

# **1. Estándares IEEE para el Aseguramiento de la Calidad del Software**

## **1.1. Introducción**

El Institute of Electrical and Electronics Engineers (IEEE) se formó en 1963 cuando el American Institute of Electrical Engineers (AIEE) se fusionó con el Institute of Radio Engineers (IRE). Su sede central está en Nueva York. Generalmente, los estándares IEEE relacionados con el aseguramiento de la calidad del software (SQA) son concebidos y patrocinados por la IEEE Computer Society y desarrollados por la IEEE Standards Association (IEEE-SA).

Los estándares de ingeniería del software del IEEE proporcionan el conjunto de requerimientos y guías más importante para el aseguramiento de la calidad del software.

## **1.2. IEEE Std 730™-2002 - Standard for Software Quality Assurance Plans**

La planificación de la calidad es el proceso en el cual se desarrolla un plan de calidad para un proyecto determinado. El plan de calidad define la calidad del software deseado y describe cómo valorarla. Por lo tanto, define lo que es software de “alta calidad”.

El plan de calidad selecciona los estándares organizacionales apropiados para un producto y un proceso de desarrollo particular. Si en el proyecto se utilizan nuevos métodos y herramientas, se tienen que definir nuevos estándares.

El Estándar IEEE 730 es una recomendación para elaborar un Plan de Aseguramiento de la Calidad del Software (SQAP, Software Quality Assurance Plan) para los proyectos de desarrollo de software. Proporciona los requisitos mínimos aceptables para la preparación y el contenido de los planes de aseguramiento de la calidad de software. Fue escrito para ser utilizado en las fases de desarrollo y mantenimiento del software. El plan SQA sirve como guía de las actividades de SQA en el proyecto.

Este estándar describe la preparación y los contenidos de los planes SQA. Las actividades principales del SQA incluyen la gestión, documentación, mediciones, revisiones, testing, informes de problemas y las acciones correctivas, control de medios de comunicación, control de proveedores, gestión de registros, capacitación y gestión de riesgos. En las descripciones de las piezas relacionadas con el plan de SQA, el Estándar IEEE 730 nos proporciona una valiosa información sobre cada una de estas actividades.

En la cláusula 4, se describe el contenido mínimo de un plan de SQA. Dentro de las descripciones se identifican los elementos fundamentales del proceso de SQA. En un plan de SQA se pueden aplicar o citar requisitos y directrices establecidas en otros estándares de la IEEE.

Cuando en un proyecto de desarrollo de software se incluye un plan de este tipo, las decisiones relacionadas con la calidad deben ser tomadas con anticipación y, por lo tanto, deben ser estudiadas y razonadas suficientemente antes de iniciar el desarrollo. El equipo de desarrollo deberá entonces ajustarse al plan y así, mejorar continuamente los procesos de desarrollo en beneficio del proyecto en curso y de los proyectos futuros.

Este estándar se aplica (o puede afectar) a tres grupos: el usuario, el proveedor y el público.

- a) El usuario, que puede ser otro elemento de la misma organización que desarrolla el software, que necesita el producto con un grado razonable de confianza.
- b) El proveedor, que necesita establecer un estándar para planificar y medir.
- c) El público, que puede verse afectado por el uso del producto.

### 1.3. Plan SQA

Un Plan SQA puede incluir los siguientes puntos:

#### 1. Propósito:

- Delinea el propósito específico y el alcance del plan SQA.
- Lista los nombres de los elementos software cubiertos por el plan SQA y el uso de dichos elementos.
- Determina la porción del ciclo de vida cubierta por el plan para cada elemento software.

#### 2. Documentos de referencia

- Proporciona una lista completa de los documentos referenciados en el plan o utilizados en su elaboración.

#### 3. Gestión

- Está muy ligado al plan del proyecto del software.
- Idealmente redactado en formato IEEE Std. 1058-1998.

##### 3.1. Organización

- Describe la estructura organizativa que influye y controla la calidad del software.
- Identifica roles y responsabilidades dentro del plan SQA.
- Identifica a los responsables de preparar y mantener el plan SQA.

### 3.2. Tareas

- Describe:
  - La porción del ciclo de vida cubierta por el plan SQA.
  - Las tareas a desarrollar.
  - Los criterios de entrada y salida para cada tarea.
  - Las relaciones entre estas tareas y los principales puntos de control planeados.

### 3.3. Roles y responsabilidades

- Identifica los elementos organizativos específicos responsables de llevar a cabo cada tarea.

### 3.4. Recursos estimados de garantía de calidad

- Proporciona la estimación de recursos y costes gastados en garantía de calidad y en las tareas de control de calidad.

## 4. Documentación

- Describe toda la documentación que se va a generar durante el proceso de desarrollo.

### 4.1. Propósito

- Identifica la documentación que dirige el desarrollo, verificación y validación, uso y mantenimiento del software.
- Lista los documentos que serán revisados o auditados, así como los criterios de revisión.

### 4.2. Requisitos mínimos de documentación

- Para asegurar que la implementación del software satisface los requisitos técnicos, se requiere como mínimo la siguiente documentación:
  - Descripción de requisitos software.
  - Descripción del diseño del software.
  - Planes de verificación y validación.
  - Informe de resultados de verificación e informe de resultados de validación
  - Documentación de usuario.
  - Plan de gestión de la configuración software.

#### 4.2.1. Descripción de requisitos software

- Es la SRS (Software Requirements Specification).
- Idealmente redactada según IEEE Std 830-1998.

