

CC51A – Ingeniería de Software

El Modelo de Capacidad de Madurez (CMM) en una Empresa

Sergio Ochoa D.

1

Estructura de la Presentación

- ◆ El "Software Engineering Institute" (SEI).
- ◆ Modelo de Madurez del SEI.
- ◆ Modelo de Capacidad de la Madurez (CMM).
- ◆ Estado de la práctica.
- ◆ Proyecciones.
- ◆ Conclusiones.
- ◆ Bibliografía y casos.

2

Discusión

¿En un proyecto de software, cuál es la relación entre:

- ♦ calidad del software,
- ♦ costo del proyecto, y
- ♦ tiempo de desarrollo del proyecto?

¿Cuán estable es esta relación?

¿De qué depende la estabilidad de la relación?

3

Software Engineering Institute

- ◆ Financiado por el Departamento de Defensa de EEUU, en *Carnegie Mellon University*, en 1984 (<http://www.sei.cmu.edu>).
- ◆ Primer director: Watts Humphrey.
- ◆ **Objetivo:** Establecer estándares de excelencia en ingeniería de software y acelerar la transición de tecnología y métodos de avanzada, a la práctica.

4

Software Engineering Institute

- ◆ **Resultado importante:** *Modelo de Madurez del Proceso de Software* - desarrollado para evaluar las capacidades de una organización de software e identificar las áreas más importantes de mejoramiento - tratando el proceso completo de desarrollo de software como un proceso que puede ser controlado, medido y mejorado.

5

Software Engineering Institute

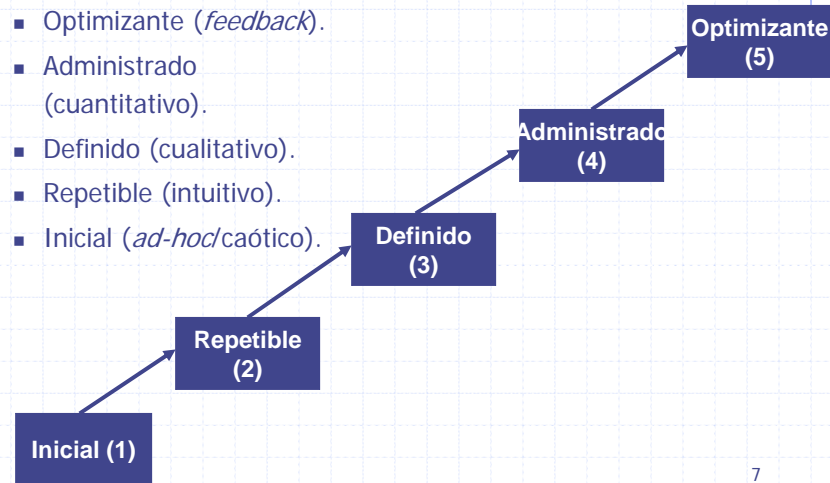
- ◆ Para mejorar sus capacidades, las organizaciones de software deben:
 - comprender el *estado actual* de sus procesos de software;
 - desarrollar una *visión de los procesos deseados*;
 - establecer una lista de las *acciones de mejoramiento* requeridas en orden de *prioridad*;
 - producir *un plan* para cumplir dichas acciones;
 - *comprometer los recursos* para ejecutar el plan;
 - *ejecutar* el plan trazado; y
 - *reevaluar* la situación actual (se vuelve al primer punto).
- ◆ **Herramienta:** *framework* de madurez CMM v0.

6

Modelo de madurez CMM

5 niveles de CMM v0:

- Optimizante (*feedback*).
- Administrado (cuantitativo).
- Definido (cualitativo).
- Repetible (intuitivo).
- Inicial (*ad-hoc*/caótico).



