
Modelo para la estandarización y clasificación de métricas del proceso de desarrollo de SW y definición de un ambiente de apoyo automatizado.

**Ricardo A. Gacitúa B.
Marcela Muñoz C.
Universidad del Bío – Bío
Concepción – Chile
e-mail: rgacitua@rayen.face.ubiobio.cl**

ABSTRACT.

Innumerable es la literatura acerca de métricas de software. Sin embargo, innumerables también son las dudas que se presentan al querer utilizarlas. Muy pocas son las referencias y recomendaciones de cuáles son las métricas que deben ser utilizadas en alguna etapa específica del desarrollo o qué recomendaciones se deben considerar acerca de cómo ir registrando los valores requeridos en las expresiones a lo largo del proceso de desarrollo.

El presente trabajo pues, definió un catastro muy acabado de métricas de software, recogidas de literatura (libros, papers, congresos, casos prácticos, etc.) y, a partir de un esquema propuesto, denominado modelo de definición de actividades de mejoramiento para generar una línea base como fundamento para un plan de métricas¹, en el cual se estableció una relación acabada entre CMM, ISO9000 y propuestas propias, se continúa dicho trabajo en función, ahora, de definir métricas asociadas a cada una de las etapas identificadas del proceso y que deberían ser mejoradas. Del mismo modo, se establecen categorías de métricas tendientes a facilitar su utilización. Por ejemplo: Métricas Directas (medidas directamente) o Indirectas (Requieren varias mediciones no inmediatas). Como producto, se construyó un software que guía la búsqueda de métricas y definiciones entregadas.

El presente trabajo, forma parte de los estudios que se realizan en al Universidad del Bío-Bío de la ciudad de Concepción en Chile, desde el año 1994, sobre temas de mejoramiento de procesos de desarrollo de Software.

Key word : Ingeniería de Software, Planeación Estratégica, Métricas

¹ Trabajo presentado en "I Congreso Latinoamericano de Calidad y Productividad de Software", realizado en la ciudad de Santiago – Chile, en el año 1998.

INTRODUCCIÓN

Es de sumo interés establecer que, dada la gran cantidad de literatura que existe respecto a qué se debe hacer para optimizar el proceso de desarrollo de software y, que a pesar de esta disponibilidad de material, a este tema no se le ha brindado la importancia que merece por parte de los profesionales dedicados al desarrollo de software.

Sabemos por una parte que, para disminuir el riesgo de malas estimaciones, aumentar la productividad del proceso, obtener un alto nivel de calidad, etc., se ha desarrollado una gran cantidad de técnicas y herramientas que proporcionan los conceptos necesarios para alcanzar tales objetivos. Sin embargo, existe también una gran confusión de cual será la apropiada para la realidad a la cual se enfrenta. Las métricas corresponden a una de las herramientas de control que por diversas razones no son muy utilizadas en el área de la Ingeniería del Software. Las causas de tal desinterés definitivamente no corresponde a la carencia o existencia de ellas, por el contrario, sorprende la gran disponibilidad que existe en la literatura de diferentes tipos de métricas y mediciones que son recomendadas para la industria del software.

Debemos dejar claro que mejorar la calidad esta muy asociado a los conceptos de Evaluación y Control. La evaluación brinda el *cómo* estamos realizando alguna actividad, y luego el control ayudará a mantener una línea sobre la cual encaminar los esfuerzos hacia un proceso ideal. La evaluación puede ser cualitativa o bien cuantitativa, entregando ésta

última información más precisión sobre el desempeño. Por su parte, el control es paralelo, pues con estos datos se pueden tomar decisiones ante situaciones críticas que nos alejen de la meta propuesta. Es así, como comenzamos hablar del concepto de métricas, las cuales corresponden a las mediciones obtenidas al aplicar la evaluación, que a pesar de ofrecer numerosos beneficios, simplemente no son ampliamente aplicadas al proceso de desarrollo de software, por la dificultad que presentan al momento de decidir tal o cual utilizar.