#### 4.2.2. Descripción de diseño software

- Describe la estructuración del software para cumplir con los requisitos de la SRS.
- Debe describir los componentes y subcomponentes del diseño del software.
- Idealmente redactado según IEEE Std. 1016-1998, IEEE Recommended Practice for Software Design Descriptions.

#### 4.2.3. Planes de validación y verificación

- Estos planes se utilizan para determinar si el producto software desarrollado se ajusta a sus requisitos, y si cumple con las expectativas del usuario.
- Idealmente redactado según los estándares:
  - IEEE Std. 829-1998 for Software Test Documentation.
  - IEEE Std. 1008-1997 IEEE for Software Unit Testing.
  - IEEE Std. 1012-1998 for Software Validation and Verification.

#### 4.2.4. Informe de resultados de verificación e informe de resultados de validación

- Describen los resultados de las actividades de verificación y planificación del software llevados a cabo según los planes descritos en el punto anterior.

#### 4.2.5. Documentación de usuario

- La documentación de usuario guía al usuario en la instalación, operación, gestión y mantenimiento de los productos software.
- Debería describir las entradas y salidas, así como los mensajes de error.
- Idealmente redactado según IEEE Std. 1063-1987 for Software User Documentation.

#### 4.2.6. Plan de gestión de la configuración software

- Describe el proceso de gestión de configuración software.
- Idealmente redactado según IEEE Std. 828-1998 for Software Configuration Management Plans.

#### 4.3. Otra documentación

- Identifica otros documentos necesarios durante el proceso de desarrollo, como:
  - Plan de proceso de desarrollo.
  - Descripción de estándares de desarrollo de software.
  - Descripción de métodos/procedimientos/herramientas de IS.
  - Plan de gestión del proyecto de software (idealmente según IEEE Std. 1058).
  - Plan de Mantenimiento (idealmente según IEEE Std. 1219-1998).
  - Planes de seguridad del software (idealmente según IEEE Std. 1228-1994).
  - Plan de integración del software.

#### 5. Estándares, prácticas, convenciones y métricas

- Esta sección es un poco miscelánea en SQA.

##### 5.1. Propósito

- Identifica:
  - Estándares.
  - Prácticas.
  - Convenciones.
  - Técnicas estadísticas.
  - Métricas aplicables al proyecto.
- Las medidas se incluirán en las métricas utilizadas y podrían identificarse en un plan de medición independiente (idealmente redactados según IEEE Std. 1219-1998 for Software Maintenance e IEEE Std. 1228-1994 for Software Safety Plans).
- También determina como se monitoriza y garantiza la conformidad con el plan.

##### 5.2. Contenido

- Como mínimo debe incluir:
  - Estándares de documentación.
  - Estándares de diseño.
  - Estándares de codificación.
  - Estándares de comentarios.
  - Prácticas y estándares de prueba.
  - Métricas del producto y proceso de garantía de calidad seleccionada.

## 6. Revisiones del software

- Determina las revisiones del software.

### 6.1. Propósito

- Fija las revisiones del software.
- Idealmente redactado según IEEE Std. 1028-1997.

### 6.2. Requisitos mínimos

- Como mínimo deberían producirse las siguientes revisiones:
  - Revisión de las especificaciones software.
  - Revisión del diseño arquitectónico.
  - Revisión del diseño detallado.
  - Revisión del plan de verificación y validación.
  - Auditoría de la funcionalidad (cumplir SRS).
  - Auditoría física (consistencia y fecha entrega).
  - Auditoría durante el proceso (consistencia del diseño).
  - Revisiones de gestión (garantizar cumplimiento plan SQA).
  - Revisión del plan de gestión de la configuración software.
  - Revisión post-implementación.

### 6.3. Otras revisiones y auditorías

- Por ejemplo, revisión de la documentación de usuario.

## 7. Prueba

- Identifica todas las pruebas no incluidas en el plan de verificación y validación.

## 8. Informe de problemas y acción correctiva

- Describe las prácticas y procedimientos de informe, seguimiento y resolución de problemas, tanto a nivel producto como proceso.
- Determina las responsabilidades organizativas relativas a su implementación.

## 9. Herramientas, técnicas y metodologías

- Herramientas, técnicas y metodologías utilizadas para soportar el proceso de SQA.

## 10. Control de medios

- Determina los métodos para:
  - Identificar el medio físico de cada producto software.
  - Protegerlo de daños durante el proceso.

## 11. Control de proveedores

- Determina las técnicas para garantizar que el software proporcionado por proveedores externos cumple sus requisitos.
- También es aplicable a código heredado.

## 12. Colección de registros, mantenimiento y conservación

- Identifica la documentación SQA que no se debe tirar tras acabar el proceso.
- Determina los métodos y medios para ensamblar, archivar, proteger y mantener la documentación.
- Fija el periodo de conservación de la información.

## 13. Formación

- Identifica las actividades de formación necesarias para satisfacer las necesidades del plan SQA.

## 14. Gestión del riesgo

- Especifica el plan de gestión del riesgo.
- Idealmente redactado según IEEE Std. 1540-2001 for Software Life Cycle Processes – Risk Management

## 15. Glosario

- Términos específicos del plan SQA.

## 16. Procedimiento de cambio e historia del plan SQA

- Procedimientos de modificación del plan SQA.
- Procedimientos de mantenimiento del historial de cambios.
- Historial de cambios.

