

INDICADORES EMPÍRICOS FORMALES Y MUY TEMPRANOS DE COMPLEJIDAD ESENCIAL DE SISTEMAS DE GESTIÓN INTENSIVA DE DATOS: UN MODELO CONCEPTUAL

Pedro Salvetto¹, José Carrillo², Oscar Marbán², Julio Fernández³, Juan Carlos Nogueira¹,
Javier Segovia²

¹Laboratorio de Investigación en Sistemas de Información,
Facultad de Ingeniería Universidad ORT Uruguay Cuareim 1451 CP 11100, Montevideo, Uruguay
E-Mail : psalvetto,nogueira@ort.edu.uy

²Departamento de Lenguajes y Sistemas e Ingeniería de Software,
Facultad de Informática, Universidad Politécnica de Madrid Campus de Montegancedo 28660 – Boadilla del Monte Madrid

E-Mail : jcarrillo,omarban,fsegovia@fi.upm.es

³ Universidad ORT Uruguay Cuareim 1451 CP 11100, Montevideo, Uruguay

E-Mail : jfernandez@ort.edu.uy

Abstract Despite many years of research, the problem of formal and structured estimation (that is independent of human expertise) of time and effort required for intensive data management systems remained as an open issue. This paper introduces:

(a) A set of early and formal indicators for essential complexity of IS, which can be calculated from the set of final user visions. Those metrics are independent from: (1) the human expertise, (2) the technology used in the development of the IS, and (3) the set of final user visions from which they have been obtained.

(b) A set of empirical formal models independent from human expertise, which deliver very early estimates for IS development time and effort. These models can be applied to intensive data management IS, based on relational DBMS, developed with agile evolution processes and automatic code generation from formal specifications¹.

Resumen A pesar de largos años de investigación y desarrollo el problema de la estimación formal y estructurada (independiente del juicio experto) del tiempo y esfuerzo requeridos para desarrollar un sistema de información intensivo en gestión de datos, permaneció abierto.

En este trabajo se presentan

(a) Indicadores formales, y muy tempranos de complejidad esencial de SI, calculables a partir del conjunto de las visiones de datos de sus usuarios finales. Estos indicadores son independientes (1) del juicio experto, (2) de la tecnología usada para desarrollar el SI y (3) del conjunto de visiones de datos de los usuarios a partir del que se obtengan (la forma en que los usuarios visualicen los datos que el sistema debe procesar del SI).

(b) Modelos empíricos, formales, independientes del juicio experto de estimación muy temprana del tiempo y esfuerzo de desarrollo de SI.

Los modelos de estimación de tiempo y esfuerzo son aplicables a Sistemas de Información de Gestión Intensiva de Datos, basados en bases de datos relacionales, desarrollados con procesos evolutivos y ágiles, metodologías de desarrollo orientadas a los datos y generación automática de código a partir de especificaciones formales².

Palabras Clave: Métricas e Indicadores Tempranos de Complejidad, Metodologías Ágiles, Modelos de Estimación de de Tiempo y Esfuerzo de Proyectos Software

¹ Our definition for formality in the specification is not restricted to rigorous mathematical formal languages. We assume that a specification is formal if it contains enough information to generate automatically the code in any particular language and platform, using an existing code generator.

² Nuestro criterio respecto de la formalidad de la especificación no está restringido al rigor matemático de los lenguajes de especificación formal. Consideramos que una especificación es formal si contiene información suficiente para generar automáticamente la aplicación en cualquier lenguaje y plataforma (siempre que esté disponible el generador correspondiente).

1. INTRODUCCIÓN Y MOTIVACIONES

La ingeniería de software tiene numerosos puntos de contacto con disciplinas más antiguas como la arquitectura y la ingeniería civil. Sin embargo, debido a su juventud y la intangibilidad del producto que genera, no ha alcanzado, todavía, el mismo grado de formalización [1,2,3,4,5]. En particular, carece de patrones de medida de tamaño del software independientes del juicio experto y que puedan ser útiles para seguir el avance de los proyectos de desarrollo, estimar tempranamente sus costos, gestionar cambios, apoyar relaciones contractuales más laxas y naturales sin riesgo para las partes, y aumentar la transparencia e integración de la relación entre clientes y desarrolladores [1,6,7,8].

La industria, la academia y la sociedad en general, reclaman que el desarrollo de software se aleje más de ser una actividad artesanal, se profesionalice y comience a ser más predecible. Una de las principales dificultades para ello es nuestra pobre comprensión de los procesos internos que se producen en los equipos de trabajo. Sin embargo, hasta tanto alcancemos ese conocimiento, podemos construir modelos empíricos que nos permitan estimar [9].

Un problema de los modelos de estimación actuales es que son dependientes del juicio experto y requieren gerentes con experiencia para aplicarlos [10].

Los gerentes con experiencia, son naturalmente escasos [10] y la tendencia actual hacia metodologías ágiles con equipos reducidos, altamente motivados y comprometidos con el proyecto profundiza el problema.

Para reducir la incidencia de estos problemas es importante disminuir las fuentes de variación tanto en el proceso de desarrollo como en el de estimación. Entre las principales se encuentran la tecnología, el juicio experto y las métricas de tamaño no

tomadas automática y tempranamente como LOC y FP [1,11,12].

Independizar la estimación (no la interpretación de sus resultados) del juicio experto, (1) aumenta la credibilidad de la industria, (2) facilita la relación entre clientes y desarrolladores y aumenta su transparencia, apoya (3) la gestión conjunta de los cambios, (4) la discusión sobre bases objetivas y claras, para todos los actores, de plazos, alcance de los proyectos y los riesgos derivados de ellos [1], (5) contribuye a poner en evidencia, sobre bases objetivas, los riesgos que introducen las fechas fijadas con criterios no técnicos (por ejemplo políticos, o no objetivos), (6) puede ser el comienzo de metodologías de gestión de contratos que no se basen en un precio cerrado en el momento en que menos se sabe del proyecto sino en una metodología de cálculo de acuerdo a las condiciones de ejecución del proyecto (volatilidad de requisitos, eficiencia de la organización, esfuerzo medio o velocidad de desarrollo, etc.) y la complejidad del sistema a construir y (7) apoya equipos pequeños que no dispongan de gerentes con experiencia [11,12,13].

Disciplinas como la arquitectura e ingeniería civil disponen de elementos de apoyo como planos, maquetas etc. a partir de las cuales se puede extraer métricas en escala de proporción. Esto permite ajustar el diseño del producto a desarrollar y, al mismo tiempo, evaluar, sobre bases concretas y objetivas, su impacto sobre los costos, plazos e incluso las tecnologías adecuadas para su construcción.

Indicadores calculables tempranamente sobre la base de las visiones de datos de los usuarios y modelos que, a partir de ellos y las condiciones de ejecución del proyecto, permitan estimar plazos y costos, seguramente no elevarán la formalización de la ingeniería de software a niveles tan altos como los de las disciplinas antes mencionadas pero impactarán el estado del arte y serán útiles a la hora de definir el alcance, estimar plazos y costos de un

proyecto y negociar las condiciones de su ejecución [1].

2. ¿CRISIS DEL SOFTWARE O VARIABILIDAD INTRÍNSECA DE LOS PROCESOS DE DESARROLLO Y ESTIMACIÓN?

A diferencia de los procesos de producción industrial, los procesos de producción de software generan productos intangibles y requieren comunicación y coordinación intensivas lo que contribuye a aumentar los riesgos y dificultar la estimación.

La industria del software es bien conocida por la frecuencia con que los proyectos fracasan al tratar de alcanzar sus objetivos de planificación, esfuerzo y calidad [14,15,16,17,18,19,20,21,22,23,24,25,26].

Según Kitchenham (1996)[27] una de las razones por las que esto ocurre es que los desarrolladores no están en condiciones de estimar con suficiente antelación y precisión el tiempo y esfuerzo requeridos para el desarrollo de software.

La estimación es importante, no sólo para mantener los proyectos bajo control sino incluso para determinar su viabilidad; ya que no tendría sentido iniciar un proyecto sin disponer de tiempo y/o presupuesto para completarlo o si este no alcanza la funcionalidad y calidad requeridas [1]. Para que la estimación sirva a estos fines debe ser suficientemente temprana y precisa [28]. El software representa, hoy, la mayor parte del costo de los sistemas, a diferencia de lo que ocurría en los años 50 y una estimación acertada y temprana de su costo es esencial [29, 30,31,32,33].

2.1 La Estimación en Etapas muy Tempranas del Ciclo de Vida y la Crisis del Software.-

La estimación juega un rol central a la hora de evaluar el éxito o fracaso de un proyecto.

Esta valoración surge de la comparación entre el tiempo, esfuerzo y calidad planificados y los resultantes de la ejecución del proyecto [10,11]. Así, nuestra dificultad para estimar tempranamente puede ser una de las causas contribuyentes a la crisis del software.

En el momento en que normalmente se solicita a los desarrolladores que estimen tiempo y esfuerzo, estos conocen tan poco sobre el proyecto que puede suceder que las estimaciones pesimistas sean 16 veces mayores que las optimistas [2,29,34,35]. Esta situación es mencionada por algunos autores como el cono de incertidumbre[4,29,34].

Posiblemente esta incertidumbre, contribuya a explicar lo que ha dado en llamarse la crisis del software [27], caracterizada por sobrepasar sistemáticamente los presupuestos y tiempos planificados [9,15,16,18,19,20,22,23,24,25,26,36,37,38,39,40]. Así muchos proyectos podrían estar irremisiblemente condenados al fracaso desde su nacimiento debido a estimaciones inadecuadas, aunado al hecho de que, durante el lanzamiento, generalmente se realizan trabajos apresurados y no hay tiempo para detenerse a pensar [8,13].

La mayoría de los modelos de proceso de desarrollo en Ingeniería de Software parten de la suposición poco realista de que existe una especificación de requisitos y que esta no va a cambiar [41,42,43,44].

Esta es una debilidad ya que el cambio en los requisitos es inevitable [10,29,30,41,42,45,46,47,48,49,50,51,52,53,54].

Incluso el rol de los requisitos en la validación y verificación está siendo crecientemente cuestionado [42] y cada vez más se considera que no son una base adecuada para estimar tiempo y esfuerzo aún cuando estén completos [1].