Nos encontramos frente a la problemática entonces del *"cómo"*. Sabemos que es necesario medir para evaluar, luego controlar y así mejorar para obtener la calidad deseada, pero *cuándo* medir, *qué* medir y *cómo* medir son algunas de las interrogantes que surgen, al enfrentarnos al concepto de métricas asociadas al proceso de desarrollo de software. Es este el problema al cual esta orientado este trabajo, y se proponen un marco de referencia que ayude a aclarar, en parte, esta nebulosa que existe alrededor.

IDENTIFICACIÓN MARCO DE REFERENCIA

Uno de los objetivos de este proyecto, es realizar una clasificación de las métricas existentes, para lo cual es necesario un marco de trabajo al cual acogerse y el cual, entregue la estructura adecuada para la organización que se busca alcanzar. Como ya hemos señalado, se encuentran a nuestra disposición gran cantidad de métricas y mediciones de software, las cuales entregan muchos beneficios si son utilizadas correctamente, sin embargo existen también

dificultades al momento de decidir cuál métrica utilizar, debido a la falta de claridad que existe con respecto a su organización (clasificación) y a la dificultad de aplicación que presentan. Es común además, encontrar que diversos autores se refieran a la misma métricas con diferente nombre, por lo cual el valor de este trabajo además, es el de homologar definiciones concretas respecto de métricas.

Los modelos, así como estándares, de calidad entregan un marco de referencia para guiar a las organizaciones a incorporar los conceptos de calidad en sus procesos. Las métricas complementan los modelos como herramientas de control que permiten evitar y detectar las desviaciones a los objetivos planteados y además ayudan a realizar estimaciones más precisas que las cuales actualmente se realizan sobre la base de la experiencia del Ingeniero.

Por otra parte, no es posible medir algo que no existe, por lo tanto, es importante que bajo el modelo o norma de calidad que se elija, exista también un esquema de evaluación asociado a él, pues se debe, antes de utilizar la medida, conocer en que áreas puede realmente aplicarlas, lo que implica que primero debe someterse a un proceso de evaluación para luego, así conocer cual es su realidad.

El marco de trabajo seleccionado para la clasificación de métricas del proceso de desarrollo de software fue CMM, por presentar las características que mejor se adaptan a los objetivos y alcances de este proyecto. Las Normas ISO 9001 e ISO 9000-3 constituirán marcos secundarios es decir, a través de la identificación de las interrelaciones que presentan estas normas con el modelo escogido como estructura principal, será posible utilizar las métricas que sean definidas para dichas

áreas, dando la posibilidad de que en el futuro sea expandido hacia la totalidad de estas normas y eventualmente otros modelos de calidad.

ANÁLISIS DE MODELOS Y NORMAS DE CALIDAD SELECCIONADOS

Para los fines de este proyecto, fue necesario realizar un completo estudio de CMM a fin de conocer, en forma acabada, las características de este modelo y establecer con ello, una correcta relación entre sus áreas claves y las métricas que sean recopiladas y eventualmente definidas, ampliando con ello las métricas bases definidas por el modelo.

Del mismo modo, se analizó en forma acabada la documentación de las Normas ISO 9001 e ISO 9000-3, dado que existe una fuerte relación entre CMM y éstas. Por su parte, CMM e ISO 9001 no son excluyentes, por el contrario, poseen relaciones directas entre áreas y cláusulas. Esto permite, que si bien este proyecto se basa en las métricas asociadas al CMM, una empresa que opte por la certificación ISO, podrá utilizar aquellas (métricas) que se encuentren definidas para una determinada área clave que a su vez se relaciona directamente con una determinada cláusula.

Por otro lado, las normas ISO 9001 e ISO 9000-3 están muy relacionadas², excepto por

² Recordar que las ISO 9000-3 son guías para la aplicación de las Normas ISO 9001 en el área del desarrollo de software

(continuación de nota al pie)

ciertas cláusulas de la ISO 9001 que no son cubiertas por las ISO 9000-3, lo cual se debe a que no poseen aplicabilidad en el área de la Ingeniería de Software. A pesar de esto, se pueden relacionar CMM - ISO 9001 - ISO 9000-3, lo que permitirá la reutilización de las métricas encontradas, para el modelo de referencia, por los estándares de calidad, y que en investigaciones posteriores puede ser finalizado, obteniendo así una poderosa herramienta de apoyo a la gestión informática.

