

# Comparación entre las Arquitecturas para la Definición de Procesos de Software de los Modelos CMM, CMMI, el Estándar ISO 15504 y el Metamodelo SPEM, en apoyo al Análisis de la Capacidad de los Procesos de Software.

José Luis Torres Pérez

Instituto de Investigación y Desarrollo de Ingeniería de Software  
Universidad Autónoma de Guadalajara  
Ave. Patria 1201, Zapopan, Jalisco, México, 44100. México  
Tel (013)-36-48-88-24 ext 6003, e-mail jtorres@cu.gdl.uag.mx

## RESUMEN

Aunque actualmente existen varias propuestas de Arquitecturas para la Definición de Procesos de Software, no presentan de manera explícita los elementos necesarios para poder realizar la medición y control estadístico de los mismos. El presente trabajo muestra las arquitecturas para la definición de procesos de software de los modelos CMM, CMMI, del Estándar ISO 15504 y del metamodelo SPEM. Se expone un método de análisis de la capacidad de un proceso estable que presenta una distribución normal estándar con el objetivo de identificar los elementos de apoyo al control estadístico que ofrece cada una de las propuestas, y en base a ello, presentar los elementos mínimos necesarios para realizar este control.

### Palabras Clave:

Análisis de Capacidad, Arquitectura de Procesos, Procesos de Software.

## I. INTRODUCCIÓN.

La complejidad de los problemas que hoy en día buscan una solución en el software ha aumentado de manera considerable. Este crecimiento sobrepasa la habilidad de desarrollar y mantener el software por parte de las organizaciones dedicadas a ello.

Las empresas desarrolladoras de software están evolucionando hacia la cultura de excelencia en la ingeniería y la administración del software. Actualmente, los dirigentes de las organizaciones quieren desarrollar y entregar software confiable apegado al presupuesto y tiempo acordado con el cliente. Por esta razón, ya no sólo basta medir la calidad del producto terminado, sino también, medir la calidad del proceso en el que se desarrolla tal producto. Esto motivó a varias organizaciones a desarrollar distintas propuestas que tuvieran como objetivo ofrecer modelos o estándares que llevaran a las

organizaciones desarrolladoras de software a un proceso de mejora.

## II. ESTADO DEL ARTE.

Entre las principales propuestas existentes podemos destacar al Capability Maturity Model (CMM) [1], presentado por el Instituto de Ingeniería de Software (SEI) en 1991; al Estándar ISO 15504 [2] (anteriormente conocido como SPICE) presentado formalmente en 1998 y al Capability Maturity Model Integration (CMMI) [3], también presentado por el SEI en el año 2000.

El SEI en su Modelo CMM no ofrece una guía o recomendación explícita para la definición de procesos, pero en su reporte técnico titulado "A software Process Framework for the SEI Capability Maturity Model: Repeatable Level" [4] describe 15 puntos que debe contener toda definición de proceso que desee cubrir lo requerido por el modelo de CMM versión 1.1. En su Modelo CMMI, tampoco ofrece una guía explícita para la definición de los procesos de software, pero en su Área Clave de Proceso denominada "Definición de los Procesos de la Organización" muestra los elementos que debe estar presentes en tal definición. Por su parte ISO en su estándar ISO / IEC 15504 sí incluye explícitamente una guía para la definición de procesos, esta guía se describe en la segunda parte del documento marcado como "Anexo C".

Existe otra propuesta para la definición de procesos de software, pero ésta no se encuentra relacionada con modelos o estándares internacionales de mejora de procesos, sino que surge como una iniciativa para atacar el problema desde una perspectiva orientada a objetos. Es una propuesta de la OMG (Object Management Group) presentada en Noviembre del año 2001 que tiene por nombre: Software Process Engineering Metamodel (SPEM) [5] y cuyo objetivo es describir procesos o familia de procesos para el desarrollo de software.

### III. METODOLOGÍA UTILIZADA

La realización de la Investigación consistió en tres pasos. Primero se identificaron los elementos que conforman a cada una de las Arquitecturas, posteriormente se buscaron los métodos de Análisis de la Capacidad de un Proceso de Software [6] y finalmente se realizó un análisis para establecer el apoyo que ofrecen dichas Arquitecturas al Análisis de la Capacidad.

#### 3.1 Estructura de las Arquitecturas

A continuación se presenta una tabla comparativa que presenta los distintos elementos que han sido considerados como parte de un proceso de software por los trabajos anteriormente mencionados.

