

CAPÍTULO 2. MODELOS Y ESTÁNDARES DE CALIDAD DE SOFTWARE

2.1 Ingeniería de Software

Los modelos y estándares de calidad de software forman parte de la ingeniería de software. Es por eso que comenzaremos con algunas definiciones de lo que es la Ingeniería de Software:

- Es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de Software, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido [FARLEY; 1988]
- Objetivo de las organizaciones fabricantes de software: producir software de buena calidad de una manera sistemática y previsible [FARLEY; 1988]
- Es la disciplina cuyo fin es la producción de software libre de fallas, entregado a tiempo, dentro del presupuesto y que satisfaga las necesidades del cliente [SCHACH; 1998]

Y por último la definición que nos da el IEEE:

- La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software, es decir la aplicación de ingeniería al software [Estándar IEEE 610.12]

Si unificamos lo anterior concluimos que la ingeniería de software es una disciplina que integra proceso, métodos y herramientas para el desarrollo del software de computadora.

La calidad es la base de todos ellos, como lo podemos observar en la figura 2.1

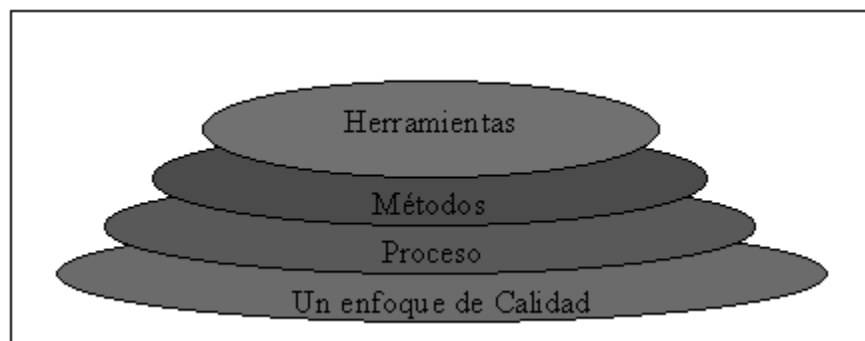


Figura 2.1 Capas de ingeniería de software [PRESSMAN;97]

El proceso es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería del software. Los métodos indican cómo construir técnicamente el software. Las herramientas proporcionan un soporte automático o semi-automático para el proceso y para los métodos [PRESSMAN; 98]

2.2 Por qué es importante Implementar Modelos y Estándares de Calidad de Software

Dado que la competencia cada día es más fuerte, es necesario que las empresas se preocupen en dar un mejor producto. Pero la calidad del producto no sólo se mide al terminarlo. La complejidad de los problemas que hoy en día buscan una solución en el software ha aumentado de manera considerable. Pero este crecimiento ha sobrepasado de sobremanera al aumento en la habilidad de desarrollar y mantener el software por parte de las organizaciones dedicadas a desarrollarlo o mantenerlo.

Enfrentamos una situación con dos caras. Por una parte las organizaciones quieren ser capaces de desarrollar y entregar software confiable, a tiempo y apegado al presupuesto acordado con el cliente. La segunda cara de la moneda nos muestra la perspectiva del cliente, el cuál quiere saber con certeza que todo lo anterior se cumplirá. Por esto las organizaciones deben buscar una norma, estándar o modelo que pueda ayudarlas a conseguir su meta de calidad (competitividad)

Sin embargo, la competitividad no es la única razón por la cuál se busque la calidad en el software. Debemos darle importancia a cada programa que se desarrolla. Debemos tomar conciencia y responsabilidad de las consecuencias que un defecto en nuestro producto podría ocasionar. Algunos defectos de software han ocasionado serios daños y hasta perjudicado físicamente a personas. Gente ha muerto debido a software defectuoso [LEVESON; 95]

El problema es que los sistemas cada vez son más rápidos, más complejos y automáticos. La posibilidad de una falla catastrófica aumenta a la par que el potencial del

daño que podría ocasionar [PERROW; 84] Así que debemos saber distinguir entre simple y fácil. Un error simple no necesariamente será fácil de encontrar, por tanto todos estamos involucrados en la calidad del producto, al ser responsables de la calidad de nuestro trabajo.