CATASTRO DE MÉTRICAS

Para lograr este catastro, fue necesario revisar extensamente la literatura disponible (algunos ejemplos son [2], [3], [4], [5], [6], [10], [11], [12], [13], [14], [24], entre otros), a fin de obtener un número apropiado de métricas de software. En esta búsqueda, se confirma el hecho de que existe gran variedad de métricas mencionadas en la literatura, sin embargo se nota claramente la ausencia de guías para su aplicación, debido a que no siempre, se encuentran junto a una adecuada descripción que señale aspectos tales como: descripción de sus elementos, rangos aceptables para los resultados obtenidos, interpretaciones adecuadas, pautas de aplicación, etc.

Otro elemento importante descubierto en el transcurso de esta búsqueda, fue que muchas mediciones son tratadas con diferentes nombres para su identificación, o bien sus elementos primitivos a considerar, denominados con diferentes siglas o nomenclaturas, siendo que sus resultados son básicamente los mismos, lo

cual sin duda, contribuye a aumentar la confusión de los usuarios. Por lo tanto, es importante mencionar, que las métricas que se presentan, son representativas de las áreas que generalmente son abordadas en la literatura, y para aquellas que no se encuentran consideradas en este estudio, lo más probable es que exista su equivalente en alguna de las que si han sido incluidas.

A pesar de la extensa literatura investigada, quizás el número total de métricas recopilada pareciera bajo, pero una explicación a este hecho, se debe a que distintos autores se refieren a los mismos conceptos, con distintos enfoques y, en ocasiones hasta con diferentes identificadores para una misma métrica, así aunque Fenton - Lawrence, en "Software Metrics"[34], destinan un texto completo a esta área, la orientación de su libro consiste explicar el contexto en el cual se insertan o deberían insertarse las métricas, más que una explicación a fondo de ellas. Por lo demás, explican por qué es importante medir, sus relaciones con modelos de confiabilidad, definiendo el concepto de plan de métricas, etc.

Stephen Kan en "Metrics and Models in Software Quality Engineering"[23], entrega una muy clara descripción del concepto de mediciones, y la importancia que posee, sin querer entregar nuevas métricas que las que generalmente se encuentran, él, entrega un marco de referencia para asociar conceptos y las causas de su relación.

Otro caso fue en "Software Metrics: Establishing a Company - Wide Program", de Robert Grady y Deborah Caswell, [35], en el cual se describe todo el proceso utilizado para implementar un plan de métricas en HP, los pasos que se siguieron para llevar a cabo la implementación, describiendo formularios para

la obtención de información, gráficos para mostrar resultados, etc. Excelente patrón para imitar, puesto que, a ese nivel de detalle, es muy escasa y restringida la información que existe. Por lo anterior, es claro que el objetivo de los autores no está en la descripción de cada una de las métricas, por lo que sólo se incorporan algunas de las utilizadas en el plan de métricas de HP.

La serie de revistas "IT Metrics" [17], en una importante fuente de información de métricas, asociadas al software y al negocio.

Es importante destacar, que las fuentes bibliográficas investigadas que influyeron notablemente tanto para este trabajo, como para otros autores, son los Estándares de la IEEE, tal como "IEEE Standard Dictionary of Measures to Produce Reliable Software" [11], y "IEEE Standard for Software Productivity Metrics" [13], los cuales se dedican exclusivamente a la descripción de algunas métricas.

En el caso de las Métricas para Sistemas Orientados a Objeto, Lorenz & Kidd, "Object - Oriented Software Metrics: A Practical Guide" [24], entregan un completo análisis de las métricas asociadas a este tipo de desarrollo, y el aporte que ellos realizan en esta área, supera cualquier otro, lo que se demuestra en el hecho que de las métricas incorporadas en el catastro, que caen en esta clasificación, ellos aportaron casi la totalidad.

El resto de la literatura analizada, cae dentro de esto mismos casos, es decir, se refieren al concepto de métricas, pero abarcando temas de diferentes ámbitos, identificando algunas, pero no siendo objetivo principal de sus propias investigaciones. Así se confirma la gran variedad de temas relacionados, y la extensa información disponible, pero también se ratifica,

la carencia de abarcar directamente el punto inicial, es decir "Las Métricas".