#### **1.4. Relación del Estándar IEEE 730 con las Normas ISO 9000**

- El ISO 9001 y el IEEE 730 se relacionan más directamente con la Norma ISO 9000-3.
- IEEE 730 se centra en el plan de garantía de calidad y en como implementarlo, mientras que ISO 9001 busca una gestión global de la calidad centrada en auditorías externas.
- El IEEE 730 es el puente entre la gestión de la calidad y la Ingeniería del Software.



## 2. Estándares del U.S. Department of Defense

### 2.1. Introducción

Durante los años 1970s, 1980s y 1990s, los estándares de mayor interés para las personas dedicadas al SQA eran las publicaciones del U.S. Department of Defense (DoD). Hoy en día, las personas que trabajan en SQA se interesan principalmente en los estándares no gubernamentales (NGSs).

La razón principal de este cambio fue que, a mediados de la década de 1990, el gobierno de EE.UU. optó por el uso de las normas de consenso voluntario en lugar de las normas hechas únicamente para el estado. Esto incluía al DoD, sin embargo, no significaba que el uso de los estándares relacionados con el software era obligatorio para el mismo.

Todos los esfuerzos de estandarización del DoD, incluyendo el cumplimiento de las políticas relacionadas con el mismo, son coordinados y administrados por el Defense Standardization Program (DSP).

### 2.2. Estándares Influyentes del Pasado

A continuación hacemos una rápida revisión de algunos de los estándares más influyentes del DoD de los últimos 30 años.

#### 2.2.1. Estándares de Desarrollo de Software Anteriores al MIL-STD-498

**DOD-STD-1679A, Software Development (1983)**, sustituye al MIL-STD-1679, Weapon System Software Development (1978). Esta serie independiente fue una de las primeras en describir lo que se conoce como el proceso de desarrollo del software. Incluía los requisitos para las actividades básicas de desarrollo tales como la gestión, definición de requerimientos, diseño, codificación y prueba, y también lo que ahora se conoce como actividades de soporte, tales como la gestión de configuración y el control de calidad. Fue sustituido en 1985 por el DOD-STD-2167.

**DOD-STD-2167, Defense System Software Development (1985)**, fue una mirada distinta de la misma parte del ciclo de vida del software (desarrollo) que había descrito el DOD-STD-1679A. Este se diferenció del estándar anterior de varias formas. Incorporó siete partes de otros estándares por referencia. Describía la gestión de configuración y el control de calidad del software. Este estándar influyó en los primeros modelos de capacidad y madurez para el software. Fue reemplazado por el DOD-STD-2167A en 1988.

**DOD-STD-2167A, Defense System Software Development (1988)**, fue una mejora y reducción del DOD-STD-2167. Se redujo el número de estándares que habían sido incorporados por referencia. Se separaron los requisitos para la gestión de la configuración y evaluación de la calidad del software. Se agregó una conexión explícita entre el desarrollo del software y el contexto que rodea el desarrollo del sistema. Fue reemplazado por el estándar MIL-STD-498 en 1994.

**DOD-STD-2168, Defense System Software Quality Program (1988)**, se convirtió en un estándar compañero para el DOD-STD-2167A. Este estándar fue una descripción de un programa de calidad del software para la adquisición, desarrollo y soporte de sistemas de software. Interpretó los requisitos aplicables del estándar MIL-Q-9858 para el software e incorporó los requisitos aplicables del estándar MIL-STD-1535. El estándar fue cancelado en 1995.

**DOD-STD-7935A, DOD Automated Information Systems (AIS) Documentation Standards (1988)**, describe los requisitos para 11 tipos diferentes de documentos de los que ahora son llamados sistemas de TI o aplicaciones. Fue sustituido por el estándar MIL-STD-498 en 1994.

### **2.2.2. MIL-STD-498 - Software Development and Documentation**

El estándar MIL-STD-498, aprobado para el uso del DoD el 5 de diciembre de 1994, armonizaba y sustituía a dos estándares anteriores, el DOD-STD-2167A y el DOD-STD 7935A, colocando de esta forma el desarrollo de todo el software del DoD bajo un único estándar. Interpretaba todas las cláusulas aplicables del estándar MIL-Q-9858A, ofreciendo así una alternativa al DOD-STD-2168. También interpretaba todas las cláusulas aplicables al software de la norma ISO 9001, e implementaba los procesos de desarrollo y documentación de la norma ISO/IEC 12207 (la versión preliminar de la Norma Internacional), con lo que se logró la armonización de los estándares militares de los EE.UU. para el desarrollo de software con los principales estándares internacionales de calidad y los procesos del ciclo de vida del software.

El MIL-STD-498 se diferenció de los estándares anteriores también de otras maneras. Era independiente, como lo era el DOD-STD-1679A, es decir, que no dependía de otros estándares. Se eliminaron las agrupaciones de las actividades de ingeniería y soporte que se habían creado para el DOD-STD-2167. Las agrupaciones en el DOD-STD-2167 habían sugerido el ciclo de vida en cascada. En el MIL-STD-498 se contemplaban modelos alternativos del ciclo de vida, siempre y cuando resultaran apropiados para la situación del desarrollo.

En concordancia con su mayor amplitud de alcance y mayor flexibilidad en sus requisitos, el MIL-STD-498 asignaba una mayor responsabilidad a los usuarios para adaptar el estándar a su situación de desarrollo.