7

Modelo de madurez del SEI

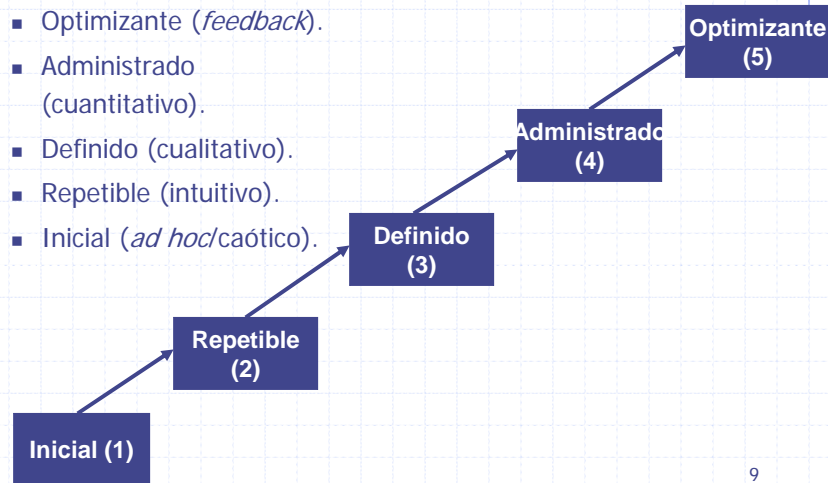
- ◆ Modelo en 5 niveles (presentado en 1987).
- ◆ Cada nivel establece un conjunto intermedio de objetivos para alcanzar niveles más altos de madurez.
- ◆ "Los 5 niveles de madurez representan razonablemente las fases históricas de mejoramiento evolutivo de organizaciones de software reales, representan una medida de mejoramiento que es razonable alcanzar desde el nivel previo, sugieren objetivos de mejoramiento y mediciones de avance intermedios, y hacen obvio un conjunto de prioridades de mejoramiento inmediato una vez que el status de una organización con respecto al *framework* es conocido" (Watts Humphrey).

8

Modelo de madurez del SEI

5 niveles de CMM v0:

- Optimizante (*feedback*).
- Administrado (cuantitativo).
- Definido (cualitativo).
- Repetible (intuitivo).
- Inicial (*ad hoc*/caótico).



9

Modelo de madurez del SEI

Nivel inicial (1): Caótico/Ad-Hoc

- Carencia de procedimientos formales, estimaciones de costo, planes de proyecto, mecanismos de administración para asegurar que los procedimientos se siguen, herramientas mal integradas, control del cambio inexistente, no se administran riesgos, administración superior no comprende temas claves.

10

Modelo de madurez del SEI

Nivel repetible (2):

- Proceso dependiente de los individuos, se establecen controles básicos de los proyectos, fortalezas cuando se desarrollan trabajos similares, gran riesgo cuando se enfrentan nuevos desafíos, cambios impactan fuertemente en la ejecución de un proyecto, carece de un *framework* ordenado para mejorar.

11

Modelo de madurez del SEI

Nivel definido (3):

- Proceso de Desarrollo está definido (escrito) e institucionalizado (todos deben adherir a él)
- Existe un Grupo de Proceso de Ingeniería de Software (SEPG) establecido para liderar el mejoramiento.

Nivel administrado (4):

- Proceso es completamente medido.
- Se establece un mínimo conjunto de mediciones de calidad y productividad, y una base de datos de resultados, cuyo análisis es usado para la mejora.¹²

Modelo de madurez del SEI

Nivel optimizante (5):

- Los mejoramientos propuestos retroalimentan al proceso
- La recolección de datos es automatizada y usada para identificar elementos más débiles del proceso
- Hay evidencia numérica se usa para justificar la aplicación de tecnología a las tareas críticas
- Hay un riguroso análisis de causas y prevención de defectos.

13

Modelo de madurez del SEI

Procedimiento de evaluación:

- Se revisa y evalúa el proceso de software de una organización
- La evaluación es hecha por un equipo de profesionales certificados, *on-site*, durante una semana aproximadamente.
- Se evalúan proyecto de no más de 2 años de antigüedad.

Herramienta principal: cuestionario de 101 preguntas del tipo si/no.

Resultado:

- se indica en qué punto de la escala exactamente se encuentra la organización evaluada;
- se entregan key issues a resolver para escalar hacia el próximo nivel.