Algunos ejemplos del catastro de métricas de desarrollo de software y del formato acabado de definición se presentan a continuación

1.- DENSIDAD DE FALLAS

Se puede utilizar para:

- Predecir reaparición de fallas mediante la comparación con las fallas esperadas.
- Determinar si, el testeado ha sido finalizado basado en las metas predeterminadas para clases de severidad.
- Establecer la densidad de fallas estándar para comparar y predecir.

DATOS PRIMITIVOS

F = Número total de faltas únicas encontradas en un intervalo de tiempo determinado, resultantes en fallas de un nivel de severidad.

KSLOC = Número de líneas fuentes de código ejecutable y no ejecutable en miles.

IMPLEMENTACIÓN

1. Los tipos de fallas podrían incluir I/O y usuarios y podrían resultar del diseño, codificación, documentación e inicialización.
2. Observar cada falla.
3. Determinar la falta(s) de programa que causa la falla. Clasificar las faltas por tipo. Podrían ser encontradas faltas adicionales en las faltas globales, siendo mayor el número que las fallas observadas, o una falta podría manifestarse por si misma por diversas fallas. De todas formas, tanto las faltas como las fallas pueden ser medidas.
4. Determinar el total de líneas de código fuente de declaraciones ejecutables y no ejecutables (KSLOC).
5. Calcular la densidad de faltas para un nivel de severidad determinado.

$$F_d = F / KSLOC$$

2.- DENSIDAD DE DEFECTOS

Puede ser utilizada después de las inspecciones del diseño y código de nuevos desarrollos o modificaciones a grandes bloques.

DATOS PRIMITIVOS

- D_i = Número total de objetos únicos detectados durante el i-ésimo proceso de inspección del diseño o del código
- I = Número total de Inspecciones.
- KSLOD = En la fase de diseño, el número de líneas de código fuente de declaraciones de diseño en miles.
- KSLOC = En la fase de implementación, el número de líneas fuente de declaraciones de código ejecutable y no ejecutable en miles

IMPLEMENTACIÓN.

Se debe establecer un esquema de clasificación para la severidad y las clases de defectos. Luego, para cada inspección, registrar el tamaño del producto y el número total de defectos únicos. Por ejemplo, en la fase de diseño, calcular la relación .

$$DD = \frac{\sum_{i=1}^I D_i}{KSLOD}$$

Esta medida asume que se utiliza un lenguaje estructurado.

CLASIFICACIÓN Y ORGANIZACIÓN DE MÉTRICAS VS. MODELOS Y NORMAS DE CALIDAD

Las etapas anteriores, correspondieron a una recopilación de información para definir el marco de trabajo sobre el cual se basa la investigación para luego, realizar una completa revisión de la literatura a objeto de obtener el Catastro de Métricas asociadas al proceso de desarrollo de software y, definir con ello, un formato de definición estándar. Ahora se relacionan estos conceptos, para conformar un esquema conceptual consistente, base para desarrollar posteriormente la Herramientas de Software.

Objetivo central de esta investigación es "Organizar y Clasificar Métricas asociadas al proceso de desarrollo de software, bajo algún esquema que normalice o guíe el desempeño de dicho proceso", para lo cual se ha seleccionado como marco de trabajo principal el Modelo de Capacidad de Madurez CMM. Sin embargo, existen otros marcos que pueden ser considerados como complemento a este, como por ejemplo las Normas ISO 9001 e ISO 9000-3, que incluyen en sus alcances áreas concernientes al proceso de desarrollo SPICE, BOOSTRAP.