Para poder superar el cono de incertidumbre que existe cuando solamente se dispone de una especificación de requisitos de usuario de alto nivel debe recurrirse a alguna medida de

complejidad esencial del sistema a construir [11,12,13,55,56,57].

2.2 Estimación Temprana y Variabilidad

Algunas de las características del desarrollo de software que dificultan la construcción de modelos de estimación son: (1) requiere trabajo intelectual e interacción humana intensivos; (2) a pesar del nivel repetible del CMM propuesto por el SEI, los procesos de desarrollo de software repetibles son raros; (3) los proyectos de desarrollo de software pueden ser replicados, pero no repetidos; (4) se trata de un proceso social cuyos detalles no conocemos en profundidad; (5) la tecnología e incluso los paradigmas cambian con tanta rapidez que es difícil, e incluso propenso a errores, incorporar la experiencia pasada; (6) existen numerosos factores que influyen al proceso e introducen una gran variabilidad.

Estas características contribuyen a explicar lo que tradicionalmente llamamos crisis del software y que se expresa en cifras como las de la Tabla 1 con las que nos hemos acostumbrado a convivir. Estas muestran, más que una crisis (que duraría 4 décadas), la inmadurez de la ingeniería del software que se encuentra en etapas iniciales de su desarrollo.

	1994	1996	1998	2000	2002	2004
Exitosos	16%	27%	26%	28%	15%	29%
Comprometidos	53%	33%	46%	49%	51%	53%
Fallaron	31%	40%	28%	23%	34%	18%
No Exitosos	84%	73%	74%	72%	84%	91%

Tabla 1 Fuente Standish Group Inc.

El proceso de construcción presenta aspectos sociales y grupales que hacen a la relación entre usuarios, desarrolladores, gestores y estimadores y pueden afectar de manera muy importante la productividad tanto personal como grupal.

Los cambios en las tecnologías y paradigmas de desarrollo son cada vez más frecuentes y sus ciclos de vida más cortos. Los desarrolladores no han terminado de escalar una curva de aprendizaje y se encuentran

nuevamente en el llano perdiendo, además, significación y relevancia la experiencia previa tanto personal como grupal y organizacional.

También operan cambios en el entorno, que traen como consecuencia que pierda valor, aunque en menor medida, la experiencia en el dominio del negocio.

Se han reportado importantes variaciones de productividad tanto personales como grupales. Boehm [28,38,58,61] reconoce que la gerencia puede incidir multiplicando hasta por dos tiempo y esfuerzo, pero, no encontrando la forma de evaluarla, asume que los proyectos están correctamente gestionados.

En los procesos de desarrollo y estimación existen numerosas fuentes de variación. A modo de ejemplo se presentan algunas de ellas en la Figura 1.

A pesar de largos años de investigación el problema de la estimación formal y estructurada (independiente del juicio experto, repetible y apoyada por métricas, indicadores y modelos que pueden ser automatizados) del tiempo y esfuerzo requeridos para desarrollar un sistema de información intensivo en gestión de datos, en adelante SIGID, permanece abierto. Las técnicas de estimación más extendidas actualmente se apoyan en la premisa - poco realista - de estabilidad de requisitos y datos, requieren expertos humanos y se basan en métricas disponibles recién en la fase de diseño temprano del sistema.

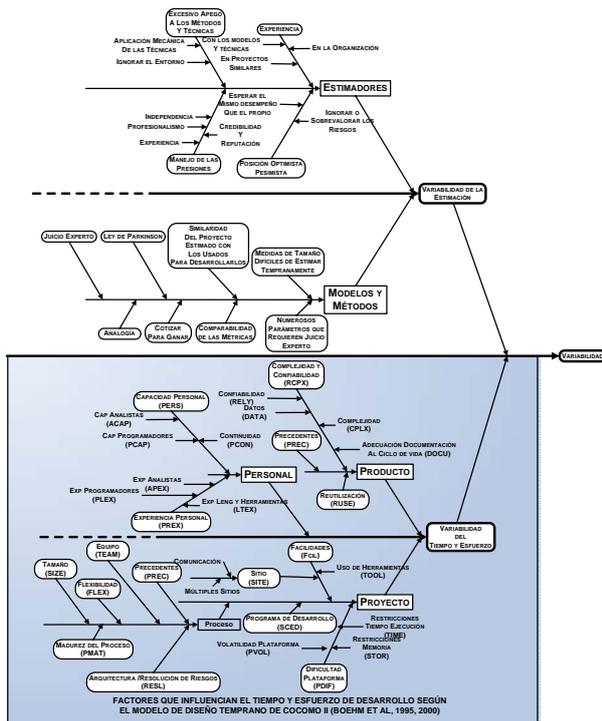


Figura 1 Variabilidad de los Procesos de Estimación y Desarrollo

La falta de capacidad que la industria del software ha evidenciado para producir estimaciones de tiempo y esfuerzo con un nivel de precisión y consistencia adecuado es bien conocida [26,27,28,30,32,59,60,62].

Del 60% al 80% de los proyectos exceden sus presupuestos y/o cronogramas en alrededor del 40%. Sin embargo, los sobrecostos, siendo preocupantes, no parecen ser tan altos como los reportados por el Standish Group [37,38].

El desarrollo de software es hoy el componente más costoso de los sistemas informáticos [28,30,32,63]. El mayor contribuyente de este costo es el esfuerzo humano. En consecuencia, la mayoría de los modelos de estimación de costos arrojan meses-hombre [63].

Al mismo tiempo que el software aumentó su tamaño, aumentó su complejidad y la dificultad de la estimación [46].

Esta situación fue vivenciada como una amenaza por la industria y un desafío por la academia, desencadenando la primera ola de

reacciones que se manifestaron de variadas formas:

1. modelos empíricos de estimación que intentan capturar la influencia de fuentes de variación incorporando numerosos parámetros que requieren juicio experto para su estimación (al punto que Boehm (1989)[34], sugiere que los parámetros de estos modelos son buenas listas de verificación al estimar el riesgo de un proyecto) y toman como entrada medidas del tamaño que, en etapas tempranas del ciclo de vida, sólo pueden ser estimadas groseramente como LOC o FP
2. metodologías de identificación, control y seguimiento de riesgos basadas en heurísticas, listas de verificación y juicio experto
3. no reconocer que los requerimientos y, en general, las condiciones de ejecución del proyecto, más allá de nuestros deseos o intenciones, van a cambiar, refugiándose en el mito de los requerimientos y datos estables y conocidos
4. establecer las condiciones de ejecución del proyecto desde el inicio, el momento en que menos conocen ambas partes del sistema a construir
5. propuestas de formalización excesiva en áreas cuya madurez actual las hace contraproducentes o, en las antípodas, basar la estimación exclusivamente en el juicio experto.

Estas aproximaciones dominaron hasta mediados de los 90 y no hicieron otra cosa que profundizar el problema ya que presentan debilidades comunes:

1. requieren juicio experto, por lo que distintos evaluadores, observando el mismo escenario, pueden arribar a conclusiones significativamente diferentes, lo que nos confunde a nosotros mismos y aún más a nuestros clientes
2. la intangibilidad del producto genera un espacio semántico entre usuarios y desarrolladores ya que no disponemos de elementos comprensibles y significativos

para los usuarios a partir de los cuales discutir, negociar y analizar el impacto sobre el negocio y los costos y riesgos del proyecto de fechas, requerimientos y planes de versiones.

3. los gerentes expertos en que se basan son un recurso naturalmente escaso
4. este estado de cosas condujo a las tendencias actuales al trabajo con equipos reducidos y altamente motivados dando lugar al movimiento agilista, lo que aumenta la cantidad de expertos que se requieren y profundiza el problema.

A pesar de que esfuerzo y costo se encuentran íntimamente relacionados, esta relación no es simple. Basar las estimaciones en el juicio experto presenta dificultades ya que (1) es un proceso no repetible, (2) no es fácil disponer de expertos para ello [64], (3) la relación entre tamaño y costo no es lineal [28] y (4) no necesariamente la experiencia pasada es aplicable, peor aún, si las circunstancias son diferentes sería contraproducente su aplicación [63,65,66,67] y el buen desempeño anterior de un experto no es garantía del presente [10].

En suma, se requieren modelos de estimación e indicadores de complejidad independientes del juicio experto aplicables en etapas tempranas del ciclo de vida.

3. MODELO CONCEPTUAL

Deseamos desarrollar modelos de estimación independientes del juicio experto y la tecnología usada para desarrollar los Sistemas de información estudiados e inferir indicadores de complejidad esencial, que no contengan información ajena a los sistemas y el dominio del problema que estos intentan resolver.

Con la finalidad de disminuir la variabilidad inherente a los procesos de estimación y desarrollo, se decidió prescindir del juicio experto y observar sistemas desarrollados con herramientas de generación de aplicaciones a partir de especificaciones formales siguiendo

procesos orientados a los datos basados en la visión de los datos de los usuarios y sus solicitudes no redundantes.

3.1 Definiciones, Axiomas y Supuestos en que se Basa la Investigación.-

Definición 1: Sistema de Información de Gestión Intensiva de Datos (SIGID) Es un sistema de información desarrollado en torno a bases de datos relacionales utilizadas para almacenar información y no para soportar bases de datos orientadas a objetos, geográficas, almacenar objetos complejos, u otros usos no afines a los sistemas de gestión en el que

1. los algoritmos de alta complejidad son raros o no existen
2. sobre cuyos tiempos de respuesta no existen cotas máximas consideradas críticas, aunque pueden existir restricciones sobre los tiempos de respuesta promedio

Definición 2: Atributo Dato atómico que tiene una semántica definida para algún usuario del SIGID y al que se asocia un tipo y nombre.

Definición 3: Vista de datos de usuario (VDU) Conjunto de atributos que representan la visión que un usuario tiene de alguna entidad que debe ser representada en el SIGID, tal como este la percibe sin aplicar ningún criterio de diseño, normalización o de cualquier otro tipo por parte de los desarrolladores del sistema.

Definición 4: Visión de datos de los usuarios de un sistema (VDUS) Conjunto de vistas de datos de usuario (VDU) de un SIGID, no necesariamente disjuntas, en las que, atributos de igual semántica tienen el mismo nombre.