Otro aspecto negativo de los defectos es el económico. Cada defecto representa un costo adicional. Un error identificado en la misma fase donde se produjo es mucho más barato de resolver que el mismo defecto en una fase posterior, y aún más caro si éste sale a la luz después que el producto ya ha sido entregado.

Las siguientes razones son importantes para implementar un sistema de calidad:

- Satisfacción del cliente
- Competencia
- Defectos

2.3 Por qué puede fallar la implementación de software

La buena implementación no sólo involucra el seguir los puntos o requerimientos que cada uno de los modelos o estándares señalan. El tener un proceso y prácticas documentadas de nada sirven si no se siguen.

La norma por si sola no dará un avance si no existe un compromiso por parte de la alta gerencia. O más aún, si las prácticas no se ejercen por cada uno de los integrantes de la organización.

La alta gerencia juega un papel muy importante dado que su visión del sistema de calidad es la que se manifiesta a todos los empleados. Si la gerencia observa a la norma como algo requerido por los clientes y no como algo beneficioso, lo mismo ocurrirá con

el personal. La gerencia también es responsable de proporcionar los recursos necesarios para poder implementar el sistema de calidad. Debe existir un compromiso por parte de la gerencia en darle seguimiento y avance al sistema de calidad [MONTERO; 2000]

Para asegurar la buena implementación de cualquier norma o modelo se deben tomar en cuenta tres componentes:

- Las prácticas
- Las herramientas
- La gente

Las prácticas deben institucionalizarse. La gente debe de ser capaz y responsable de seguir cada una de las prácticas que están definidas para toda la organización. Para poder ayudar a la gente a dar seguimiento a las prácticas correspondientes se puede hacer uso de herramientas especializadas. Las herramientas harán que las personas no vean al proceso como algo hostil y fastidioso. Es necesario definir que es lo que se va a hacer, por quien y cuando. Otro punto importante es el ciclo de vida de los procesos. El hecho de haber definido, documentado, medido e institucionalizado los procesos no significan que sean los mejores. Todo proceso está sujeto a cambios. Tener un mal proceso que no evoluciona representa más un obstáculo que una ayuda.

Un último punto sería el enfoque con que se ve el proceso. Los procesos deben ayudarnos a lograr un objetivo de la organización más no son ellos mismos el objetivo. La burocratización es el resultado de ver al proceso como objetivo [HUMPHREY; 2000]

2.4 Algunos estándares

Debido al gran número de estándares y modelos de procesos que se aplican en la industria del desarrollo de software, el Consorcio de Productividad de Software (Software Productivity Consortium) realizó un estudio sobre éstos.

El estudio abarcó los marcos de trabajo relevantes para las organizaciones dedicadas a la construcción intensiva de sistemas de software (ver figura 2.2 y tabla 2.1)