Utilizando la relación existente entre CMM y Normas ISO, es posible ampliar las proyecciones de la investigación, pues no sólo se entregan métricas para el modelo CMM, sino que también para las Normas ISO y así, en el futuro, se podrá complementar este trabajo con las cláusulas no cubiertas en ellos. Además, se incluyen otras dos clasificaciones como marcos de referencias, la primera, correspondiente al

ámbito que poseen las actividades desarrolladas en el proceso de desarrollo, dentro de las cuales se han identificado tres grandes grupos:

- Actividades de Proyecto
- Actividades del Ciclo de Desarrollo
- Actividades asociadas al Producto de Software

La segunda clasificación se refiere específicamente a las Fases que incluye un ciclo estándar de desarrollo, para lo que se han identificado las siguientes etapas características:

- Requerimientos
- Diseño
- Codificación
- Documentación
- Prueba
- Puesta en Marcha

Para demostrar como se relacionan las clasificaciones mencionadas, se han construido diferentes matrices de referencias cruzadas, así también se demuestra que el modelo seleccionado como principal, no es una entidad aislada, y que, por el contrario, está estrechamente ligado a conceptos de uso habitual para los profesionales del área.

Finalmente, y el punto de mayor relevancia, corresponde a la clasificación de las métricas de acuerdo a los esquemas previamente explicados, lo cual significó un análisis detallado del Modelo CMM, y marcos secundarios, a fin de entender claramente sus objetivos, y poder de esta forma asignarle la(s) métrica(s) que fuesen apropiadas para cumplir con esos objetivos.

Las interconexiones realizadas se pueden apreciar gráficamente en la *Figura 1*, en donde las líneas se traducen en matrices de referencia cruzadas construidas para demostrar dichas relaciones, las cuales corresponden a:

- 1) Áreas Claves CMM vs. Cláusulas Normas ISO 9001
- 2) Áreas Claves CMM vs. Cláusulas Normas ISO 9000-3
- 3) Áreas Claves CMM vs. Ámbito (Proyecto - Ciclo - Producto)
- 4) Áreas Claves CMM vs. Etapas Ciclo Desarrollo
- 5) Métricas Orientadas a Objeto vs. Etapas Ciclo Desarrollo
- 6) Métricas Orientadas a Objeto vs. Ámbito (Proyecto - Ciclo - Producto)
- 7) Métricas de Aplicación General vs. Ámbito (Proyecto - Ciclo - Producto)
- 8) Métricas de Aplicación General vs. Etapas Ciclo Desarrollo
- 9) Métricas de Aplicación General vs. Áreas Claves CMM
- 10) Métricas de Aplicación General vs. Forma de Implementación (Directa - Indirecta)
- 11) Métricas de Aplicación Indirecta vs. Mediciones a Implementar.

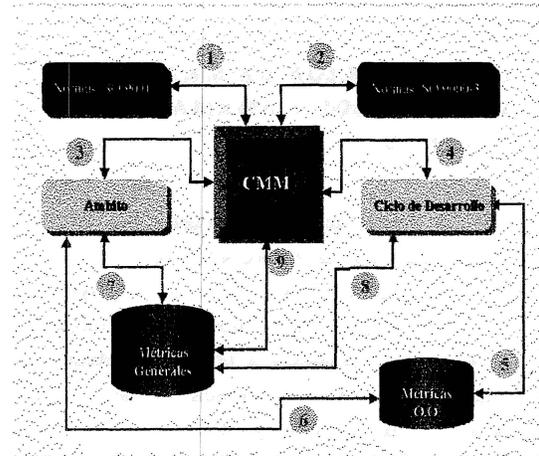


Figura 1.- Interconexiones Modelos y Métricas

Otra clasificación importante, dice relación con la forma de implementación, es decir, si es necesario medir uno o más atributos para obtener el resultado de ella. Según esto, se han identificados en dos grupos:

- Métricas de Aplicación Directa
- Métricas de Aplicación Indirecta

Una métrica se considerará directa cuando sólo es evaluado un atributo, y será indirecta en caso de necesitar combinar diferentes evaluaciones. Luego, se identificaron los parámetros (o mediciones) necesarios para cada métrica de categoría indirecta, de manera de identificar los puntos de interacción entre ellas.