Axioma 1: Necesidad de Integración de las Vistas de Datos de Usuario (VDUs) En una organización mediana o grande, nadie tiene una visión global de los datos y los procesos. Es necesario, entonces, integrar las visiones

de datos de los usuarios de un sistema para obtener una visión integral de sus datos.

Definición 5: SIGID esencial (SIGIDES) SIGID en el que sólo se desarrolló funcionalidad no redundante a partir de la información contenida en la VDUS proporcionada por sus usuarios y las solicitudes de estos (sin considerar datos o funcionalidades que al desarrollador le parecieron interesantes o introducidos como artificio de diseño).

Definición 6: Modelo Empírico Global de Estimación Formal (MEGEF) Modelo empírico de estimación global que permite estimar tiempo o esfuerzo totales de desarrollo de un SIGIDES, no requiere juicio experto y puede ser automatizado.

Definición 7: Indicador de Complejidad esencial de un SIGIDES (ICES) Número real que puede ser obtenido, antes de que tenga lugar ninguna actividad de diseño, sin recurrir a juicio experto, a partir de una VDUS (cualquiera) del SIGIDES y es una variable independiente de algún MEGEF para el SIGIDES a partir del cual se obtuvo.

Axioma 2: El ICES es independiente de la VDUS a partir de la que se obtenga. Intuitivamente esto se explica porque los sistemas objeto de estudio son SIGIDES con una proporción mínima de algoritmos de alta complejidad, las visiones de usuario no están influenciadas por la tecnología, conceptos de diseño o cualquier otro criterio ajeno al sistema a construir introducido por los diseñadores y el mismo sistema puede ser visto por los usuarios de múltiples maneras, pero la VDUS resultante deberá contener información para desarrollarlo, entonces, desde este punto de vista, todas son equivalentes

Axioma 3: Cambio Continuo inherente a los Sistemas de Información Los requisitos, el entorno y las visiones de datos de los usuarios sufren cambios continuos e inevitables a lo largo del tiempo. Así, la integración de las VDUS debe ser automática y continua.

Definición 8: Información Esencial de una VDUS (IEV) La máxima cantidad de información no redundante sobre los atributos de alguna entidad o las relaciones entre ellas que puede inferirse a partir de las VDUs que contiene.

Definición 9: Funcionalidad Esencial de una VDUS (FEV) La máxima funcionalidad, no redundante, que es posible obtener a partir de su IEV.

Axioma 4: La Información Esencial de todas las VDUS de un SIGIDES es la misma.

Axioma 5: Una VDUS de un SIGIDES determina los ICES del SIGIDES.

Definición 10: Funcionalidad Potencial de un SIGIDES (FPS) Porción de la FEV aún no desarrollada.

Axioma 6: El ICES es un indicador de la FEV y su semántica es la “energía potencial de la estructura de los datos del Sistema de Información”. El ICES podría describirse como la energía potencial de los datos del sistema. Es un indicador de lo máximo que se puede hacer con la estructura de los datos disponibles. Si se quisiera hacer más debería agregarse más datos o relaciones entre ellos y aumentaría la complejidad o se introduciría funcionalidad redundante, lo que contradice la definición 5.

Supuesto 1: La Funcionalidad Potencial final de un SIGIDES (FPS) será nula. La FEV representa lo máximo que es posible hacer y también lo que se hará ya que fue obtenida a partir de las visiones de datos de los usuarios y si un usuario evocó en su memoria un dato, en algún momento solicitará algún requisito relacionado con él. Es decir que, al final del proceso de desarrollo, no se podrá agregar nuevas funcionalidades útiles para el negocio o algún usuario y no redundantes sin incluir nuevos datos y/o relaciones entre ellos. Esto es, la energía potencial de los datos se habrá transformado totalmente en información y funcionalidad no redundante.

Axioma 7: Un ICES contiene información útil como entrada de MEGEFs. No está determinado en que se transforma la energía

potencial de la estructura de los datos. Distintos usuarios pueden solicitar la funcionalidad e información que es posible obtener a partir de los datos disponibles de distintas maneras, pero el esfuerzo requerido para producirla, bajo condiciones similares, será el mismo.

Supuesto 2: Disminución de la variabilidad y estimación de resultados globales. Aunque no comprendamos, en profundidad, los complejos procesos internos e interacciones que ocurren durante el proceso de desarrollo, podemos observarlo y construir modelos que estimen, aceptablemente, su resultado global si disminuimos las fuentes de variabilidad mediante el uso de herramientas que automaticen el diseño de bases de datos, la generación del código, las tareas de integración y apoyen el trabajo de grupos reducidos con usuarios integrados a los mismos usando una metodología estándar.

Supuesto 3: Es posible determinar empíricamente el ICES a partir de la observación del papel que juegan en MEGEFs métricas e indicadores de complejidad de la información.

Definición 11: Especificación Formal de un SIGID (EFS) Especificación de un sistema de información que contiene elementos suficientes para, disponiendo del generador adecuado, generar, automáticamente, el código necesario para poder operar el Sistema en cualquier lenguaje de programación y plataforma de ejecución y no hace referencia a tablas ni elementos del esquema de base de datos relacional.

Definición 12: Especificación Formal de Datos de un SIGID (EFD) Especificación de alguna VDUS que contiene elementos sobre los datos que debe manejar el sistema y las relaciones existentes entre ellos suficientes para inferir, automáticamente, un esquema de base de datos relacional que represente la información contenida en la VDUS.

Definición 13: Herramienta de Especificación Formal de Datos de un SIGID (HEFD) Herramienta de software que apoya

las actividades de desarrollo y almacenamiento de una EFD.

Definición 14: Herramienta de Generación de Estructura de Datos Herramienta de software que, a partir de una EFD, genera un esquema de bases de datos relacionales que lo representa.

Definición 15: Herramienta de Especificación Formal y Desarrollo de un SIGID (HEFDS) Herramienta de software que apoya las actividades de desarrollo y mantenimiento de una EFS y cuyas capacidades incluyen al menos las siguientes:

1. especificar la VDUS
2. tratar atributos con igual nombre como el mismo, independientemente de donde se encuentren referenciados, e inferir, a partir de ellos, las relaciones entre las entidades representadas en el SIGID
3. generar un esquema de bases de datos relacionales representativo de la información presente en la VDUS
4. mantener las VDU que forman parte de la VDUS del SIGID permitiendo
 - eliminar una VDU
 - agregar una VDU
 - eliminar un atributo de una VDU
 - agregar un atributo a una VDU
 - especificar relaciones entre VDUs
5. especificar los procesos mediante referencias a los datos dependientes sólo del VDUS e independientes del esquema de bases de datos relacional
6. generar automáticamente un esquema de bases de datos relacionales representativo de la información contenida en la VDUS
7. generar automáticamente el código para los procesos en cualquier lenguaje y plataforma de ejecución, si se dispone del generador adecuado
8. evaluar automáticamente el impacto de un cambio en el VDUS sobre el esquema de bases de datos relacional, reportarlo y, en caso de que el cambio sea confirmado y consistente

- generar automáticamente el nuevo esquema de bases de datos relacionales que representa al nuevo VDUS
 - generar automáticamente el código necesario para migrar los datos del viejo esquema al nuevo
 - generar automáticamente el código para los procesos que se vieron impactados por el cambio
9. integrar automáticamente, siempre que no existan nombres repetidos, las especificaciones de varios SIGID para producir uno solo que representa la visión integrada de los sistemas individuales, reportando las inconsistencias si las hubiera

4. ALCANCE DE LA INVESTIGACIÓN Y ÁMBITO DE APLICACIÓN DE SUS RESULTADOS

4.1 Alcance.-

Esta investigación se basó en sistemas de gestión que realizan un procesamiento transaccional intensivo de datos desarrollados con bases de datos relacionales.

De estos quedan excluidos aquellos sobre los cuales se imponen restricciones a los tiempos de respuesta y/o que incluyen algoritmos de alta complejidad (SIGID, Definición 1).

De los SIGID, se estudiaron solamente los esenciales (SIGIDES, Definición 5), debido a que si se desarrolla funcionalidad o se ingresan datos redundantes el tiempo y esfuerzo no están acotados.

Se obtuvieron Modelos Empíricos Globales de Estimación Formal (MEGEFs, Definición 6) a partir de la observación post-mortem de SIGIDES desarrollados mediante una herramienta de especificación formal y desarrollo de SIGID (HEFDS), con las capacidades y características que se establecen en la Definición 15, por equipos reducidos con usuarios integrados a ellos

aplicando metodologías ágiles apoyadas por la herramienta de especificación.

Los indicadores de complejidad esencial de (ICES, Definición 7) se obtuvieron a partir de estos modelos.

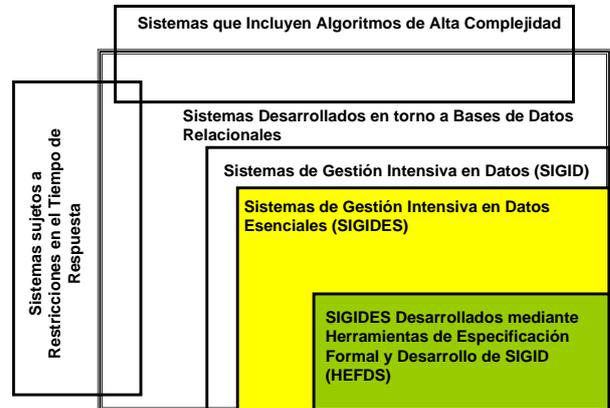


Figura 2 Alcance de la Investigación

4.2 Ámbito de Aplicación de los Resultados Obtenidos.-

Los modelos de estimación son aplicables a SIGIDES desarrollados bajo condiciones similares, con herramientas de similares características a la usada en los proyectos observados y que cumplen los axiomas y supuestos del apartado 3.1.

