Quality systems audit (QSA)

Where self-examination (that is, application of the audit process by an element(s) not organizationally independent of the area audited) is desired, the general audit process is applied, with the independence requirement waived.

**C1. The Functional Audit (FA )**

The objective of the functional audit (FA ) is to provide an independent evaluation of software products, verifying that its configuration items' actual functionality and performance is consistent with the requirement specifications. Specifically, this audit is held prior to the software delivery to verify that all requirements specified in the Software Requirements Specification have been met. See ANSI/IEEE Std 730-1984 [A1].

**C1.1** Central to the FA is the identification of software elements to be audited by:

- 1) Nomenclature
- 2) Specification identification number
- 3) Configuration item number

**C1.2** In addition, materials to be provided by the audited organization include:

- (1) The software requirements specification
- (2) Object copy of code
- (3) Current listing of waivers against specific configuration items
- (4) When available, the status of test programs to test configured items with automatic test equipment
- (5) The software verification and validation report
- (6) All in-process audit reports for items under audit (see C3)

- (7) Any other test documentation (for example, plans, specifications, procedures, and reports) for the configuration item
- (8) Listing of all successfully accomplished functional testing
- (9) Listing of all planned, yet unexecuted, functional testing

**C1.3** The following activities are required to conduct a functional audit:

- (1) An audit of the formal test documentation shall be performed against test data. Results shall be checked for completeness and accuracy, and all deficiencies documented.
- (2) An audit of the software verification and validation report shall be made to validate that the reports are accurate and completely describe the testing effort. All approved changes (problem reporting and corrective action) shall be reviewed to ensure that they have been technically incorporated and verified.
- (3) Updates to previously delivered documents shall be reviewed to ensure accuracy and consistency.
- (4) Preliminary design review and critical design review outputs shall be sampled to ensure that all findings have been incorporated and completed.

The audit team shall compare the code with the documented software requirements as stated in the current specifications document, to determine that the code addresses all (and only) the documented requirements.

**C1.4** This audit is concerned not only with functionality, but also with performance as follows: Testing accomplished with appropriate test documentation and validated data shall sufficiently establish configuration item performance. For performance parameters that can not be verified completely during testing, simulations or other analysis shall be accomplished to establish confidence that the configuration item does meet performance criteria when in operation as stated in its specifications.

The FA report should include an evaluation of each software element(s) examined, and provides this evaluation within the standard audit report framework. The evaluation shall take the form of

- (1) *Approval*. If the software element(s) had been observed as having all required functions and shows evidence of meeting minimum performance standards.
- (2) *Contingent Approval*. If the evaluation of the software element(s) would result in approval given successful completion of specific, well-defined (observable) corrective action.
- (3) *Disapproval*. If the software element(s) were observed to be seriously inadequate.

## C2. The Physical Audit (PA)

The objective of the physical audit (PA) is to provide an independent evaluation of a software product configuration item to confirm that components in the *as-built version* map to its specifications. Specifically, this audit is held to verify that the software and its documentation are internally consistent and are ready for delivery. See ANSI/IEEE Std 730-1984 [A1].

**C2.1** Central to the PA is the identification of software elements to be audited by

- (1) Nomenclature
- (2) Specification identification number
- (3) Configuration item number

**C2.2** In addition, materials to be provided by the audited organization include

- (1) Current listing of waivers against specific configuration items
- (2) Requirements and architectural and detailed design specifications
- (3) Listing of approved changes against the configuration with outstanding changes identified
- (4) Acceptance test documentation (for example, plans, specifications, procedures, and reports) for the configuration item
- (5) Appropriate customer deliverable documentation (for example, operating manuals, and maintenance manuals)

- (6) Approved nomenclature, markings, and nameplates
- (7) Software version description document
- (8) FA reports

In addition, the audited organization shall identify any differences between the configuration of the system in the development environment, and the specific configuration of the system under examination. Moreover, they shall certify or otherwise demonstrate that any differences do not degrade the functional or performance characteristics of the system.

**C2.3** To provide evidence of adequate control of system content and consistency, the audit team should as a minimum, examine

- (1) The system specification document(s) for format and completeness
- (2) Functional (FA) reports for discrepancies and actions taken
- (3) Design descriptions (at least on a sampling basis) for proper symbols, labels, references, and data descriptions
- (4) Design descriptions at various levels of abstraction, cross-checking for consistency
- (5) Manuals (for example, user's manuals, programmer's manuals, and operator's manuals,) for format completeness and conformance with data item descriptions. Formal verification or acceptance of these manuals shall be waived until all other system testing is completed to ensure that procedural contents are correct.
- (6) The software element(s) delivery media (and media controls) to ensure proper transfer or transmittal.

The audit team shall examine the system for vendor-provided or supplier-developed software elements, and seek evidence that any such software elements were produced under reasonable and prudent quality controls.