Estas relaciones se presentan en este capítulo en formato de matrices, que gráficamente se pueden observar en la Figura 2, donde las líneas corresponden a:

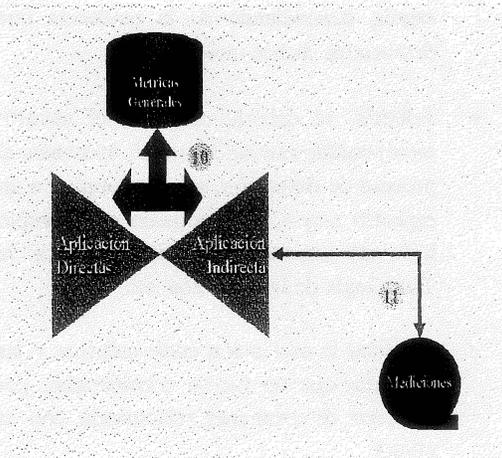


Figura 4.2.- Organización Interna Métricas

CONCLUSIONES

Uno de los principales aportes al área informática, que deja esta investigación, corresponde al "Catastro de Métricas asociados al Proceso de Desarrollo de Software", el cual toma característica de inédito según lo indica la literatura estudiada, ya que no existe una recopilación acabada de métricas - que además incluya las específicas para sistemas orientados a objetos - ya que por el contrario, éstas se encuentran dispersas en estudios con diferentes orientaciones, que no siempre se dedican netamente a este concepto.

El catastro, como se indicó en el párrafo anterior, incorpora métricas que son de aplicación genérica, esto quiere decir, que pueden ser aplicadas indistintamente del modelo de ciclo de vida utilizado, o forma particular de desarrollo; además son descritas aquellas métricas asociadas específicamente con desarrollo de sistemas orientados a objeto, las

que no constituyen un gran porcentaje dentro del total, lo que no significa que carezcan de importancia, por el contrario, información asociada a este tipo en particular es muy escasa, contradictoriamente con el gran auge que tienen hoy los ambientes de desarrollos orientados a objeto.

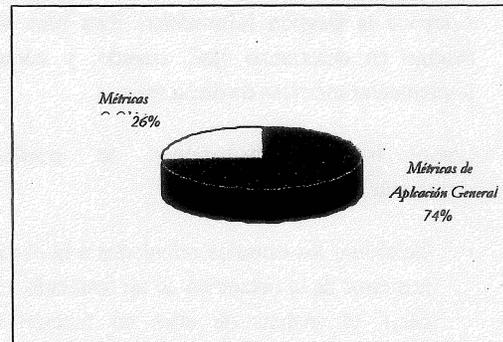


Figura C1.- Distribución de Tipo Métricas en el Catastro

De igual importancia y relevancia al conocimiento son también las clasificaciones realizadas para cada una de las métricas incorporadas en el catastro. Es decir, la asociación a un esquema de trabajo, que oriente a los administradores de proyecto en cual es el conjunto de métricas apropiado según las diferentes formas de desarrollo que desee adoptar, por lo que se pueden obtener métricas de software para cada área del Modelo de Capacidad de Madurez CMM, que ha sido el marco principal para la clasificación, como también Métricas para las Normas ISO 9001 e ISO 9000-3, y conocer, a un nivel de concepto más habitual como lo son las etapas típicas de un ciclo de desarrollo de proyecto y las métricas posibles de implementar en cada una de ellas. Por último, una clasificación algo más extensa, es el nivel del ámbito de las actividades que se

encuentran en un proceso de desarrollo de software, esto es, actividades del Proyecto de Software, del Ciclo de Desarrollo de Software y, actividades del Producto de Software.

Con lo anterior, se deja una base sólida, que ayuda a aclarar la nebulosa que siempre surge, al querer utilizar Métricas como herramientas de Apoyo a la Gestión Informática. Esta base se resume en determinar qué, cuando, y cómo implementar métricas de dicha área.

Como futuras proyecciones se pueden mencionar algunas como:

- Continuar los estudios orientados a la etapa posterior de la obtención de un resultado, es decir, el análisis de ellos en función a estándares, que permitan definir acciones a

seguir, dependiendo de la tendencia que demuestren dichos indicadores.

- Además, se debería mantener el catastro incorporando nuevas Métricas, asociadas al proceso de desarrollo, y luego ampliar a un espectro mayor, incorporando por ejemplo Métricas asociadas al Negocio, a la Tecnología de Información, etc.
- Finalizar la asociación entre métricas y las cláusulas que no fueron consideradas, por no estar directamente relacionado con el CMM.
- Incorporar Nuevos Modelos de Calidad, como otra forma de clasificar y ordenar los conceptos de métricas analizados.