Los indicadores de complejidad esencial se formularon en base a métricas de complejidad para bases de datos relacionales obtenidas automáticamente a partir de las vistas de datos de los usuarios (VDUS, Definición 4) por lo que su utilidad se extiende a los SIGIDES en general, cualquiera sean las herramientas y metodologías utilizadas. Hemos desarrollado herramientas que permiten calcular los indicadores de complejidad automáticamente a partir de la VDUS de un SIGID o directamente del esquema de bases de datos relacionales. Si se parte del esquema de bases de datos relacionales puede haberse introducido elementos en el diseño de las bases de datos que reflejan puntos de vista de los desarrolladores, pero, siempre que los

sistemas a partir de los cuales se extraigan hayan sido desarrollados bajo condiciones similares, cada organización o grupo de desarrollo podrá vincular el ICES resultante con su propia experiencia.

5. DESARROLLO DE MODELOS E INFERENCIA DE LOS INDICADORES DE COMPLEJIDAD ESENCIAL

El desarrollo de los modelos de estimación se apoya en la reducción de la variabilidad mediante

1. la sistematización del desarrollo soportada por metodologías ágiles, grupos reducidos con usuarios integrados a ellos y herramientas de especificación formal que apoyan el proceso y se basan en modelos matemáticos de los datos construidos a partir de las visiones que tienen de los mismos los usuarios finales.
2. la obtención automática de métricas independientes de la tecnología y consideraciones de diseño a partir de la especificación
3. prescindencia del juicio experto.

Los indicadores de complejidad esencial (ICES) se obtuvieron a partir de estos modelos y la observación del papel que jugaron dentro de ellos las métricas de complejidad del modelo de datos.

5.1 Los Modelos a Desarrollar.- Para construir los modelos se realizó un análisis causal cuyos resultados se muestran en la Figura 3. Las variables respuesta tiempo y esfuerzo requeridos para construir un SIGIDES dependen de las variables independientes eficiencia de la organización (EF), volatilidad de los requisitos (VR), ritmo al que se ejecute el proyecto (RE) y complejidad del sistema que se desea construir (CES). Nuestra hipótesis es que existe una complejidad esencial de SIGIDES, que no depende de la tecnología usada para construirlo ni de consideraciones de diseño

(que es en última instancia una tecnología) y esta complejidad es conocida, implícitamente, por los usuarios finales del sistema.



Figura 3 Análisis Causal de los modelos a desarrollar

Deseamos construir modelos de estimación que, a partir de las variables independientes, estimen, en etapas muy tempranas del proceso de desarrollo y sin recurrir a juicio experto, las variables respuesta.

$$Esfuerzo = f(EF, VR, RE, CES) \quad \text{Ecuación 1}$$

$$Tiempo = f(EF, VR, RE, CES) \quad \text{Ecuación 2}$$

5.2 La Complejidad Esencial de un SIGIDES y Nuestras Hipótesis Iniciales.

El mayor desafío que debemos enfrentar en el desarrollo de estos modelos es encontrar indicadores de la complejidad esencial de un SIGIDES (Definición 7), basados en métricas presentes en etapas muy tempranas del proceso.

Por su misma esencialidad deberá ser posible definir estos indicadores a partir de elementos disponibles antes de que se haya adoptado ninguna tecnología o decisión de diseño particulares y serán, entonces, muy tempranos.

Un elemento disponible tempranamente son las visiones de los datos del sistema de sus usuarios finales. Estas visiones, son esenciales ya que están inspiradas por las características del negocio y no por tecnologías, consideraciones de diseño u

otros criterios ajenos al negocio y los problemas que se intenta resolver.

5.2.1 Nuestras Hipótesis de Trabajo.

1) Si sólo se desarrolla funcionalidad no redundante a partir de las vistas de datos de los usuarios finales y sus solicitudes, la complejidad esencial del sistema a desarrollar queda determinada por la integración de las visiones de datos de los usuarios del sistema y puede medirse a partir del esquema de base de datos relacional inferido automáticamente al integrar estas visiones.

2) Este indicador de complejidad esencial contiene información suficiente para ser útil como entrada de modelos de estimación de tiempo y esfuerzo.

3) Aunque no comprendamos en profundidad los complejos procesos internos e interacciones que ocurren durante el proceso de desarrollo, podemos observarlo y construir modelos que estimen aceptablemente su resultado global. En particular, si el desarrollo se realiza usando herramientas que automaticen la generación del código, la integración de módulos y apoyen el trabajo de grupos reducidos con usuarios integrados a los mismos usando una metodología estándar, se estará reduciendo las fuentes de variabilidad y será posible encontrar modelos útiles para la estimación temprana.

5.2.2 Indicadores de Complejidad Esencial y su Semántica.-

Admitiendo que existan indicadores de complejidad esencial para SIGIDES, surgen dos preguntas:

1. ¿cuáles son las métricas presentes tempranamente que pueden contribuir a definirlos?
2. ¿qué semántica tienen?

Estas no son preguntas independientes, nuestra definición del ICES (Definición 7) y nuestro Axioma 7 lo vincula al papel que juega en modelos de estimación.

Para responder la pregunta 1 responderemos primero la pregunta de cuáles son métricas tempranas de complejidad de un SIGIDES, construiremos modelos de estimación a partir de ellos y determinaremos el ICES a partir del rol que juegan en estos modelos las métricas de complejidad.

De acuerdo a nuestra Hipótesis 1, existe información sobre la complejidad en las visiones de datos de los usuarios finales y esta puede medirse a partir del esquema de bases de datos relacionales inferido a partir de su integración.

Tenemos, entonces, que medir la complejidad de un esquema relacional. Estas métricas son muy tempranas si el esquema relacional es inferido automáticamente a partir del conjunto de visiones de datos de los usuarios finales del sistema.

En este punto resultaron fundamentales para nuestra investigación las métricas propuestas por Calero, Piattini, Polo y Ruiz (1999)[68]: Número de Tablas (NT), Grado de referencialidad constituido por el número de claves foráneas (RD), Número de Atributos Únicos (NA) y Profundidad del Árbol Referencial (DRT). Estos autores analizan detalladamente en su publicación las propiedades formales de las métricas que proponen y destacan que se requiere un sustento empírico para las mismas. Nosotros trataremos, a partir de ellas de definir un indicador de complejidad esencial para SIGIDES y aportar, dentro de las posibilidades de la población estudiada, esta evidencia empírica.

La **Figura 4** muestra el modelo conceptual de la investigación luego de formuladas las hipótesis de trabajo y tomando en cuenta las métricas de complejidad de bases de datos relacionales propuestas por el grupo ALARCOS.

6. LOS PROYECTOS OBSERVADOS

Se observaron post mortem SIGIDES (Definición 5) desarrollados por grupos

reducidos de entre 2 y 5 personas con una herramienta de especificación formal.

Se extrajeron automáticamente métricas a partir de la especificación y se solicitó a los gerentes de los proyectos que estimaran en alta, media o baja la volatilidad de los requisitos e informaran sobre el tiempo y esfuerzo totales.

Como deseamos desarrollar modelos de estimación temprana, se tomó como indicador del ritmo de ejecución el esfuerzo medio del 10% inicial del tiempo de duración del proyecto.

La eficiencia de la organización se consideró constante dentro de la muestra por tratarse de grupos similares aplicando una metodología estándar soportada por la herramienta de especificación.

Las métricas extraídas a partir de la especificación fueron obtenidas, en todos los casos, automáticamente por lo que son exactas y confiables.

También se observaron proyectos correspondientes a talleres de estudiantes universitarios que desarrollaron la misma funcionalidad. Para estos proyectos no se dispone de registros de esfuerzo. Sin embargo, permiten apreciar que las métricas de complejidad de bases de datos relacionales son similares. La información recopilada se presenta en la **Tabla 3**.

7 LOS MODELOS FORMALES DE ESTIMACIÓN MUY TEMPRANA DE TIEMPO Y ESFUERZO

Mediante regresión lineal sobre las variables normalizadas con logaritmo se obtuvieron modelos de estimación de tiempo y esfuerzo.

7.1 Modelos de Estimación Temprana Basados en las Métricas de Complejidad de Bases de Datos Relacionales definidas por el grupo ALARCOS

$$ESFUERZO = 0,01 \text{EMI}^{0,235} \text{VR}^{0,512} \text{DRT}^{-2,234} \text{RD}^{-0,077} \text{NA}^{1,951} \text{NT}^{-0,516}$$

Ecuación 3

$$\text{TIEMPO} = 0,014 \text{EMI}^{-0,322} \text{VR}^{0,372} \text{DRT}^{-2,467} \text{RD}^{-0,209} \text{NA}^{1,84} \text{NT}^{-0,271}$$

Ecuación 4

7.2 Indicadores de Complejidad Esencial de la Estructura de los Datos

De la participación de las métricas de complejidad de bases de datos relacionales en los modelos de estimación de tiempo y esfuerzo se deducen respectivamente los indicadores ICEDE e ICEDT.

$$\text{ICEDE} = \text{DRT}^{-2,234} \text{RD}^{-0,077} \text{NA}^{1,951} \text{NT}^{-0,516}$$

Ecuación 5

$$\text{ICEDT} = \text{DRT}^{-2,467} \text{RD}^{-0,209} \text{NA}^{1,84} \text{NT}^{-0,271}$$

Ecuación 6

7.3 Modelos de Estimación muy Temprana de Tiempo y Esfuerzo Basados en los Indicadores de Complejidad Esencial

Se desarrollaron modelos de estimación temprana de tiempo y esfuerzo en base a los indicadores de complejidad esencial

$$ESFUERZO = 0,01 \text{EMI}^{0,235} \text{VR}^{0,512} \text{ICEDE}$$

Ecuación 7

$$ESFUERZO = 0,007 \text{EMI}^{0,68} \text{VR}^{0,668} \text{ICEDT}^{0,963}$$

Ecuación 8

$$\text{TIEMPO} = 0,021 \text{EMI}^{-0,778} \text{VR}^{0,216} \text{ICEDE}^{1,022}$$

Ecuación 9

$$\text{TIEMPO} = 0,014 \text{EMI}^{-0,322} \text{VR}^{0,372} \text{ICEDT}$$

Ecuación 10

8. EVALUACIÓN DE LA PRECISIÓN Y CONSISTENCIA DE LOS MODELOS Y VALIDACIÓN DE LOS INDICADORES DE COMPLEJIDAD

ESENCIAL

Con la finalidad de estudiar si la información disponible tempranamente es suficiente para estimar (1) se desarrollaron modelos de estimación en base a los indicadores de complejidad esencial y la información disponible post mortem sobre la complejidad de la estructura del flujo de control, (2) se evaluó la precisión y consistencia de los modelos mediante los criterios de Conte et al (1986) [69] y (3) se comprobó estadísticamente que con un 99% de probabilidad puede considerarse que las diferencias entre los valores de las variables dependientes tiempo y esfuerzo observados, sus estimaciones tempranas y sus estimaciones post mortem pueden atribuirse al azar.