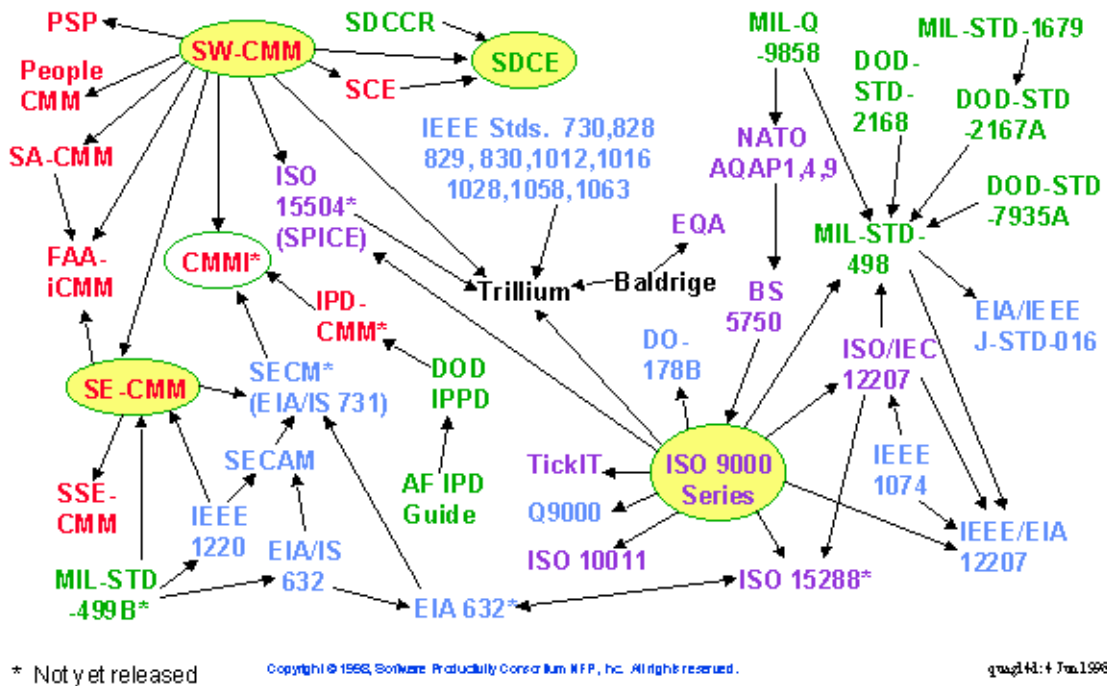


Figura 2.2 “El pantano de los Marcos de Trabajo” [SPC; 1998]

Tabla 2.1 Guía de colores para la figura 2.2

Color	Significado
Rojo	Un Modelo de Capacidad de Madurez
Verde	Un documento de gobierno o milicia de Estados Unidos
Morado	Un estándar Internacional
Azul	Un documento de una asociación industrial, profesional o comercial (en su mayoría de Estados Unidos)
Negro	Otro

[SPC; 1998]

El estudio realizado dividió a los modelos-estándares en seis grupos (tabla 2.2):

Tabla 2.2 Categorías de Marcos de Trabajo

1. Pautas y Estándares
2. Modelos de Mejoramiento de Procesos y Métodos de Evaluación Interna
3. Vehículos de selección de Contratistas
4. Premios de Calidad
5. Modelos de Ciclo de Vida de Ingeniería de Software
6. Modelos de Ingeniería de Sistemas

Elaboración Propia

La información obtenida del estudio se resume en la tabla 2.3:

Tabla 2.3 Modelos y Estándares de Software

Marco de Trabajo	Alcance	Propósito	Longitud (páginas)	Enfoque principal	Notas
CMM para Software	Organizaciones de desarrollo de Software	Mejora de procesos	500	Proceso de SW	La arquitectura por etapas provee un "Triptik" (un mapa AAA con calles, paradas y tiempos exactos) para mejorar.
SE-CMM	Organizaciones desarrolladoras de sistemas	Mejora de procesos	250	Proceso de Ingeniería de SW	Arquitectura continua que provee el mapa de un terreno.
IPD-CMM	Empresas	Mejora de procesos	220	Proceso	Arquitectura por etapas-continuas que provee un mapa además de un "Triptik"
ISO 9000	Organizaciones que producen Productos	Comercio	16	Proceso de Calidad	El registro certifica la conformidad de un sistema de calidad mínimo.
SDCE	Organización de Oferta (Bidding)	Selección de Contratistas	600	Proceso, Capacidad, Tecnología	Evalúa los riesgos de adquirir cada oferta, y reduce el riesgo con el contratista seleccionado.
Estándares de Ciclos de Vida de Software	Organizaciones de desarrollo de Software	Conformidad de Contrato	60-200	Proceso de Administración	Los estándares evolucionando para incluir el rol del comprador y otros, al igual que el del que suministra.
Trillium	Empresa	Mejora de procesos	130	Proceso	Combina requerimientos de CMM, ISO, Baldrige, y estándares de calidad de software.

[SPC; 1998]

Podemos observar el enfoque de cada uno de los marcos de trabajo. Quizá algunos demasiado específicos. Muchos son ramificaciones de otros y su contenido es en esencia el mismo. Como organización dedicada al desarrollo de software será necesario

evaluar las distintas opciones en cuanto normas, estándares y modelos, pero primero necesitaremos definir cuál será nuestro objetivo a seguir.