Elemento del Proceso	CMM	ISO 15504	SPEM	CMMI
Actividad	x		x	
Capacitación	x			
Ciclo de Vida			x	x
Contexto	x			
Criterio de Entrada	x			x
Criterio de Salida	x			x
Dueño del Proceso				x
Entrada	x			x
Fase			x	
Flujo	x			
Guía				x
Herramientas	x			
Identificador del Proceso		x		
Iteración			x	
Estándares del Producto				x
Mediciones al Proceso	x			x
Mediciones al Producto				
Meta		x	x	
Nombre del Proceso		x		
Notas informativas				
Parámetros de la Actividad			x	
Pasos			x	
Precondiciones del Proceso			x	
Procedimientos Documentados	x			

Elemento del Proceso	CMM	ISO 15504	SPEM	CMMI
Producto de Trabajo	x		x	x
Propósito del Proceso	x	x		
Revisiones y Auditorías	x			x
Rol	x		x	x
Salida	x	x		x
Tipo de Proceso				
Interfaces				x
Relaciones entre elementos internos				x
Metodología para la realización de "Peer Reviews"				x
Plantilla para conducir revisiones administrativas				x

Tabla 1. Tabla comparativa de los elementos presentes en las Arquitecturas para la Definición de Procesos de Software.

#### 3.2 Análisis de la Capacidad de un Proceso

El análisis de la Capacidad es un conjunto de cálculos estadísticos realizados a un conjunto de datos generados de un proceso estable, con el fin de determinar la capacidad del proceso o sistema. La Capacidad de un proceso indica el rango de resultados esperados que pueden ser alcanzados siguiendo dicho proceso. Las especificaciones de los límites son fijados por el cliente, la administración o la gente del área de ingeniería. Consiste en dos etapas, primeramente debe determinarse la capacidad del proceso; después, deberá comprobarse la capacidad del proceso con la especificación establecida por el cliente o la necesidad de negocio.

Para determinar la capacidad del proceso se calculan los límites naturales del proceso. Para ello se deben seguir los siguientes pasos:

- Se construye un Histograma de los datos
- Se calcula la Desviación Estándar
  - Si el análisis de la capacidad se está realizando desde un diagrama de Control, la desviación estándar ( $\sigma_x$ ) será:

$$\text{Sigma}_x = R / d_2 \quad (1)$$

- o Si el Análisis de la Capacidad se está realizando desde un histograma, la desviación se calcula en 2 pasos. Primero se obtiene la Media ( $\bar{X}$ ),

$$\bar{X} = (X_1 + X_2 + \dots + X_n) / n \quad (2)$$

Y la desviación estándar ( $\text{Sigma}_x$ ) será:

$$\text{Sigma}_x = \sqrt{\left( \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1} \right)} \quad (3)$$

- El límite natural superior del proceso (LS) será:

$$\text{LS} = \bar{X} + 3 * \text{Sigma}_x \quad (4)$$

- El límite natural inferior del proceso (LI) será:

$$\text{LI} = \bar{X} - 3 * \text{Sigma}_x \quad (5)$$

Para comparar la capacidad del proceso con la especificación es necesario calcular la Tolerancia de la Especificación, así, se conocerá la distancia que existe entre los límites de la misma. Para lograr este objetivo, se requiere saber El Límite Superior Especificado (LSE) y el Límite Inferior Especificado (LIE), para posteriormente aplicar la siguiente fórmula:

$$\text{Tolerancia} = (\text{LSE} - \text{LIE}) / \text{Sigma}_x \quad (6)$$

Los porcentajes del proceso fuera de las especificaciones, requieren del valor Z para cada especificación, este valor está dado por:

- El valor Z del Límite superior ( $Z_{LS}$ ) que se calcula:

$$Z_{LS} = (\text{LSE} - \bar{X}) / \text{Sigma}_x \quad (7)$$

- El valor Z del Límite Inferior ( $Z_{LI}$ ) se calcula:

$$Z_{LI} = (\bar{X} - \text{LIE}) / \text{Sigma}_x \quad (8)$$

- o El porcentaje fuera de la especificación será la suma de los valores calculados de  $Z_{LS}$  y  $Z_{LI}$