14

Modelo de madurez del SEI

Principales críticas a CMM v0:

- Herramienta para evaluar contratistas de defensa.
- Problemas metodológicos y estadísticos en la determinación del *score* (nivel de madurez).
- Reportes finales no suficientemente detallados.
- Nivel 1 se alcanza sin esfuerzo.
- Complejidad y precisión del algoritmo de evaluación.
- Falta de flexibilidad en la determinación del *score*.
- Respuestas si/no.
- Falta de claridad en niveles 4 y 5.

15

Modelo de madurez del SEI

Sugerencias:

- modificar el método de evaluación.
- desenfatar el nivel de madurez (*score*).

Respuesta:

- nuevo modelo: CMM v1 (Capability Maturity Model), en 1991.
- Descompone cada nivel de madurez en áreas claves de proceso (KPA), prácticas claves, e indicadores claves.

16

Capability Maturity Model (CMM)

Áreas claves de proceso (KPAs): identifican objetivos a ser alcanzados para alcanzar un nivel de madurez particular.

Prácticas claves: procedimientos y actividades que contribuyen a alcanzar los objetivos.

Indicadores claves: ayudan a determinar el cumplimiento de los objetivos, forman la base para el procedimiento de evaluación.

17

Capability Maturity Model (CMM)

- ◆ Desenfatisa el *score* (nivel de madurez) de una evaluación.
- ◆ El producto final es ahora un *perfil* de áreas claves, que pueden ser satisfechas parcial o completamente, o no ser satisfechas.
- ◆ El nivel de madurez se establece como aquel en que se satisfacen todas las áreas claves en forma continua.

18

Capability Maturity Model (CMM)

CMM versión 1.0: *release* inicial, durante 1991 y 1992.

CMM versión 1.1: 1993-2003.

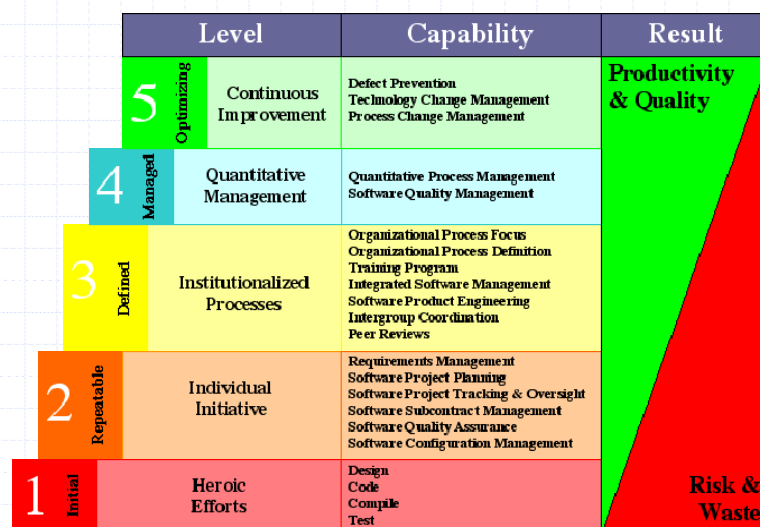
CMM versión 2: liberado en 1996 (pero rechazado).

CMMI versión 1: liberado en 2003 (aceptado por el DoD).

Framework de madurez no ha cambiado demasiado, las nuevas versiones mejoran la estructura de CMM a través del uso de áreas claves de proceso (KPAs), prácticas e indicadores claves, cuestionarios mejorados y separación de Temas (issues).

19

Capability Maturity Model (CMM) v1.1



Capability Maturity Model (CMM) v1.1

Nivel Inicial (1) - áreas claves de proceso:

- ninguna

21

Capability Maturity Model (CMM) v1.1

Nivel Repetible (2) - áreas claves de proceso:

- Gestión de requisitos.
- Planificación de proyectos de software.
- Supervisión y seguimiento de proyectos de software.
- Gestión de subcontratos de software.
- Aseguramiento de calidad de software.
- Gestión de la configuración de software.

22

Capability Maturity Model (CMM) v1.1

Nivel Definido (3) - áreas claves de proceso:

- Foco en el proceso de la organización.
- Definición del proceso de la organización.
- Programa de entrenamiento.
- Administración a través de un software integrado.
- Ingeniería del producto de software.
- Coordinación intergrupos.
- Revisión por pares.