Como los estándares DOD-STD-2167 y DOD-STD-2167A, el MIL-STD-498 también influyó en el contenido del Modelo de Capacidad y Madurez para el software (CMM, Capability Maturity Model) y en la norma ISO/IEC 12207:1995, que apareció en sus primeras versiones durante el mismo período del 1990. Más tarde, las ideas y las grandes cantidades de texto del estándar MIL-STD-498 fueron incorporados dentro del IEEE/EIA 12207.2 (algunas sin ningún tipo de modificación).

El estándar MIL-STD-498 fue cancelado el 27 de Mayo de 1998, cuando el DoD adoptó el estándar IEEE/EIA 12207.

### **3. ISO/IEC 12207, EIA/IEEE J-STD-016 e IEEE/EIA12207**

#### **3.1. Introducción**

En esta sección se describen los estándares EIA/IEEEJ-STD-016, ISO/IEC 12207 e IEEE/EIA 12207. Al final de la misma se compran las características comunes.

#### **3.2. ISO/IEC 12207 - Systems and Software Engineering - Software Life Cycle Processes**

En 1987 la International Organization for Standardization (ISO) y la International Electrotechnical Commission (IEC) crearon el Joint Technical Committee (JTC1) sobre Tecnologías de la Información. El alcance del JTC1 es la "normalización en el campo de los sistemas de tecnología de la información (incluidos los sistemas de microprocesadores) y equipos."

En junio de 1989, el JTC1 inició la elaboración de una Norma Internacional, la ISO/IEC 12207, sobre los procesos del ciclo de vida del software con la finalidad de cubrir una necesidad crítica. Desde la era de la industria "cottage" de finales de 1970, el software se ha ido consolidando como parte integral de muchas disciplinas científicas y empresariales. Sin embargo, los entornos de desarrollo y gestión del software han estado proliferando por la falta de un marco uniforme para la gestión y la ingeniería del software. La Norma Internacional satisface esa necesidad fundamental de establecer un marco común que pueda ser utilizado por los profesionales del software para la gestión y la ingeniería del software. Además, con un marco uniforme se promueve el comercio internacional de los productos y servicios de software.

La primera versión de la Norma Internacional ISO/IEC 12207 fue publicada el 1° de Agosto de 1995. En la elaboración de la norma participaron los siguientes países: Australia, Brasil, Canadá, República Checa, Dinamarca, Finlandia, Francia, Alemania, Irlanda, Italia, Japón, Corea, Países Bajos, España, Suecia, el Reino Unido, y Estados Unidos de América.

La norma es voluntaria, es decir, no impone ninguna obligación a nadie para seguirla. Sin embargo, puede ser impuesta por una organización a través de política interna o por particulares a través de acuerdos contractuales.

ISO/IEC 12207 establece un proceso de ciclo de vida para el software que incluye procesos y actividades que se aplican desde la definición de los requisitos, pasando por la adquisición y configuración de los servicios del sistema, hasta la finalización de su uso. Creció a partir de una amplia gama de otras normas, marcos, etc., incluyendo la serie ISO 9000, el estándar IEEE Std 1074 (IEEE Standard for Developing Software Life Cycle Processes) y el estándar MIL-STD-498. Sin embargo, es muy diferente a los anteriores.

Es la primera Norma Internacional que proporciona un conjunto completo de procesos para la adquisición y el suministro de productos y servicios de software. Abarca aspectos técnicos y

comerciales, y se aplica a los compradores, proveedores, desarrolladores, operadores y personal de mantenimiento (estos son los cinco procesos primarios en la norma).

Una limitación potencialmente importante de la norma es que no se pretende que sea aplicable a la adquisición de productos de software COTS (Commercial Off The Shelf), aunque no está claro el motivo. La norma está destinada principalmente a aplicarse en el contexto de un contrato de dos partes. Tal vez, el uso de este tipo de contrato para la compra de COTS resulta excesivo y, por lo tanto, la norma se considera inadecuada. No parece haber ninguna razón por la cual no se deba aplicar ISO/IEC 12207, de forma voluntaria, al desarrollo, operación y mantenimiento de software COTS, solo resulta inadecuada para su adquisición y suministro.

La estructura de la norma ha sido concebida de manera flexible y modular de forma tal que pueda ser adaptada a las necesidades de cualquiera que la use. Para conseguirlo, la norma se basa en dos principios fundamentales: modularidad y responsabilidad. Con la modularidad se pretende conseguir procesos con un mínimo acoplamiento y una máxima cohesión. En cuanto a la responsabilidad, se busca establecer un responsable para cada proceso, facilitando la aplicación de la norma en proyectos en los que pueden existir distintas personas u organizaciones involucradas.