BIBLIOGRAFÍA

- [1] *¿Puede Iso 9000 & Iso 9000-3 Evaluar La Capacidad De Madurez Del Software?*
AMERICAN PROGRAMER, FEBRERO 1994
- [2] *A Manager's Guide to Software Engineering*
Roger S. Pressman, Ph. D.
McGRAW – HILL, INC, 1993
- [3] *A Practical View Of Software Measurement and Implementation Experiences Within Motorola.*
Michael K. Daskalantonakis, Member, IEEE
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING
Vol. 18, No 11, November 1992
- [4] *An Evaluation of code metrics for object-oriented programs.*
R. Harrison, L.G. Samaraweera, M.R.
- [5] *Applied Software Measurement*
Carpes Jones
McGraw – Hill, INC, 1991
- [6] *Apunte "Análisis por Puntos de Función"*
Preparado por Tatiana Aldea
DIVISIÓN INFORMÁTICA –
UNIVERSIDAD CONCEPCION, 1995
- [7] *Cyclomatic Complexity and the year 2000.*
Thomas McCabe, McCabe & Associates
IEEE SOFTWARE, MAY 1996
- [8] *Function Point Analysis: Difficulties and Improvements*
Charles R. Symons
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING
VOL 14, N° 1, January 1988

Dobier, P.H. Lewies
INFORMATION AND SOFTWARE
TECHNOLOGY, N° 38, 1996

- [9] *How ISO 9001 Compares With The CMM*
IEEE Software, January 1995
- [10] *IEEE Guide for The Use of IEEE Standard Dictionary of Measures to Produce Software.*
SOFTWARE ENGINEERING STANDARDS SUBCOMMITTEE OF THE IEEE COMPUTER SOCIETY.
- [11] *IEEE Standard Dictionary of Measures to Produce Reliable Software.*
SOFTWARE ENGINEERING STANDARDS SUBCOMMITTEE OF THE IEEE COMPUTER SOCIETY.
IEEE Std. 982.1-1988
- [12] *IEEE Standard for a Software Quality Metrics Methodology.*
SOFTWARE ENGINEERING STANDARDS SUBCOMMITTEE OF THE IEEE COMPUTER SOCIETY.
- [13] *IEEE Standard for Software Productivity Metrics.*
SOFTWARE ENGINEERING STANDARDS SUBCOMMITTEE OF THE IEEE COMPUTER SOCIETY.
- [14] *IEEE Standard Glossary of Software Engineering Terminology*
SOFTWARE ENGINEERING STANDARDS SUBCOMMITTEE OF THE IEEE COMPUTER SOCIETY.
- [15] *Ingeniería de Software: Métricas de Software*
Ricardo A. Gacitúa B.
VERSIÓN PRELIMINAR, 1995
UNIVERSIDAD DEL BIO – BIO, CONCEPCION
- [16] *INTERNATIONAL STANDAR ISO 9000-3*
First Edition 1991 – 06 - 01
AMERICA XXI LTDA.
Reference number ISO 900-3:1991(e)
- [17] *ISO 9000 for Software Developers.*
Revised Edition
Charles H. Schmauch
1995 by ASQC
- [18] *IT Metrics*
Helping Management Measure Software & Processes and their business value.
EDITOR: HOWARD RUBIN
- Ediciones Revisadas entre Marzo 1996, hasta Abril 1998.
- [19] *Maturity, Models, and Goals: How to Build Metrics Plan.*
Shari Laurence Pfleeger.
CENTRE FOR SOFTWARE REABILITY CITY UNIVERSITY, LONDON ENGLAND – System Software, 1995
- [20] *Metodología que Genere una Línea Base como Fundamento para un Plan de Métricas*
Marcelo Meneses C. – Ricardo Gacitúa B.
Trabajo de Título Carrera IGENIERIA CIVIL EN INFORMÁTICA
UNIVERSIDAD DEL BIO – BIO, CONCEPCION, 1997
- [21] *Métricas Básicas*
INFORMATICA, JUL1990
- [22] *Métricas de Calidad para el Sistema de Información de la Nación S.A. (SILAN)*
Mauricio Lenadro, Marcelo Jenkins.
Escuela de Ciencias de la Computación e Informática.
UNIVERSIDAD COSTA RICA, SAN PEDRO COSTA RICA.
- [23] *Metrics – A tool for Defining and Measuring Quality*
QUALITY ASSURANCE FOR INFORMATION SYSTEMS
- [24] *Metrics and Models in Software Quality Engineering.*
Stephen H. Kan
ADDISON WESLEY, 1995
- [25] *Object - Oriented Software Metrics: A Practical Guide*
Mark Lorenz - Jeff Kidd
PTR PRENTICE HALL, 1994
- [26] *Quality Measures in Design. Finding Problems Before Coding*
Dennis L. Brandi
SOFTWARE ENGINEERING NOTES
VOL 15, N° 1, January 1990
- [27] *Quantitative Analytic approach in Software Engineering*
Koji Torii, Ken – ichi Matsumoto
INFORMATION AND SOFTWARE TECHNOLOGY, N° 38, 1996