23

Capability Maturity Model (CMM) v1.1

Nivel Administrado (4) - áreas claves de proceso:

- Administración cuantitativa del proceso.
- Administración de calidad de software.

Nivel Optimizante (5) - áreas claves de proceso:

- Prevención de defectos.
- Administración de cambios tecnológicos.
- Administración de cambios en el proceso.

24

Capability Maturity Model (CMM) v1.1

Nivel	Características	Desafíos claves	Resultados
5 Optimizante	Mejoramiento realimentado al proceso	Aún un proceso humano-intensivo Mantiene la organización en nivel optimizante	Productiv. y calidad
4 Administrado	(Cualitativo) Proceso medido	Cambio de tecnología Análisis de procesos Prevención de problemas	
3 Definido	(Cualitativo) Proceso definido e institucionalizado	Métricas de procesos Análisis de procesos Planes cuantitativos de calidad	
2 Repetible	(Intuitivo) Proceso dependiente de individuos	Entrenamiento, testing Prácticas técnicas y revisiones Foco en el proceso, estándares y procesos	
1 Inicial	(Ad hoc/caótico)	Administración de proyectos y planificación Administración de la configuración Aseguramiento de la calidad de software	Riesgo

25

Capability Maturity Model (CMM) v1.1

- ◆ Se ha transformado en un estándar *de facto* en la industria de software norteamericana.
- ◆ Ha enfatizado la identificación de problemas y acciones de mejoramiento por sobre el *score* numérico.
- ◆ Carece de validación concluyente; datos preliminares son favorables.

26

Estado de la práctica

- ◆ Estudio del SEI, 1991:
 - 59 sitios, 296 proyectos
 - resultados:
 - ◆ 81% al Nivel 1 CMM.
 - ◆ 12% al Nivel 2 CMM.
 - ◆ 7% al Nivel 3 CMM.
 - ◆ 0% al Nivel 4 CMM.
 - ◆ 0% al Nivel 5 CMM.
- ◆ Estado de la práctica de ingeniería de software: fundamentalmente conducida por crisis.

27

Estado de la práctica

- ◆ Estudio del SEI, 1995:
 - 440 sitios, 2200 proyectos.
 - Resultados:
 - ◆ 70.2% al Nivel 1 CMM.
 - ◆ 18.4% al Nivel 2 CMM.
 - ◆ 10.2% al Nivel 3 CMM.
 - ◆ 1.0% al Nivel 4 CMM.
 - ◆ 0.2% al Nivel 5 CMM.

28

Estado de la práctica

◆ Estudio del SEI, 1996:

- 477 sitios.
- Resultados:
 - ◆ 69% al Nivel 1 CMM.
 - ◆ 18% al Nivel 2 CMM.
 - ◆ 11% al Nivel 3 CMM.
 - ◆ 1.5% al Nivel 4 CMM.
 - ◆ 0.4% al Nivel 5 CMM.

29

Estado de la práctica

◆ Tiempos promedio para alcanzar cada nivel CMM:

- Nivel 1 a nivel 2: 2 a 3 años.
- Nivel 2 a nivel 3: 1.5 a 2.5 años.
- Nivel 3 a nivel 4: ?
- Nivel 4 a nivel 5: ?

◆ Basado en datos del SEI (1995).

- ◆ Según comentan los expertos, con CMMI se puede pasar de un nivel a otro en alrededor de 18 meses.

30

Proyecciones CMM

- ◆ EEUU: nivel 3 mínimo establecido por el Departamento de Defensa para sus contratistas para 1997/8/9.
- ◆ Relación CMM - ISO 9000: ¿CMM nivel 3 implica certificación ISO? ¿y viceversa?
- ◆ Importancia e implicancias para Chile.

31

Familia CMM

- ◆ Software CMM (SW-CMM): construcción de productos de software.
- ◆ People CMM (P-CMM): desarrollo del recurso humano.
- ◆ Software Acquisition CMM (SA-CMM): adquisición de productos y sistemas intensivos en software.
- ◆ Trusted CMM (T-CMM): construcción de productos de software de alta confiabilidad.
- ◆ Systems Engineering CMM (SE-CMM): construcción de productos y entrega de servicios (foco: ingeniería de sistemas).
- ◆ Integrated Product Development CMM (iPD-CMM): construcción y entrega de productos y servicios integrados (foco: colaboración entre varias disciplinas).