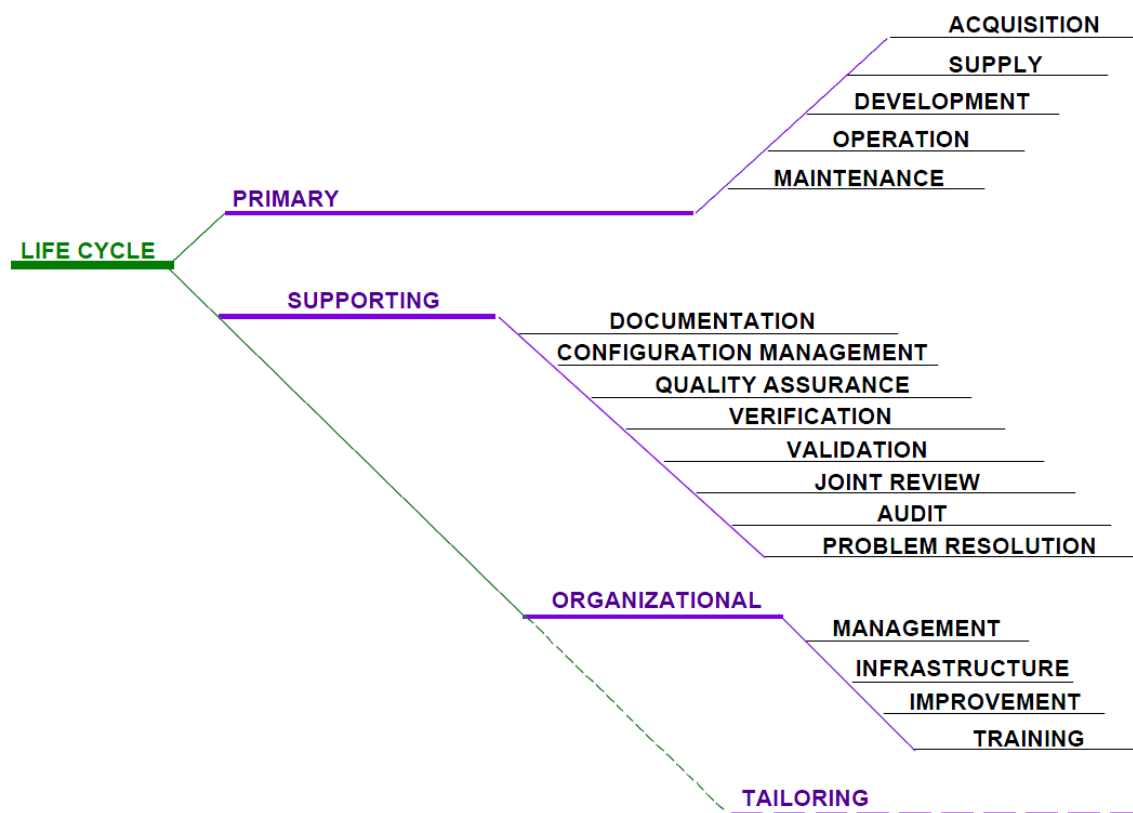


Figura 4.1: Proceso del Ciclo de Vida según ISO/IEC 12207

ISO/IEC 12207 describe la arquitectura de los procesos del ciclo de vida del software, pero no especifica los detalles de cómo implementar o realizar las actividades y tareas incluidas en los

misimos. Esta norma agrupa las actividades que deben ser realizadas durante el ciclo de vida del software en: 5 Procesos Principales, 8 Procesos de Soporte y 4 Procesos Organizacionales (Ver Figura 4.1). Cada proceso del ciclo de vida está dividido en un conjunto de actividades donde cada actividad está dividida en un conjunto de tareas.

### **3.3. EIA/IEEE J-STD-016 - Interim Standard for Information Technology Software Life Cycle Processes Software Development Acquirer-Supplier Agreement (Issued for Trial Use)**

EIA/IEEE J-STD-016-1995 es un estándar americano de prueba o provisional desarrollado por el mismo grupo que produjo IEEE/EIA 12207, publicado en Enero de 1996. Es esencialmente una versión desmilitarizada del MIL-STD-498, en la cual se elimina el lenguaje gubernamental (tal como las referencias a las Federal Aviation Requirements) y presenta los conceptos y la terminología de las normas internacionales. Con sus raíces en el estándar MIL-STD-498, el J-STD-016 se centra principalmente en el proceso de desarrollo en lugar de los otros procesos primarios de la norma ISO/IEC 12207 (a pesar de la presencia de "comprador" y "proveedor" en el título del J-STD-016). Sin embargo, se ha descrito como el primer paso en la aplicación de la norma ISO/IEC 12207.

En cierto sentido, J-STD-016 es un puente entre los estándares de defensa anteriores (MIL-STD-498) y el IEEE/EIA 12207. Dado que se reconoce que la aplicación del nivel de organización de los procesos (al estilo de IEEE/EIA 12207) puede ser muy costosa y requiere mucho tiempo, J-STD-016 es la opción disponible que puede estar más estrechamente relacionada con los procesos actualmente en ejecución. J-STD-016 es adecuado para su uso en proyectos en curso iniciados bajo el estándar MIL-STD-498 o uno de sus antecesores, o por las empresas que han implementado los procesos organizativos basados en el estándar MIL-STD-498.

J-STD-016 define un conjunto de actividades para el desarrollo y los productos resultantes de software. Proporciona un marco para la planificación del desarrollo y la ingeniería del software. J-STD-016 incorpora veintidós descripciones de los productos de software derivados de los veintidós Data Item Descriptions (DIDs) del estándar MIL-STD-498.

Al igual que la ISO/IEC 12207, J-STD-016 debe ser adaptado para asegurar que los requisitos necesarios se aplican. A diferencia de ISO/IEC12207, J-STD-016 cubre totalmente la reutilización del software (COTS).