**C2.4** The audit team shall examine the engineering release and change control function (and associated documentation) for evidence that it can

- (1) Identify the composition of any configuration in terms of its subordinate units
- (2) Determine, given a unit identifier, the parent element

- (3) Describe the composition of a configuration item or its identifier with respect to other configuration items or identifiers
- (4) Identify specifications relating to a given configuration item
- (5) Identify outstanding changes against a given configuration item
- (6) Identify changes and retain records of earlier configurations

**C2.5** The PA report should include an evaluation of all software elements as a whole, and shall provide this evaluation within the standard audit report framework. The evaluation shall take the form of

- (1) Approval. When the system of software elements has been observed as having all deliverable components accounted for, and of compatible issue.
- (2) Contingent Approval. When the evaluation of the system of software elements results in approval given successful completion of specific, well-defined (observable) corrective action.
- (3) Disapproval. When the system of software elements was observed to be seriously inadequate.

**C2.6** The audit team shall examine the engineering release and change control function (and associated documentation) for evidence that it can

- (1) Identify the composition of any configuration in terms of its subordinate units
- (2) Determine, given a unit identifier, the parent element
- (3) Describe the composition of a configuration item or its identifier with respect to other configuration items' identifiers
- (4) Identify specifications relating to a given configuration item

### **C3. In-Process Audits**

In-process audits are held during the design and development phases prior to the functional audit to verify the consistency of the design.

The objective is to verify the consistency of the product as it evolves through the development process by determining, on a sampling basis, that

- (1) Hardware and software interfaces are consistent with design requirements in the software requirements specification (SRS)
- (2) The code is fully tested by the SVVP to ensure that the functional requirements of the SRS are satisfied
- (3) The design of the product, as the software design description (SDD) is evolving, satisfies the functional requirements of the SRS
- (4) The code is consistent with the SDD

Moreover, in-process audits can be applied to determine compliance with applicable process standards (for example, auditing of the inspection process, on a sampling basis, to determine if inspection process rules are being followed).

Input material should include formal process descriptions and a product sample evidencing process execution.

In-process audit procedures are unique to the process and product examined. Specific procedures should be established locally, using the general steps found in Section 8.

The results of all in-process audits shall be documented in in-process audit reports, which identify all discrepancies found, and provide an overall opinion accordingly. The opinion should take the form of approval, contingent approval, or disapproval. Significant compliance deviations are noted and must be followed up to ensure their correction or waiver. This occurs during or before the functional audit.

### **C4. The Quality Systems Audit (QSA)**

The purpose of a quality systems audit, or quality program evaluation, is to provide an independent assessment of the compliance to a software quality assurance plan.

**C4.1** Specifically, the QSA objective is to determine, based on observable (verifiable) evidence, that

- (1) The software quality program documentation established by the development organization addresses as a minimum, the basic elements of ANSI/IEEE Std 730-1984 [A1], or other appropriate standards.

- (2) The software development organization follows its documented software quality program.

The SQA plan must incorporate all objective organizational performance criteria; internal standards and procedures; process requirements mandated by law, contract, or other policy; and conform to the ANSI/IEEE 730-1984 [A1] or other appropriate standards for software quality assurance.

**C4.2** Central to the QSA is the identification of quality program documentation, and its availability to the audit team.

The following activities are required:

- (1) A preaudit conference with audited organization personnel is held to familiarize them with audit goals and procedures
- (2) Examination of quality program documents and related items for com-

pleteness, traceability, and consistency

- (3) Perform selective compliance testing to objectively ascertain that SQA procedures and standards are implemented correctly
- (4) Interview staff
- (5) Perform in-process audits as appropriate (for example, observing inspections and their documented evidence)
- (6) Examine reports from any recent functional and physical configuration audits.

The QSA report should include an evaluation of the quality program as a whole, and should provide this evaluation within the standard audit report framework.

The evaluation (following the general format of Section 8) shall take the form of approval, contingent approval, or disapproval for each major program element(s) identified in the audit plan.

### Acknowledgments

The following organizations supported the development of this standard:

A. Frank Ackerman Associates  
AT&T Bell Laboratories  
AT&T Information Systems  
AT&T Technologies  
Bell Canada  
Bell Communications Research  
Burroughs Corporation  
C.S. Draper Laboratory  
CAP GEMINI/DASD  
Coldframe, Inc  
Combustion Engineering  
Computer Sciences Corporation  
DP Auditing Report  
Grumman-CTEC  
■ IBM Corp  
Intermetrics, Inc  
Martin-Marietta  
McDonnell Douglas Corporation  
Motorola  
Naval Weapons Center  
NCR Corporation  
Northern Telecom, Ltd  
Northern Telecom, Inc  
OASAS, Ltd  
Pacific Gas and Electric  
Pansophic Systems, Inc  
Pinpoint Retail Systems, Inc  
Quality Assurance Institute  
RCA  
RGE Software Methodologies, Inc  
Sperry  
US Department of Transportation  
University of California, Sacramento  
Worldwide Service Technologies, Inc

The IEEE offers seminars on Software Engineering throughout the year. You'll learn about:

- ✓Project Management Planning
- ✓Verification and Validation
- ✓Testing
- ✓Configuration Management
- ✓Software Quality Assurance
- ✓Software Requirements Specifications
- ✓Reviews and Audits

For details, write or call

IEEE Service Center  
445 Hoes Lane  
P.O. Box 1331  
Piscataway, NJ 08855-1331 USA  
1-800-678-IEEE