32

Normas ISO 9000

- ◆ Desde 1987, importantes por exigencias de certificación (principalmente en Europa).
- ◆ ISO 9001/9000-3, relevante para software.
- ◆ Énfasis en requisitos mínimos para un sistema de calidad, poco soporte explícito para mejoramiento continuo.
- ◆ En Chile, norma chilena Nch-ISO 9001/9000-3, oficial desde Enero 1995.
- ◆ Desarrollado por auditores certificados.

33

Norma ISO 9126

- ◆ Estándar *reciente* para medir calidad
- ◆ 6 características básicas, descompuestas en subcaracterísticas
- ◆ Refleja una visión de usuario, mientras que otros modelos (como el de McCall) reflejan una visión de producto.

34

Norma ISO 9126

- ◆ Funcionalidad -- adecuación, precisión, interoperabilidad, seguridad
- ◆ Confiabilidad - madurez, tolerancia a fallas, recuperación de fallas
- ◆ Usabilidad -- facilidad de comprensión, facilidad de aprendizaje, facilidad de operación

35

Norma ISO 9126

- ◆ Eficiencia -- comportamiento temporal, comportamiento de recursos
- ◆ Facilidad de mantención -- facilidad de análisis, facilidad de cambio, estabilidad, facilidad de prueba
- ◆ Portabilidad -- facilidad de instalación, conformidad, facilidad de reemplazo

36

PSP (Personal Software Process)

- ◆ Desarrollado por Watts Humphrey, SEI, 1994.
- ◆ Representa una síntesis elegante de conceptos probados (CMM, experience factory, inspecciones, análisis causales) *aterrizados* al nivel individual.
- ◆ Incorpora principios de calidad en procesos definidos que los ingenieros pueden seguir personalmente.
- ◆ Consiste en 7 pasos o procesos, que permiten al ingeniero recolectar y analizar sus propios datos para definir y utilizar procesos que pueden ajustarse a sus propias necesidades.

37

SPICE

- ◆ Aún un proyecto (ISO/IEC JTC1/SC7 WG10).
- ◆ **Objetivo:** desarrollar un estándar internacional para evaluaciones de procesos de software.
- ◆ **Orígenes:** CMM, ISO, y otros productos.
- ◆ Relaciona proceso, evaluación de proceso, determinación de capacidades, y mejoramiento de procesos.
- ◆ Consta de 9 productos en la forma de guías y cuestionarios.
- ◆ Desde 1995 se han desarrollado 3 fases de prueba.