### **3.4. IEEE/EIA 12207 - Systems and Software Engineering - Software Life Cycle Processes**

IEEE/EIA12207 es una adaptación americana de la norma ISO/IEC 12207 desarrollada por la Electronic Industries Alliance (EIA) y el Institute of Electrical and Electronics Engineers (IEEE) en colaboración con el U.S. Department of Defense (DoD). Esencialmente, es la norma ISO/IEC 12207 con el agregado de material extra muy influenciado por la ingeniería y los requisitos de los

estándares MIL-STD-498 y J-STD-016. La mayor parte del material adicionado está relacionado con el proceso de desarrollo. Los objetivos de la IEEE/EIA 12207 son:

- representar las mejores prácticas comerciales,
- ser adecuado para su aplicación en la adquisición de la defensa y
- ser compatible con otros estándares en el mercado mundial de software.

La intención es que, desde el punto de vista del proceso, IEEE/EIA 12207 sirva como un punto de acceso único a todos los estándares de la colección de ingeniería de software del IEEE.

Los defensores de la norma IEEE/EIA consideran que su mejora más importante con respecto a la ISO/IEC 12207 es un conjunto de alternativas para la conformidad con la norma, además de la conformidad del proyecto.

Se utiliza preferente en organizaciones que desarrollan su propio conjunto de procesos y procedimientos conformes con todos los requisitos de IEEE/EIA 12207 y se aplica a toda la organización. En comparación con la norma ISO/IEC 12207 (y MIL-STD-498), el IEEE/EIA 12207 se enfoca más sobre el cumplimiento a nivel organizacional. Debido a esto, el IEEE/EIA 12207 es más desalentador en la adaptación que la ISO/IEC 12207. Esto a su vez hará que los reclamamos por cumplimiento para IEEE/EIA 12207 sean más significativos que los reclamamos similares para ISO/IEC 12207.

La Norma IEEE/EIA 12207 está compuesta por tres partes:

**a) IEEE/EIA 12207.0-1996**

La Parte 0 contiene el texto completo de la norma ISO/IEC 12207:1995 sin las modificaciones posteriores. Las principales diferencias entre IEEE/EIA 12207.0-1996 e ISO/IEC 12207:1995 incluyen dos nuevos anexos:

- un conjunto de procesos y datos que ayudan a interpretar los objetivos de los requisitos especificados en ISO/IEC 12207 y
- una cláusula de sustitución que cambia el énfasis hacia el cumplimiento a nivel organizacional y requiere de documentación sobre los medios para el cumplimiento.

La norma IEEE/EIA 12207.0 contiene un amplio conjunto de procesos que deben ser adaptados para una situación y un propósito en particular. El proceso de adaptación en IEEE/EIA 12207.0 es el mismo que en la norma ISO/IEC 12207.

La conformidad con esta norma no está definida. Sin embargo, en el Anexo F se define el cumplimiento. En F.1, la definición de cumplimiento en la norma IEEE/EIA 12207.0-1996 es "similar" a la definición de cumplimiento en la norma ISO/IEC 12207:1995. En la cláusula F.2 se añaden las condiciones de cumplimiento relacionadas con la situación, es decir, si se reclama el cumplimiento a una organización, un proyecto, un programa multi-proveedor o

para cumplir con las decisiones reguladoras. En la cláusula F.3 se definen dos niveles de cumplimiento, adaptado y absoluto. En la cláusula F.4 se añaden en la norma dos conjuntos de criterios diferentes para realizar un proceso de ciclo de vida que fue seleccionado por el proceso de adaptación: cumplimiento por "como se especifico" y cumplimiento por "método alternativo". Un reclamo de cumplimiento según la norma IEEE/EIA 12207.0-1996 debe contener tres elementos: la situación (cláusula F.2), el nivel seleccionado (cláusula F.3) y los criterios elegidos (cláusula F.4).

#### **b) IEEE/EIA 12207.1-1997**

La Parte 1 es una guía sobre los elementos de información (los datos del ciclo de vida) mencionados en la IEEE/EIA 12207.0-1.996 (la norma básica). En total, más de 100 elementos de información diferentes son requeridos o recomendados por la norma básica.

Los elementos de información aparecen en orden alfabético en la Tabla 1 (en la cláusula 4), que es el corazón de esta guía. La Tabla 1 muestra para cada elemento donde es referenciado en la norma básica y de qué tipo es. Hay siete tipos diferentes de elementos: descripción, plan, procedimiento, registro, comunicación, solicitud y especificación.

Para algunos elementos, la Tabla 1 proporciona donde encontrar información adicional, como una descripción detallada dentro de la misma guía o de otras fuentes de información, como otros estándares IEEE. Finalmente, los datos descritos en las normas del software de IEEE y los sistemas de recopilación de normas de ingeniería se armonizarán con la serie IEEE/EIA 12207, en parte por la asignación en la Tabla 1 para los otros estándares IEEE, y, en parte, por la armonización del lenguaje en las demás normas (por ejemplo, los anexos).

La conformidad con la norma IEEE/EIA 12207.1-1997 no está definida. Sin embargo, el volumen contiene una cláusula de cumplimiento que le permite ser utilizado como una norma. Cuando se utiliza como una norma, y no sólo como una guía, se pueden realizar dos tipos de reclamos de cumplimiento diferentes. En primer lugar, uno o más documentos pueden ser reclamados para cumplir con uno o más de los elementos de información de la Tabla 1 cuando se cumplan las características relacionadas a las tablas de resumen. En segundo lugar, un proceso organizacional puede ser reclamado para cumplir con la norma IEEE/EIA 12207.1-1997 cuando cada uno de los documentos que produce puede ser reclamado para cumplir con uno o más de los elementos de información de la Tabla 1.