8.1 Modelos de estimación Post Mortem.

Se desarrollaron modelos de estimación de tiempo y esfuerzo basados en las mismas variables de entrada que los anteriores, agregando los indicadores de complejidad de la estructura de control de los procesos presentes post mortem: cantidad de estructuras de control repetitivas (FOR), cantidad de alternativas (IF), cantidad de estructuras repetitivas anidadas (FORAN) y cantidad de alternativas anidadas (IFAN).

La finalidad del desarrollo de estos modelos es verificar que el ICED contiene información suficiente, a efectos prácticos, para estimar tempranamente y que, si bien la información presente post mortem aumenta la

$$ESFUERZO = 0,015 EMI^{0,306} VR^{0,61} FOR^{-0,003} IF^{-0,146} FORAN^{0,048} IFAN^{0,084} ICEDE^{0,939}$$

Ecuación 11

$$ESFUERZO = 0,016 EMI^{0,598} VR^{0,736} FOR^{0,178} IF^{-0,187} FORAN^{-0,039} IFAN^{0,07} ICEDT^{0,815}$$

Ecuación 12

$$TIEMPO = 0,016 EMI^{-0,72} VR^{0,204} FOR^{-0,207} IF^{0,056} FORAN^{0,151} IFAN^{-0,017} ICEDE^{1,108}$$

Ecuación 13

$$TIEMPO = 0,017 EMI^{-0,374} VR^{0,351} FOR^{0,005} IF^{0,009} FORAN^{0,047} IFAN^{-0,032} ICEDT^{0,964}$$

Ecuación 14

En la Tabla 3 se presentan los datos recopilados y los valores estimados.

8.2 Evaluación de la Calidad de los Modelos de Estimación Obtenidos.

Para medir el desempeño estadístico de los modelos puede usarse el coeficiente de determinación R².

Conte (1986) [69] propone medir la magnitud del error relativo medio (medium magnitude of relative error, MMRE) y la capacidad de predicción a un nivel k (PRED (k)).

El error relativo RE se define como la desviación relativa a la magnitud del valor real observado. La magnitud del error relativo (MRE, magnitud of relative error) es el valor absoluto del error relativo.

$$RE = \frac{V_{real} - V_{est}}{V_{real}} \quad \text{Donde } V_{real} = \text{Valor observado, } RE = \text{Error relativo}$$

$$MRE = |RE| \quad V_{est} = \text{Valor estimado}$$

$$MRE = \text{magnitud del error relativo al valor real}$$

precisión y consistencia de los modelos, este aumento no es significativo.

El MMRE (medium magnitude of relative error) se calcula como el promedio de las MRE de cada una de las observaciones

$$MMRE = \frac{\sum_{i=1}^n MRE_i}{n} \quad \text{Donde } MRE_i = \text{magnitud del error relativo de la estimación de la observación } i,$$

$n = \text{número total de observaciones}$

La capacidad de predicción a un nivel k se calcula

$$PRED(k) = \frac{n_k}{n} \quad \text{Donde } k = \text{error relativo en valor absoluto aceptable (ej 10\%, 20\%)}$$

$$n_k = \text{número de observaciones con } |RE| < k$$

$$n = \text{número total de observaciones}$$

Calificación	Consistencia	Precisión
Excelente	MMRE \leq 0.20	PRED(20) \geq 0.80
Bueno	MMRE \leq 0.25	PRED(25) \geq 0.75
Aceptable	MMRE \leq 0.30	PRED(30) \geq 0.70
Pobre	MMRE $>$ 0.30	PRED(30) $<$ 0.70

Tabla 2 Criterios de Conte et al (1986)

Los criterios para calificar la precisión y consistencia de los modelos propuestos por Conte et al (1986) [69] se presentan en la Tabla 2.

En la Tabla 4 se presenta un resumen de la precisión y consistencia de los modelos que incluye estadísticos de resumen, los valores de PRED (K) y la cantidad de observaciones que caen dentro de cada rango de error para todos los casos y los reservados para contraste de los modelos.

8.3 Validación Empírica de los Modelos e Indicadores de Complejidad Esencial

En este capítulo se validan empíricamente los modelos de estimación temprana de tiempo y esfuerzo y los indicadores de complejidad de la estructura de los datos ICEDE e ICEDT. Deseamos responder la pregunta:

¿Con que nivel de confianza puede afirmarse que las diferencias entre las mediciones finales, las estimaciones tempranas y las estimaciones post mortem se deben al azar?

Para ello se realizó (1) un análisis exploratorio de los datos mediante estadísticos de resumen de los datos y gráficos, (2) pruebas no paramétricas de igualdad de medias relacionadas, (3) la prueba de la T para muestras emparejadas con la finalidad de estudiar si existen diferencias apreciables, desde el punto de vista práctico, entre las observaciones, las estimaciones tempranas y las estimaciones post mortem, los errores que se cometen al estimar tempranamente y los errores que se cometen en la estimación post mortem y (4) análisis de varianza.

Los estudios fueron realizados para el conjunto de la población, los valores seleccionados para desarrollar los modelos y los valores reservados para contrastarlos.

Se concluye que con un 99% de confianza se puede afirmar que las diferencias entre las observaciones, las estimaciones tempranas y las estimaciones post mortem pueden atribuirse al azar. En la Tabla 5 se presentan

las correlaciones de muestras relacionadas para todos los casos y los casos de contraste y se aprecia que son muy altas. En la Tabla 6 se presenta la prueba de la T para muestras relacionadas para todos los casos y los casos de contraste: Los intervalos que con una 99% de confianza contienen la media de las diferencias contienen al 0, la significación no es menor que 0,005 y los valores de t, en ambos casos, son adecuados.

9 CONCLUSIONES

Se definieron dos ICES (indicadores de complejidad esencial, Definición 7) para SIGIDES (Sistema de Información de Gestión Intensiva de Datos Esencial, Definición 1 y Definición 5) a partir de la participación de las métricas de complejidad para bases de datos relacionales propuestas por el grupo ALARCOS (2000), en modelos de estimación muy temprana de esfuerzo (ICEDE, Ecuación 5) y tiempo (ICEDT, Ecuación 6).

Las métricas de complejidad de bases de datos relacionales en que se basan estos indicadores fueron obtenidas automáticamente a partir de las vistas de los datos del sistema que tienen los usuarios finales (VDUS, Definición 4) y

1. son muy tempranas
2. no contienen información relacionada con detalles de diseño, elementos que a criterio de los diseñadores y/o ingenieros de software pudieran ser importantes, consideraciones relacionadas con la tecnología, e incluso la forma en que los usuarios finales visualizan los datos excepto el hecho de que los datos van a ser procesados mediante un sistema de gestión de bases de datos relacionales
3. no requieren juicio experto y
4. son independientes del conjunto de visiones de datos de usuario del que se parta (VDUS, Definición 4).

Estas propiedades de las métricas en que se basan se transfieren a los indicadores

convirtiéndolos en indicadores formales, independientes del juicio experto y consideraciones relativas a diseño, tecnología, la forma en que los usuarios finales visualizan los datos u otros factores no esenciales del sistema.

Estos indicadores de complejidad esencial correlacionan muy bien con tiempo y esfuerzo. Esta correlación es significativa ya que

1. está apoyada por numerosos análisis causales [11,12,55, 56,57], (Figura 3 y Figura 4) y
2. cumplen con uno de los criterios más importantes para una métrica o indicador [58,61,68,70,71,72,73]: permitieron desarrollar modelos empíricos formales estáticos globales de estimación muy temprana de tiempo y esfuerzo que no requieren juicio experto y pueden ser automatizados [11,12,57] y, según los criterios de Conte et al (1986)[69], califican como excelentes para la población observada, excepto para el caso de la estimación temprana de esfuerzo en base a ICEDT que califica como bueno y no contradicen las observaciones empíricas.

Se desarrollaron dos modelos de estimación muy temprana de tiempo y dos de esfuerzo, en los que la influencia de la complejidad de la estructura de los datos está representada por los indicadores ICEDE e ICEDT respectivamente (METEICEDE, Ecuación 7; METEICEDT, ecuación 8 ; METTICEDE, Ecuación 9 y METTICEDT, Ecuación 10).

El modelo de estimación temprana de esfuerzo basado en el ICEDE permite estimarlo con un error relativo máximo del 25,61% y un error relativo promedio del 10,62%. El basado en el ICEDT comete un error relativo máximo del 77,47% y un error relativo promedio del 16,62%.

El modelo de estimación temprana de tiempo en base al ICEDE permite estimarlo con un error relativo máximo del 34,22% y un error relativo promedio del 12,15%. El basado en

el ICEDT comete un error relativo máximo del 27,64% y un error relativo promedio del 8,42%.

Estos resultados sugieren la conveniencia de considerar ambos indicadores, pero, si hubiera que elegir uno, el que se muestra más estable es el ICEDE.

Esto no es contra intuitivo, ya que, para una complejidad esencial dada, habrá un esfuerzo mínimo necesario para desarrollar el sistema correspondiente, aún cuando intervienen otros factores como la rapidez de ejecución del proyecto y la volatilidad de los requisitos. En cambio, el tiempo para desarrollarlo puede ser algo más elástico y dependerá, en buena medida, de la forma en que se ejecute el proyecto.

Los indicadores de complejidad muestran que la mayor contribución al aumento de la complejidad es efectuada por el número de atributos únicos (NA) que figura elevado al cuadrado y que la profundidad del árbol referencial (DRT) la disminuye. Esto sugiere que, cuando aumenta la estructuración de la información disminuye la dificultad para comprenderla.