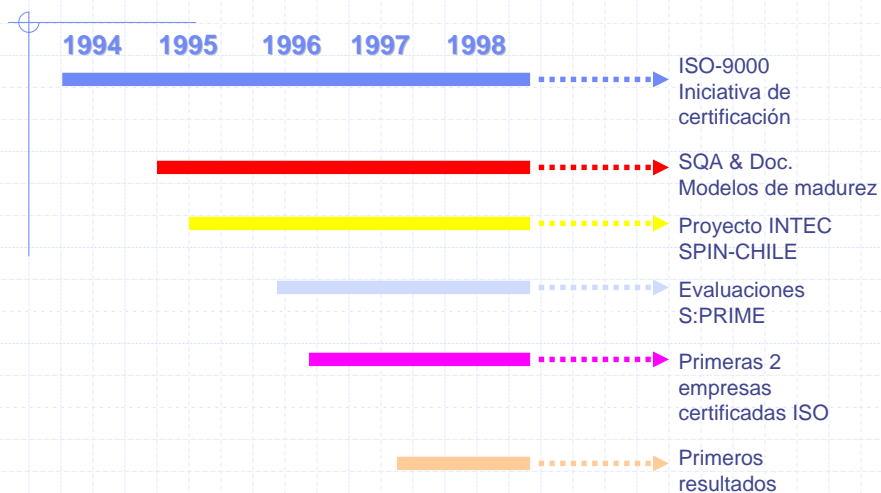
38

S:PRIME

- ◆ Desarrollado por ASEC (CGLA), CRIM, Canadá; basado en CMM, focalizado hacia organizaciones medianas y pequeñas.
- ◆ Consiste en administrar dos cuestionarios: uno para gerentes (7 categorías de riesgos), y otro para *practitioners* (15 áreas de prácticas), para determinar los grados de riesgos percibidos y probables, de acuerdo a la satisfacción de las prácticas.
- ◆ Pasos del método: adaptación de cuestionarios, selección de muestra de proyectos, respuesta a los cuestionarios, análisis y compilación de respuestas, presentación y discusión de resultados.
- ◆ Desarrollado por *lead assessors* certificados por CRIM, S:PRIME requiere un número significativamente menor de horas hombre, comparado con CBA/IPI.
- ◆ Se ha empleado para evaluar compañías en Chile.

39

¿Qué pasa en Chile?



40

Evaluaciones S:PRIME, la experiencia chilena

- ◆ 8 organizaciones.
- ◆ 38 proyectos.
- ◆ 137 participantes.
- ◆ 57 cuestionarios de riesgo contestados.
- ◆ 99 cuestionarios de prácticas contestados.

41

Conclusiones

- ◆ Se han visto resultados positivos.
- ◆ Clara inmadurez de procesos en las organizaciones.
- ◆ Creciente interés por mejorar.
- ◆ Necesidad por generar nuevas competencias y habilidades.
- ◆ Organizaciones pequeñas carecen de recursos.
- ◆ Todavía no hay un incentivo real desde los clientes.....
- ◆ Es parte de la agenda digital

42

Lecciones importantes...

- ◆ Las organizaciones necesitan instrumentos concretos para facilitar la implantación de prácticas y no quedarse en las buenas intenciones de los planes de acción.

43

Lecciones importantes...

Cuidado con los Gurús: Ingeniero muy talentoso lleva a un pequeño grupo a producir un extraordinario programa.

- El cree que después de esto puede desarrollar cualquier cosa.
- Acepta cualquier desafío con suprema confianza.
- Al exigirle un plan se rehusa y apunta a su previo éxito para justificar.
- Nuevamente es exitoso => se le asigna proyecto mas grande y complejo.
- Llega el momento que recibe el proyecto que está mas allá de su capacidad.
- No hay recuperación posible: los gurús llevan el proyecto en su cabeza (muy poco documentado).

44

Lecciones importantes...

En computación, la *magia* no existe !!!

- De algún modo encontraremos una solución elegante y sencilla a toda esta confusión.
- Tal vez nuevos lenguajes, ambientes, herramientas, etc.? Fred Brooks => "No silver bullet".
- A veces la tecnología puede ayudar, pero generalmente el problema pasa por la administración (planificación, control y seguimiento).
- Cuando la gente cree en magia los detalles parecen innecesarios y el trabajo difícil es diferido hasta que la catástrofe aparece.

45

Lecciones importantes...

Tener MUY en cuenta la *escala*

- La gente comienza sus carreras escribiendo programas pequeños.
- Ellos saben como escribir programas y no entienden mucho esto de la "crisis del software".
- Al ser asignados a un proyecto grande no tienen dificultades en completar sus propios módulos.
- La frustración aparece al tener que coordinarse con mucha gente.
- Todo tiene un impacto en todo.
- Invierte buena parte del tiempo reaccionando a una lista interminable de pedidos de cambio.

46

Siglas

CMU	Carnegie Mellon University
SEPG	Grupo de Proceso de Ingeniería de Software
SEI	Software Engineering Institute
KPA	Key Process Area
SPI	Software Process Improvement

47

Bibliografía y Casos

- ◆ SEI/CMU. URL: <http://www.sei.cmu.edu>
- ◆ CMM Implementation Guide. Kim Caputo. Addison-Wesley, 1998.
- ◆ An Integrated Collection of Software Engineering Standards. James W. Moore. IEEE Software, Nov.-Dic. 1999.

48