#### **c) IEEE/EIA 12207.2-1997**

La Parte 2 es la guía para la implementación de los procesos del ciclo de vida del software definidos en la norma IEEE/EIA 12207.0-1996. En IEEE/EIA 12207.2, el texto normativo de la norma base ha sido actualizado mediante la incorporación de los cambios que figuran en el anexo J (Fe de erratas) de IEEE/EIA 12207.0-1996.

Las cláusulas 5, 6, 7 y los anexos A hasta E repiten el texto normativo de la norma base y añaden orientaciones para la aplicación de los temas seleccionados. El Anexo A proporciona orientación sobre el proceso de adaptación. El Anexo B ofrece orientación sobre el cumplimiento con los estándares IEEE/EIA 12207.0-1996. Los otros anexos proporcionan una guía adicional.

La conformidad y el cumplimiento no están definidos con respecto a esta guía.

La Figura 4.2 muestra el árbol genealógico de la Norma IEEE/EIA 12207.

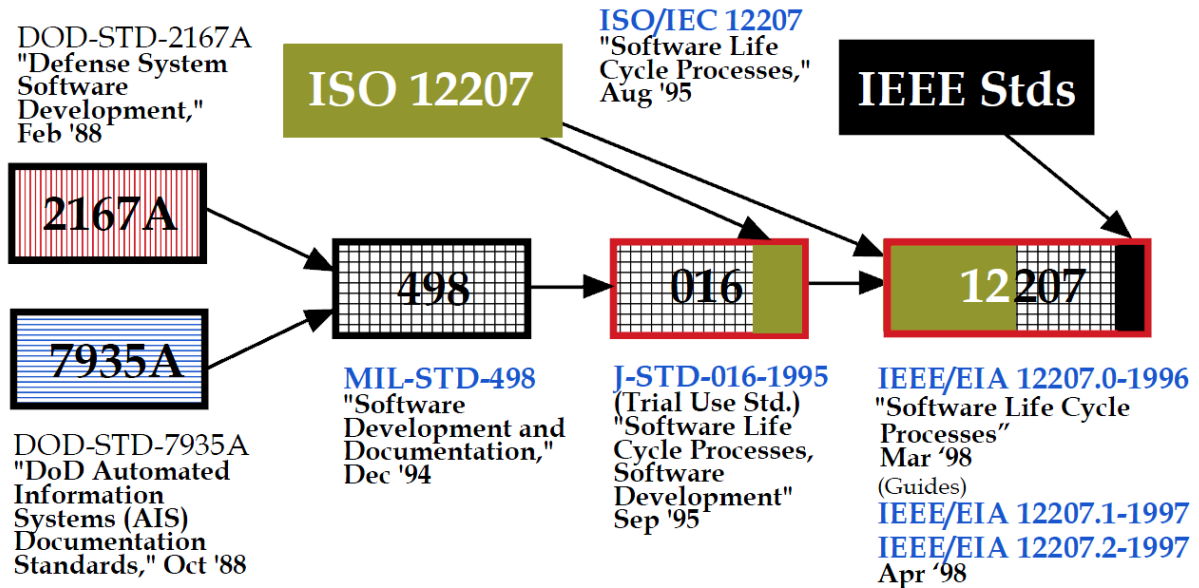


Figura 4.2: Árbol Genealógico de la Norma IEEE/EIA 12207.

### 3.5. Características Comunes de ISO/IEC 12207, EIA/IEEE J-STD-016 e IEEE/EIA12207

Dado que IEEE/EIA 12207 es una adaptación de la norma ISO/IEC 12207 y EIA/IEEE J-STD-016 e IEEE/EIA12207 fueron escritos por el mismo grupo de trabajo, no es extraño que las tres normas tengan muchas similitudes. Cabe señalar que la mayoría de las diferencias cruciales que distinguen a las normas ISO/IEC 12207, IEEE/EIA 12207 y J-STD-016 de otras normas relacionadas con el ciclo de vida del software, marcos, etc. son compartidas entre las tres normas. Estas diferencias se enumeran a continuación.

ISO/IEC 12207, IEEE/EIA 12207 y J-STD-016 se expresan en términos contractuales para facilitar su aplicación. Las tres normas tratan de evitar las limitaciones temporales. En parte, esto se logra alejándose de las grandes revisiones formales que tienden a provocar cuellos de botella en los proyectos.



Las normas son aplicables a proyectos de cualquier tamaño o nivel de complejidad, para software independiente o integrado, en cualquier sector del mercado (comercial, industria, defensa, etc.) Las tres normas pueden ser utilizadas en una organización o entre dos partes.

Las normas son compatibles con ISO 9001.

Las tres normas se concentran, en distintos grados, en qué hacer y no en cómo hacerlo (ISO/IEC 12207 muestra esta característica en mayor grado, seguida por IEEE/EIA 12207 y luego J-STD-016). En particular, las normas no exigen o prefieren un modelo de ciclo de vida en particular (esto las diferencia, por ejemplo, del DOD-STD-2167A, que se inclinaba hacia el modelo en cascada), método de diseño, lenguaje de programación, estándar de documentación, etc.

Se requiere o recomienda la especificación de indicadores de gestión (por ejemplo, los gastos) y atributos del software (tales como fiabilidad, mantenimiento, etc.) pero no se proporciona ningún detalle. En particular, no se señalan métricas o el uso de métricas.