Se desarrollaron modelos de estimación que toman en cuenta la información sobre la complejidad de la estructura del flujo de control disponible post mortem y se verificó empíricamente [56] que las diferencias entre: (1) las mediciones finales de tiempo y esfuerzo, (2) sus estimaciones tempranas y (3) sus estimaciones post mortem pueden ser consideradas con un 99% de confianza debidas al azar y, a efectos prácticos, no existe diferencia entre las estimaciones tempranas y las post mortem para la población observada.

Esto significa que la información disponible tempranamente, a efectos prácticos, no difiere sustancialmente de la disponible post mortem. Así, se confirma o al menos no se contradice (1) lo que ya estableciera DeMarco (1982) [74] sobre que los sistemas data-strong pueden ser estimados a partir de los datos, (2) la bondad de las métricas de complejidad de

bases de datos relacionales propuestas por el grupo ALARCOS, aportando, además, el soporte empírico que sus propios autores estimaron necesario para complementar las propiedades formales ya demostradas por ellos, (3) la existencia de una complejidad esencial para los SIGIDES que depende de los datos esenciales que estos sistemas deben procesar (VDUS) y que los indicadores ICEDE e ICEDT explicitan en función de las métricas de complejidad de las bases de datos relacionales inferidas automáticamente a partir de ellos, (4) los axiomas, hipótesis y supuestos en que se basa esta investigación.

Estos resultados impactan (1) el estado del arte en modelos de estimación empíricos estáticos y globales, permitiendo realizar automáticamente y, en etapas iniciales del ciclo de vida, estimaciones con niveles de precisión y consistencia que las hacen útiles para la planificación y negociación del alcance de proyectos de desarrollo de SIGIDES con herramientas de especificación formal, (2) definiendo indicadores de complejidad que pueden ser calculados automáticamente y muy tempranamente a partir de las vistas de datos de los usuarios finales del sistema a construir aún cuando las metodologías y herramientas aplicadas para su desarrollo difieran de las usadas para construir los sistemas observados y (3) permitiendo que cualquier organización pueda relacionar estos indicadores con su propia experiencia .

Como es frecuente en los estudios empíricos, el tamaño de la población observada, limita la validez de las conclusiones extraídas; pero, al mismo tiempo, refleja nuestra decisión y compromiso de procesar datos sobre cuya calidad tuviéramos adecuadas seguridades.

10. LÍNEAS DE TRABAJO FUTURAS

Desarrollo de modelos de estimación para SIGIDES desarrollados con herramientas y metodologías distintas de las que se usaron en los sistemas observados. En este sentido

representaría un gran aporte que el ISBSG incluyera en su repositorio las métricas de complejidad de bases de datos propuestas por el grupo ALARCOS que conforman la base de los indicadores de complejidad esencial, la Volatilidad de los Requerimientos y el esfuerzo medio inicial.

Desarrollo de Modelos detallados. Los modelos fueron obtenidos a partir de la observación post mortem de proyectos y las mediciones de tiempo y esfuerzo corresponden al total insumido.

Es importante obtener modelos detallados para poder estimar el esfuerzo requerido por cada tarea a lo largo del ciclo de vida. Con esta finalidad se desarrolló en el Centro de Cómputos del Ministerio de Transporte y Obras Públicas de Uruguay un ambiente integrado de desarrollo que se encuentra en su versión beta y permite tomar, en tiempo real, las métricas del PSP [75,76] y relacionar los trabajos con los requisitos y sus cambios. Así, podrían medirse en tiempo real las variables independientes de los modelos.

Si se desarrollara una cantidad suficiente de sistemas con esta herramienta podrían construirse modelos detallados.

Modelos Formales de Estimación de Riesgo. Si se obtuviera información sobre una población suficientemente amplia como para deducir una distribución para el tiempo y esfuerzo podría estimarse la probabilidad de concluir el proyecto en un tiempo determinado, dada la complejidad del sistema a construir, la volatilidad de los requisitos esperada y la eficiencia de la organización e inversamente, cuál debería ser el tiempo asignado al proyecto para tener determinada probabilidad de cumplir lo planificado. Algo que Nogueira [10] ya hizo para sistemas de tiempo real.

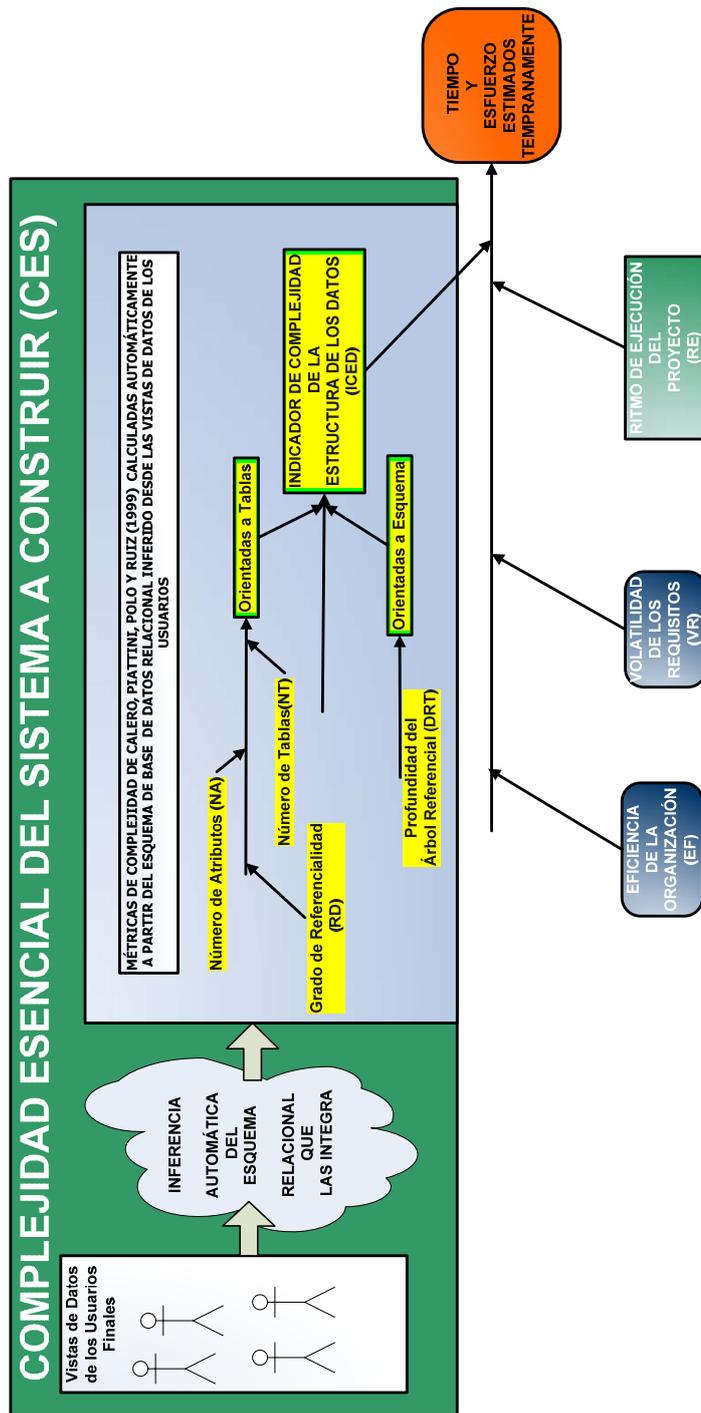


Figura 4. Modelo Conceptual de los Modelos de Estimación Formales tomando en cuenta las métricas de complejidad de bases de datos relacionales propuestas por el grupo ALARCOS

Tabla 4 Precisión y Consistencia de los Modelos de Estimación Temprana

Correlaciones de muestras relacionadas (todos los casos)				Correlaciones de muestras relacionadas (Casos no seleccionados)					
	N	Correlación	Sig.		N	Correlación	Sig.		
Par 1	ESFUERZO y METEICEDE	20	,983 ^a	,000 ^b	Par 1	ESFUERZO y METEICEDE	6	,984 ^a	,000 ^b
Par 2	ESFUERZO y METEICEDT	20	,972 ^a	,000 ^b	Par 2	ESFUERZO y METEICEDT	6	,950 ^a	,004 ^b
Par 3	ESFUERZO y MEPMEICEDE	20	,984 ^a	,000 ^b	Par 3	ESFUERZO y MEPMEICEDE	6	,966 ^a	,002 ^b
Par 4	ESFUERZO y MEPMEICEDT	20	,968 ^a	,000 ^b	Par 4	ESFUERZO y MEPMEICEDT	6	,920 ^a	,009 ^b
Par 5	METEICEDE y MEPMEICEDE	20	,986 ^a	,000 ^b	Par 5	METEICEDE y MEPMEICEDE	6	,988 ^a	,000 ^b
Par 6	METEICEDT y MEPMEICEDE	20	,967 ^a	,000 ^b	Par 6	METEICEDT y MEPMEICEDE	6	,962 ^a	,002 ^b
Par 7	eaMETEICEDE y eaMEPMEICEDE	20	,567	,009	Par 7	eaMETEICEDE y eaMEPMEICEDE	6	,831 ^a	,040 ^b
Par 8	eaMETEICEDT y eaMEPMEICEDT	20	,433	,056	Par 8	eaMETEICEDT y eaMEPMEICEDT	6	,718 ^a	,108
Par 9	erMETEICEDE y erMEPMEICEDE	20	,335	,149	Par 9	erMETEICEDE y erMEPMEICEDE	6	,213	,685
Par 10	erMETEICEDT y erMEPMEICEDT	20	,172	,469	Par 10	erMETEICEDT y erMEPMEICEDT	6	,405	,426
Par 11	VAerMETEICEDE y VAerMEPMEICEDE	20	,524	,018	Par 11	VAerMETEICEDE y VAerMEPMEICEDE	6	,564	,244
Par 12	VAerMETEICEDT y VAerMEPMEICEDT	20	,304	,193	Par 12	VAerMETEICEDT y VAerMEPMEICEDT	6	,063	,906

a. La correlación es alta

b. La significación es baja

a. La correlación es alta

b. La significación es < 0,05

Tabla 5 Correlaciones de Muestras Relacionadas esfuerzo y sus estimaciones para todos los casos y Casos de Contraste