Existen índices que pueden ser calculados a partir de la información de la capacidad del proceso, los más empleados son los índices  $C_p$ ,  $C_{pk}$  y  $C_r$ . El índice  $C_p$  indica la capacidad del sistema para cumplir con los límites de la especificación. El índice  $C_{pk}$  indica el porcentaje del proceso en la generación de

salidas dentro de los límites de la especificación. El índice  $C_r$  indica el porcentaje de la especificación utilizada por el proceso. Para calcular esto límites se utilizan las siguientes operaciones:

$$C_p = (\text{LSE} - \text{LIE}) / (6 * \text{Sigma}_x) \quad (9)$$

$$C_{pk} = (Z_{\min}) / 3 \quad (10)$$

$$C_r = 1 / (C_p) \text{ ó } C_p = (6 * \text{Sigma}_x) / (\text{LSE} - \text{LIE}) \quad (11)$$

$Z_{\min}$  es el valor menor obtenido de la tabla de la Distribución Normal Estándar, en base a los resultados de  $Z_{LS}$  y de  $Z_{LI}$ .

#### IV. RESULTADOS.

A pesar del valor incuestionable de cada una de estas propuestas para la mejora de procesos en las empresas desarrolladoras de software, estas arquitecturas para la definición de procesos no presentan de manera clara los elementos suficientes que permitan realizar un control estadístico de los mismos.

Para poder realizar el Análisis de la Capacidad de un Proceso de Software, es necesario conocer el límite inferior y el límite superior especificados para el proceso. Además se tiene que establecer la periodicidad para la recolección de datos, ya sea mediante el establecimiento de fechas o de eventos disparadores. Ninguna de las Arquitecturas analizadas menciona de manera explícita alguno de estos elementos, los modelos CMM y CMMI presentan la necesidad de establecer mediciones al proceso, adicionalmente el modelo CMMI recomienda establecer mediciones al producto.

Tanto el Estándar ISO 15504 como el metamodelo SPEM no presentan de manera explícita la necesidad de establecer mediciones en la definición del proceso. Además es importante establecer el o los atributos que se medirán en el proceso, así como la herramienta de medición que se utilizará y ninguno de estos elementos se presenta de manera directa en las Arquitecturas anteriores.

#### V. CONCLUSIONES Y TRABAJOS FUTUROS DE INVESTIGACIÓN.

Las distintas propuestas requieren de la definición de procesos de software, debido a la necesidad que tienen los administradores de las empresas de coordinar las actividades de un grupo de

personas que intervienen en un mismo proyecto de software. Un proceso definido garantizará que todos los miembros del equipo de desarrollo sepan qué hacer, el qué momento se debe hacer y el modo en que se debe hacer, por consiguiente, es válido decir que un proceso de software definido puede ser visto como un marco de referencia para la realización de las tareas de desarrollo de software que a su vez, sirve para la prevención de eventualidades y como un medio para la transmisión de conocimiento. La administración de estos procesos permitirá a la empresa conocer sus fortalezas y debilidades en la forma de realizar el proceso, sirviendo como base para su mejora.

Será importante para las empresas desarrolladoras de software contar con una Arquitectura para la Definición de Procesos de Software que ofrezca las diferentes ventajas de las iniciativas expuestas, así como la incorporación de los elementos para la medición y control estadístico del proceso anteriormente identificados, logrando con ello una Arquitectura clara y precisa que contemple todas las dimensiones de un proceso de software.

Agradecimientos. Quisiera agradecer a los revisores por sus oportunos y pertinentes comentarios, y al Instituto de Investigación y Desarrollo de Ingeniería de Software (IIDIS) de la Universidad Autónoma de Guadalajara por el apoyo brindado para el desarrollo de esta investigación.

## VI. REFERENCIAS.

- [1] Mark C.Paulk, Bill Curtis, Mary Beth Chrissis, and Charles V.Weber. Capability Maturity Model for Software, Version 1.1.CMU/SEI-93-TR-024. 1993. Software Engineering Institute.
- [2] ISO. ISO / IEC 15504-2-Anexo C:1998. 1-11-1998. ISO.
- [3] Software Engineering Institute. Capability Maturity Model Integration (CMMI),Version 1.1. CMU/SEI-2002-TR-011. 1-3-2002. Software Engineering Institute.
- [4] Timothy G.Olson, Neal R.Reizer, and James W.Over. A Software Process Framework for the SEI Capability Maturity Model. 1-9-1994. Software Engineering Institute.
- [5] OMG. Software Process Engineering Metamodel Specification. 1-11-2001. Object Management Group.
- [6] Norman Fenton and Shari Lawrence, Software Metrics- A Rigorous & Practical Approach, 1st ed. PWS Publishing Company, 1997.