## 4. Conclusión

Los beneficios que podemos obtener por implementar el uso de estándares se pueden clasificar en:

- Beneficios internos:
  - Reducción de costos.
  - Mejor diseño y calidad del producto.
  - Reducción de desechos.
  - Aumento de la productividad.
- Beneficios externos:
  - Aumento de la satisfacción del cliente.
  - Aumento de la porción de mercado.
  - Aumento de la confianza del cliente.

En el área del software, en particular, los beneficios principales que se pueden obtener por el uso de los estándares son:

- La capacidad de aplicar metodologías y procedimientos del más alto nivel profesional al desarrollo y el mantenimiento de software.
- Mejora del entendimiento y la coordinación entre los equipos de desarrollo, especialmente entre los equipos de desarrollo y mantenimiento.
- Mayor cooperación entre el desarrollador de software y los participantes externos en el proyecto.
- Mejora del entendimiento y la cooperación entre proveedores y clientes.

Estos beneficios, junto con la creciente complejidad y alcance de los proyectos de software, han llevado a una mayor aplicación de los estándares en la industria.

Por otro lado, cualquier empresa que implemente estándares en forma deficiente, por ejemplo solo en respuesta a la presión de sus clientes pero sin una convicción real, es muy poco probable que aproveche todo su potencial, y terminará perdiendo tiempo y recursos.

## 5. Bibliografía

- ASSC (Avionic Systems Standardisation Committee). Processing Working Group. *Emerging Commercial Software Development Standards (Doc No: ASSC/330/2/159)*. [en línea] c. 1999. [citada 25 de octubre de 2010]  
Disponible en Internet en:  
[http://assconline.co.uk/documents/ASSC\\_Emerging\\_Software\\_Development\\_Standards\\_Report.pdf](http://assconline.co.uk/documents/ASSC_Emerging_Software_Development_Standards_Report.pdf)
- Galín, Daniel. *Software Quality Assurance, From theory to Implementation*. Londres: Pearson Education Limited, 2004. 590 p. ISBN 0-201-70945-7.
- Gray, Lewis. *ISO/IEC 12207 Software Lifecycle Processes*. [en línea] c. 1996. [citada 25 de octubre de 2010]  
Disponible en Internet en:  
<http://www.stsc.hill.af.mil/crosstalk/1996/08/isoiec.asp>
- Gray, Lewis. *A Comparison of IEEE-EIA 12207, ISO-IEC 12207, J-STD-016 and, MIL-STD-498*. [presentación] [en línea] c. 1999. [citada 25 de octubre de 2010]  
Disponible en Internet en:  
[http://www.abelia.com/docs/122\\_016.pdf](http://www.abelia.com/docs/122_016.pdf)
- Moore, James W.; DeWeese, Perry R. y Rilling, Dennis. *U.S. Software Lifecycle Process Standards*. [en línea] c. 1997. [citada 25 de octubre de 2010]  
Disponible en Internet en:  
<http://www.stsc.hill.af.mil/crosstalk/1997/07/lifecycle.asp>
- Newberry, Maj. George A. *Changes from DOD-STD-2167A to MIL-STD-498*. [en línea] c. 1995. [citada 25 de octubre de 2010]  
Disponible en Internet en:  
<http://www.stsc.hill.af.mil/crosstalk/1995/04/Changes.asp>
- Perry, William J. *Specifications & Standards - A New Way of Doing Business*. [en línea] c. 1994. [citada 25 de octubre de 2010]  
Disponible en Internet en:  
<http://www.militec1.com/NewWay1.html>
- Pressman, Roger S. *Ingeniería del Software, Un enfoque Práctico*. 5º ed. Madrid: McGraw-Hill / Interamericana de España, 2002. 601 p. ISBN: 0-07-709677-0.
- Radatz, Jane; Olson, Myrna y Campbell, Stuart. *"MIL-STD-498" by Logicon, San Diego CA*. [en línea] c. 1995. [citada 25 de octubre de 2010]  
Disponible en Internet en:  
<http://www.stsc.hill.af.mil/crosstalk/1995/02/MILSTD.asp>

Schulmeyer, G. Gordon. *Handbook of Software Quality Assurance*. 4° ed. Norwood: Artech House, Inc. 2008. 464 p. ISBN-13: 978-1-59693-186-2.

Singh, Raghu. *International Standard ISO-IEC 12207 Software Life Cycle Processes*. [en línea] c. 1996. [citada 25 de octubre de 2010]

Disponible en Internet en:

<http://www.abelia.com/docs/12207cpt.pdf>

Singh, Raghu. *An Introduction to International Standard ISO-IEC 12207 Software Life Cycle Processes*. [presentación] [en línea] c. 1999. [citada 25 de octubre de 2010]

Disponible en Internet en:

<http://www.abelia.com/docs/12207tut.pdf>

Sommerville, Ian. *Ingeniería del Software*. 7° ed. Madrid: Pearson Educación S.A., 2005. 712 p. ISBN: 84-7829-074-5.

Sorensen, Reed. *MIL-STD-498, J-STD-016, and the U.S. Commercial Standard*. [en línea] c. 1996. [citada 25 de octubre de 2010]

Disponible en Internet en:

<http://www.stsc.hill.af.mil/crosstalk/1996/06/JSTD016.asp>

Sorensen, Reed. *Software Standards, Their Evolution and Current State*. [en línea] c. 1999. [citada 25 de octubre de 2010]

Disponible en Internet en:

<http://www.stsc.hill.af.mil/crosstalk/1999/12/sorensen.asp>