Prueba de muestras relacionadas para todos los casos									
		Diferencias relacionadas					t	gl	Sig. (bilateral)
		Media	Desviación típ.	Error típ. de la media	99% Intervalo de confianza para la diferencia				
					Inferior	Superior			
Par 1	ESFUERZO - METEICEDE	-,494	4,325	,967	-3,261 ^a	2,272 ^a	-,511 ^b	19 ^b	,62 ^c
Par 2	ESFUERZO - METEICEDT	-1,025	5,438	1,216	-4,503 ^a	2,454 ^a	-,843 ^b	19 ^b	,41 ^c
Par 3	ESFUERZO - MEPMEICEDE	,178	4,181	,935	-2,497 ^a	2,853 ^a	,190 ^b	19 ^b	,85 ^c
Par 4	ESFUERZO - MEPMEICEDT	1,031	5,922	1,324	-2,757 ^a	4,820 ^a	,779 ^b	19 ^b	,45 ^c
Par 5	METEICEDE - MEPMEICEDE	,672	3,960	,886	-1,861 ^a	3,206 ^a	,759 ^b	19 ^b	,46 ^c
Par 6	METEICEDT - MEPMEICEDE	2,056	6,061	1,355	-1,822 ^a	5,934 ^a	1,517 ^b	19 ^b	,15 ^c
Par 7	eaMETEICEDE - eaMEPMEICEDE	,672	3,960	,886	-1,861 ^a	3,206 ^a	,759 ^b	19 ^b	,46 ^c
Par 8	eaMETEICEDT - eaMEPMEICEDT	2,056	6,061	1,355	-1,822 ^a	5,934 ^a	1,517 ^b	19 ^b	,15 ^c
Par 9	erMETEICEDE - erMEPMEICEDE	,040	,133	,030	-,045 ^a	,125 ^a	1,348 ^b	19 ^b	,19 ^c
Par 10	erMETEICEDT - erMEPMEICEDT	,118	,293	,066	-,070 ^a	,305 ^a	1,798 ^b	19 ^b	,09 ^c
Par 11	VAerMETEICEDE - VAerMEPMEICEDE	,032	,068	,015	-,012 ^a	,075 ^a	2,101 ^b	19 ^b	,05 ^c
Par 12	VAerMETEICEDT - VAerMEPMEICEDT	,037	,212	,047	-,098 ^a	,173 ^a	,785 ^b	19 ^b	,44 ^c

a. El intervalo en que con un 99% de confianza se encuentra la diferencia contiene al 0 para todos los pares

b. $t(0,005,19)=2,861$. Ninguno de los valores de t en valor absoluto supera este valorc. La significación es $\geq 0,05$ en todos los casos

Prueba de muestras relacionadas (Casos no seleccionados)									
		Diferencias relacionadas					t	gl	Sig. (bilateral)
		Media	Desviación típ.	Error típ. de la media	99% Intervalo de confianza para la diferencia				
					Inferior	Superior			
Par 1	ESFUERZO - METEICEDE	-2,219	4,718	1,926	-9,986 ^a	5,548 ^a	-1,152 ^c	5	,30 ^b
Par 2	ESFUERZO - METEICEDT	-4,473	8,292	3,385	-18,123 ^a	9,177 ^a	-1,321 ^c	5	,24 ^b
Par 3	ESFUERZO - MEPMEICEDE	,332	7,101	2,899	-11,357 ^a	12,021 ^a	-,114 ^c	5	,91 ^b
Par 4	ESFUERZO - MEPMEICEDT	3,464	10,428	4,257	-13,702 ^a	20,629 ^a	,814 ^c	5	,45 ^b
Par 5	METEICEDE - MEPMEICEDE	2,550	4,123	1,683	-4,236 ^a	9,337 ^a	1,515 ^c	5	,19 ^b
Par 6	METEICEDT - MEPMEICEDT	7,937	7,305	2,982	-4,088 ^a	19,961 ^a	2,661 ^c	5	,04 ^b
Par 7	eaMETEICEDE - eaMEPMEICEDE	2,550	4,123	1,683	-4,236 ^a	9,337 ^a	1,515 ^c	5	,19 ^b
Par 8	eaMETEICEDT - eaMEPMEICEDT	7,937	7,305	2,982	-4,088 ^a	19,961 ^a	2,661 ^c	5	,04 ^b
Par 9	erMETEICEDE - erMEPMEICEDE	,124	,178	,072	-,168 ^a	,417 ^a	1,716 ^c	5	,15 ^b
Par 10	erMETEICEDT - erMEPMEICEDT	,368	,385	,157	-,266 ^a	1,002 ^a	2,339 ^c	5	,07 ^b
Par 11	VAerMETEICEDE - VAerMEPMEICEDE	-,008	,073	,030	-,129 ^a	,113 ^a	-,261 ^c	5	,80 ^b
Par 12	VAerMETEICEDT - VAerMEPMEICEDT	-,048	,328	,134	-,588 ^a	,492 ^a	-,359 ^c	5	,73 ^b

a. Los intervalos de confianza contienen al 0.

b. La significación no es < 0,005

c. $t(0,005,5)=4,032$ y ninguno de los valores de t en valor absoluto superan este valor

Tabla 6 Prueba T de muestras relacionadas esfuerzo y sus estimaciones para todos los casos y casos de Contraste

11.REFERENCIAS BIBLIOGRÁFICAS

- [1] Kavoussanakis, K., Sloan, T. THE UNIVERSITY OF EDINBURGH UKHEC Report on Software Estimation Document Identifier: EPCC-UKHEC 0.1
- [2] McConnell, S. Rapid development: taming wild software schedules. Microsoft Press, 1996.
- [3] [3] Construx Software Inc. <http://www.construx.com/estimate>.
- [4] [4] Guidelines for Successful Acquisition and Management of Software-Intensive Systems: Weapon Systems, Command and Control Systems, Management Information Systems Version 3.0, Fall 1999, DEPARTMENT OF THE AIR FORCE, Software Technology Support Center
- [5] Guidelines for Successful Acquisition and Management of Software-Intensive Systems: Weapon Systems, Command and Control Systems, Management Information Systems Version 4.1, Condensed Handbook ,2003, DEPARTMENT OF THE AIR FORCE, Software Technology Support Center
- [6] Sommerville, I. Software Engineering, Sixth Edition. Addison-Wesley Publishers Limited, 2001.
- [7] Briand L., Khaled, E., Morasca, S Theoretical and Empirical Validation of Software Product Measures. This paper appears as Technical Report number ISERN-95-03, International Software Engineering Research Network, 1995.
- [8] W.S.Humphrey. Your Date or Mine. In The Watts New Collection. Software Engineering Institute, Carnegie Mellon University, <http://interactive.sei.cmu.edu/>, 2001.
- [9] Boehm, B. Sullivan, K. Software Economics: A Roadmap. University of Southern California University of Virginia Department of Computer Science Thornton Hall Los Angeles, CA 90089-0781 USA Department of Computer Science. <http://www.cs.virginia.edu/~sullivan/publication/s/ICSE-2000-Roadmap.PDF>
- [10] Nogueira, J. A Formal Model for Risk Assessment in Software Projects. PhD Dissertation. Naval Postgraduate School, Monterrey, California. September, 2000.
- [11] Salvetto, P., Nogueira J. y Segovia, J Modelos Automatizables de Estimación muy Temprana del Tiempo y Esfuerzo de Desarrollo de Software de Gestión Aceptado XXX Conferencia Latinoamericana de Informática XXX CONFERENCIA LATINOAMERICANA DE CIENCIAS DE LACOMPUTACIÓN (CLEI 2004)
- [12] Salvetto, P., Nogueira J. y Segovia, J. Gestión de Cambios Apoyada por Modelos Formales de Estimación de Tiempo y Esfuerzo. 4 tas JORNADAS IBEROAMERICANAS DE INGENIERÍA DE SOFTWARE E INGENIERÍA DEL CONOCIMIENTO JIISIC'04. Madrid, España.
- [13] Latorres, Salvetto, Nogueira, Larreborges. Una herramienta de apoyo a la gestión del proceso de desarrollo de software. Proceedings CACIC 2003. La Plata, 2003.
- [14] Brooks, F. The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Addison Wesley, ISBN 0-201-83595-9.
- [15] Jenkins, A.M., J.D. Naumann, and J.C. Wetherbe, Empirical Investigation of Systems Development Practices and Results. Information & Management, 1984. 7: p. 73-82.
- [16] Heemstra, F.J. and Kusters. R.J. Controlling Software Development Costs: A Field Study. in International Conference on Organisation and Information Systems. 1989. Bled, Yugoslavia.
- [17] Heemstra, F.J. Software cost estimation, Information and Software Technology (34:10) 1992, pp 627-639.
- [18] Bergeron, F. and J.-Y. St-Arnaud, Estimation of Information Systems Development Efforts: A Pilot Study. Information & Management, 1992. 22: p. 239-254.
- [19] Gibbs, W. Software's Chronic Crisis TRENDS IN COMPUTING Scientific American; September 1994; Page 86.
- [20] Lederer, A.L. and J. Prasad, Causes of Inaccurate Software Development Cost Estimates. Journal of Systems and Software, 1995(31): p. 125-134
- [21] Ropponen, Janne and Lyytinen, K .Components of Software Development Risk:How to Address Them? A Project Manager Survey. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 26, NO. 2, FEBRUARY 2000, pages 98-112
- [22] The Standish Group, 1994 THE CHAOS REPORT http://www.standishgroup.com/quarterly_reports/
- [23] The Standish Group, 2004 THIRD QUARTER RESEARCH REPORT http://www.standishgroup.com/quarterly_reports/
- [24] Moløkken, K. and Jørgensen, M. A Review of Surveys on Software Effort Estimation IEEE International Symposium on Empirical Software Engineering (ISESE 2003) 2003. September 30 -