- [28] *Software Development Cost Estimation Using Function Points.*
Jack E. Matson, Bruce E. Barrett, and Joseph M. Mellichamp
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING
VOL 20, N° 4, APRIL 1994
- [29] *Software Faults, Software Failures and Software reliability Modeling*
John C. Munson
INFORMATION AND SOFTWARE TECHNOLOGY, N° 38, 1996
- [30] *Software Metrics*
SEI Curriculum Module SEI – CM – 12 – 1.1
Carnegie Mellon University
SOFTWARE ENGINEERING INSTITUTE
December, 1998
- [31] *Software Metrics and Measurement Principles*
John M. Roche BSc, Msc
SOFTWARE ENGINEERING NOTES
VOL 19, N° 1, January 1994
- [32] *Software Metrics Best Practices*
Peter J. Kulik
[Http://www.klci.com/accelerate](http://www.klci.com/accelerate)
- [33] *Software Metrics in the Process Maturity Framework.*
Shari Lawrence Pfleeger, Clement McGowan.,
JORNAL OF SYSTEMS AND SOFTWARE, VOL 12, N°3, JUL 1990
- [34] *Software Metrics: A Practitioner's guide to improved product Development.*
K. – H. Möller, D. J. Paulish
CHAPMAN & HALL COMPUTING, 1993
- [35] *Software Metrics: A Rigorous & Practical Approach*
Norman E. Fenton - Shari Lawrence Pfleeger
SECOND EDITION 1997-
INTERNATIONAL THOMSON COMPUTER PRESS
- [36] *Software Metrics: Establishing a Company – Wide Program*
Robert B. Grady - Deborah L. Caswell
PTR Prentice Hall, 1987
- [37] *Software Metrics: Guest Editor's Introduction*
Bill Curtis
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING
VOL SE – 93, N°6, NOV 1983
- [38] *Software Products Evaluation System: quality models, metrics and processes – International Standards and Japanese Practice*
Motoei Azuma
INFORMATION AND SOFTWARE TECHNOLOGY, N° 38, 1996
- [40] *Software Size Estimation of Object Oriented Systems.*
Luis A. Larenjeira
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING
VOL 16, N°5, MAY 1990
- [41] *Software Sizing and Estimating*
Charles R. Symons
JOHN WILE & SONS, 1991
- [42] *Technical Reports*
CMU/SEI- 93 – TR – 25
ADA 263403
SOFTWARE ENGINEERING INSTITUTE
CARNEGIE MELLON UNIVERSITY
- [43] *Technical Reports*
CMU/SEI- 93 – TR – 24
ADA 263403
SOFTWARE ENGINEERING INSTITUTE
CARNEGIE MELLON UNIVERSITY
- [44] *The Evolution of the SEI's Capability Maturity Model for Software*
Mark C. Paulk,
SOFTWARE PROCESS – IMPROVEMENT AND PRACTICE, 1995
- [45] *Tutorial Calidad del Proceso de Desarrollo de Software*
XVII TALLER DE INGENIERIA DE SISTEMAS
UNIVERSIDAD DE CHILE, 1994