- October 1, Rome, Italy. Page 223-230. IEEE Computer Society. ISBN 0-7695-2002-2. http://www.ifi.uio.no/~isu/INCO/Papers/Review_final8.pdf
- [25] Sauer, C. and Cuthbertson, C. The State of IT Project Management in the UK 2002-2003 Templeton College, University of Oxford. Survey sponsored by the French Thornton Partnership. www.cw360ms.com/pmsurveyresults/surveyresults.pdf
- [26] Jensen, R. Extreme Software Cost Estimating This article was awarded the Outstanding Software Paper at the International Society of Parametric Analysts 2003 Annual Conference in Orlando, Fla. CROSSTALK Jan 2004 Issue
- [27] Kitchenham, B., Software Metrics: Measurement for Software Process Improvement. 1996, Oxford: NCC Blackwell.
- [28] Boehm, B. Software Engineering Economics. Prentice Hall, 1981.
- [29] Morgan, J. A Software Development Cost Estimation Model for Higher Level Language Environments ABAS ACADEMY OF BUSINESS & ADMINISTRATIVE SCIENCE <http://www.sba.muohio.edu/abas/2003/proceedings.html> June 24-26, 2003 Vancouver, BC, Canada July 11-13, 2003 Brussels, Belgium
- [30] Boehm, B. et al. Software Cost Estimation with COCOMO II. Prentice Hall, 2000.
- [31] Presman, R. and Herron, S. R. Software Shock. Dorset House, 1991.
- [32] Presman, R. Software Engineering, Fifth Edition.
- [33] Presman, R. Software Engineering, Sixth Edition.
- [34] Boehm, B. Tutorial: Software Risk Management. IEEE Computer Society Press. 1989.
- [35] Lum, K. Handbook for Software Cost Estimation, Jet Propulsion Laboratory Pasadena, California. The research described in this report was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. May 30, 2003
- [36] Phan, D., Information Systems Project Management: an Integrated Resource Planning Perspective Model, in Department of Management and Information Systems. 1990, Arizona: Tucson
- [37] Moløkken, K., Jørgensen, M., Tanilkan, S., Gallis, H., Lien, A. Hove, S. Survey on Software Estimation in the Norwegian Industry. METRICS 2004.
- [38] Moløkken, K. and Jørgensen, M., Tanilkan, S., Gallis, H., Lien, A. Hove, S. Does Use of Development Model Affect Estimation Accuracy and Bias? Product Focused Software Process Improvement: 5th International Conference, PROFES 2004, Kansai Science City, Japan, April 5-8, 2004. Proceedings ISBN: 3-540-21421-6. pp. 17-29. Springer-Verlag
- [39] Moløkken, K. and Jørgensen, M., Group Processes in Software Effort Estimation Journal of Empirical Software Engineering, 2004, Volume 9, Issue 4, pp 315-334.
- [40] Moløkken, K. Effort and Schedule Estimation of Software Development Projects. Thesis submitted for the degree of PhD. Department of Informatics Faculty of Mathematics and Natural Sciences University of Oslo August 2004.
- [41] Acuña, S. and Juristo, N. A process model applicable to software engineering and knowledge engineering. International Journal of Software Engineering and Knowledge Engineering 9(5): 663- (1999)
- [42] Juristo, N. and Morant, J. L. COMMON FRAMEWORK FOR THE EVALUATION PROCESS OF KBS AND CONVENTIONAL SOFTWARE Knowledge-Based Systems Journal. Vol 11, no. 2, pp. 145-160. Elsevier, 1998.
- [43] Alonso, F. Juristo N. and Pazos, J. Trends in life-cycle models for SE and KE: proposal for a spiral-conical life-cycle model approach", Int. J. Software Engineering and Knowledge Engineering. 5, 3 (1995) 445-465.
- [44] Gómez, A. Juristo, N., Montes, C. Pazos, J.. Ingeniería del Conocimiento. Colección de Informática Editorial Centro de Estudios Ramón Aceres S.A. ISBN 84-8004-269
- [45] Boehm, B., Turner, R. Balancing Agility and Discipline: A Guide for the Perplexed Addison-Wesley 2004 ISBN 0-321-18612-5
- [46] Boehm, B., Abts, C., Chulani, S. Software Development Cost Estimation Approaches – A Survey. University of Southern California Ann. Software Eng. 10: 177-205 (2000)
- [47] Charette, R., Adams, K., & White, M. Managing Risk in Software Maintenance. IEEE Software, May-June, 1997.
- [48] Boehm, B., Fairley, R. Software Estimation Perspectives. IEEE SOFTWARE November/December 2000, pp 22-26
- [49] Kontio, J. The Riskit Method for Software Risk Management, version 1.00 Institute for Advanced Computer Studies and Department of Computer Science University of Maryland CS-TR-3782 UMIACS-TR-97-38
- [50] Malaiya, Y. and Denton, J. "Requirement Volatility and Defect Density," Proc. IEEE Int.

- Symp. Software Reliability Engineering, Nov. 1999, pp. 285-294
- [51] Stark, G., Oman, P., Skillicorn, A. Ameele, R. An Examination of the Effects of Requirements Changes on Software Maintenance Releases www.stsc.hill.af.mil/crosstalk/1998/12/stark.pdf
- [52] Cockburn, A. and Highsmith, J., Agile Software Development: The Business of Innovation, IEEE Computer, pp. 120-122, Sept. 2001
- [53] Zowghi, D., Nurmuliani, N. A Study of the Impact of Requirements Volatility on Software Project Performance Ninth Asia-Pacific Software Engineering Conference (APSEC'02) December 04 - 06, 2002 Gold Coast, Australia
- [54] Jacobs, D. Requirements Engineering So Things Don't Get Ugly Focal Point Associates www.stsc.hill.af.mil/crosstalk/2004/10/0410Jacobs.pdf
- [55] Salvetto, P and Nogueira J. Size Estimation for Management Information Systems Based on Early Metrics :An Automatic Metric Tool Based in Formal Specifications. Proceedings of the International Conference on Computer Science, Software Engineering, Information Technology, e-Business and Applications (CSITeA'03), June 5-7, 2003 Rio de Janeiro, Brazil in Cooperation with the International Society for Computers and Their Applications (ISCA), USA Winona State University (WSU), USA Universidad Nacional de San Luis (UNSL), Argentina Net of National Universities with Computer Science Careers (RedUNCI), Argentina. Pags 72-77. ISBN 0-9742059-0-7.
- [56] Salvetto, P, Nogueira J., Fernández, J, Segovia J. Una Verificación Empírica de Modelos Automatizables de Estimación muy Temprana de Proyectos de Desarrollo de Sistemas de Gestión. 4 tas JORNADAS IBEROAMERICANAS DE INGENIERÍA DE SOFTWARE E INGENIERÍA DEL CONOCIMIENTO IIISIC'04. Madrid, España.
- [57] Salvetto, P., Martínez, M., Luna C., Segovia, J.. A Very Early Estimation of Software Development Time and Effort Using Neural Networks. CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN 2004 (CACIC04)
- [58] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., Selby, R. Cost Models for Future Software Life-cycle Processes: COCOMO 2.0., Annals of Software Engineering Special Volume on Software Process and Product Measurement, J.D. Arthur and S.M. Henry (Eds.), J.C. Baltzer AG, Science Publishers, Amsterdam, The Netherlands, Vol 1, 1995, pp. 45 - 60.
- [59] Londeix, B. Cost Estimation for Software Development. Addison-Wesley, 1987.
- [60] Bennett, R. Predicting software system development effort Very early in the life-cycle using Idef0 and ideo models. A Phd Dissertation Faculty of Mississippi State University, December 1996.
- [61] Boehm, B., Madachy R., Selby, R. Cost Models for Future Software Life Cycle Processes: COCOMO 2.0. Ann. Software Eng. 1: 57-94 (1995)
- [62] Kitchenham, B., Linkman, S. Estimates, Uncertainty, and Risk. IEEE Software. May-June, 1997.
- [63] Leung, H., Fan Z. Software Cost Estimation, Handbook of Software Engineering and Knowledge Engineering, Vol. II 2002 (ISBN: 981-02-4974-8)
- [64] Idri, A. , UN MODÈLE INTELLIGENT D'ESTIMATION DES COÛTS DE DÉVELOPPEMENT DE LOGICIELS , in INFORMATIQUE , Montréal , UNIVERSITÉ DU QUÉBEC À MONTRÉAL , 2003 Tesis Doctoral
- [65] Jørgensen, M, Kirkebøen, G., Sjøberg, D., Anda, B., Bratthall, L. Human judgement in effort estimation of software projects Beg, Borrow, or Steal Workshop, International Conference on Software Engineering, pp 45-51, June 5, 2000, Limerick, Ireland.
- [66] Jørgensen, M, Sjøberg, D., The importance of NOT Learning from Experience, EuroSPI'2000 - European Software Process Improvement, 7 - 9 Nov. 2000, Copenhagen, Denmark, pp. 2.2 - 2.8.
- [67] Jørgensen, M, Sjøberg, D, Kirkebøen, G. The prediction ability of experienced software maintainers 4th European Conference on Software Maintenance and Reengineering , Jurgen Ebert and Chris Verhoef (editors), IEEE Computer Society, pp. 93-100, Zurich, Switzerland, Feb 29 - March 3, 2000
- [68] Calero, Coral, Piattini, Mario, Polo, Macario, Ruiz, Francisco. Grupo ALARCOS, Departamento de Informática, Universidad de Castilla La Mancha. Métricas para la evaluación de Complejidad de Bases de Datos Relacionales. Computación y Sistemas Vol. 3, N° 4, pp 264-273, 2000, CIC - IPN. ISSN 1405-5546
- [69] Conte, S, Dunsmore, H. and Shen, V. Software Engineering Metrics and Models. Benjamin Cummings. 1986.
- [70] Briand, L. m Morasca, S. , Basili, V. An Operational Process for Goal-Driven Definition of Measures. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 28, NO. 12, DECEMBER 2002 pp 1106-1125

- [71] Juristo, N & Moreno M. Basics of Software Engineering Experimentation. Kluwer Academic Publishers 2001.
- [72] Chulani, S.,Boehm, B. Steece, B. Bayesian Analysis of Empirical Software Engineering Cost Models. IEEE Transactions on Software Engineering. July-August, 1999.
- [73] Fenton, N. and Pfleeger, S. Software Metrics. A Rigorous & Practical Approach. PWS Publishing Co. 1997.
- [74] DeMarco, T. Controlling Software Projects. Yourdon Press, New York, 1982.
- [75] Humphrey, W. A discipline for Software Engineering: The Complete PSP. Addison-Wesley, 1995.
- [76] Humphrey, W.S.; Chapter 15, Team Synergism. Managing technical people, Addison Wesley Logmann, 1997.