

Diseño e implementación de un sistema de métricas de software para mejorar los procesos de una organización de desarrollo de sistemas

Marcelo Jenkins

Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica
San Pedro, Costa Rica 2060
mjenkins@cariari.ucr.ac.cr

Roberto López

Dirección de Informática
Credomatic de CA
San José, Costa Rica
rlopez@credomatic.com

Resumen

El uso sistemático de métricas es requisito indispensable para poder administrar eficazmente un proceso de software. Este artículo describe la experiencia en diseñar e implementar un sistema de métricas de software para medir y mejorar los procesos de una organización de desarrollo de sistemas. Explicamos cómo diseñamos e implementamos nuestro sistema de métricas, se describe el uso de una herramienta para recolección y análisis de las métricas en todos los niveles de la organización, y resumimos los beneficios que hemos obtenido de su uso desde el punto de vista del mejoramiento continuo del proceso y de la evaluación del desempeño del personal.

Este artículo puede interesar las organizaciones que desean mejorar sus procesos de software mediante el uso sistemático métricas de calidad y productividad para desarrollo y mantenimiento de software.

Palabras claves: ISO 9000, aseguramiento de la calidad del software, métricas de software.

1. Introducción

Un elemento clave en cualquier proceso de desarrollo de software es la medición [1],[7],[10],[15]. Estas medidas las empleamos para valorar la calidad de los productos de software que desarrollamos. Si sabemos cómo estamos trabajando, dónde cometemos los errores, y qué acciones correctivas debemos aplicar, podemos asegurar la calidad de lo que construimos [11],[14].

Este artículo describe la experiencia en diseñar e implementar un sistema de métricas de software para medir y mejorar los procesos de la Dirección Regional de Informática de CREDOMATIC de Centroamérica. El objetivo es medir el proceso de desarrollo y determinar cuánto dura cada etapa del ciclo de desarrollo de sistemas y cada proyecto, determinar la cantidad de proyectos que se atienden, la cantidad de ítemes de software modificados, la cantidad de líneas de código cambiadas, los puntos de función desarrollados, y la cantidad de fallas o errores que ocurren. Con base en esta información, se pretende dar a la Gerencia un conjunto de estadísticas que sirvan como insumo para mejorar la calidad de los productos de software desarrollados, la calidad de los servicios prestados, y la productividad en general, así como establecer una metodología estructurada que permita evaluar el desempeño del personal y el mejoramiento continuo de la calidad del software [8][9].

2. La organización

CREDOMATIC de CA se fundó en 1974 como una empresa dedicada a proporcionar soluciones financieras a través de los medios de pagos. El grupo está posicionado en toda la región y cuenta con empresas en Guatemala, El Salvador, Nicaragua, Honduras y Costa Rica. Actualmente, CREDOMATIC y los bancos en Centroamérica tienen aproximadamente 4.500 empleados; y la corporación se divide en once direcciones:

1. ·Dirección de Banca Personal.
2. ·Dirección de Banca Corporativa.
3. ·Dirección de Comercios Afiliados.
4. ·Dirección de Cartera
5. ·Dirección de Informática.
6. ·Dirección de Operaciones.
7. ·Dirección de Calidad.
8. ·Dirección de Recursos Humanos.
9. ·Dirección de Tesorería.
10. ·Dirección de Canales de Distribución.
11. ·Dirección de Finanzas.

La Dirección de Informática está compuesta por tres gerencias de desarrollo: Tarjeta, Banca y Comercio electrónico. A la fecha contamos con uno 450 empleados en el área de informática, de los cuales 121 son técnicos y profesionales dedicados al desarrollo y mantenimiento de software. La corporación maneja actualmente más de 120 aplicaciones en producción en 4 plataformas principales: IBM AS/400 con OS/400, Windows NT y 2000, servidores HP9000 con Unix, y servidores RISC/6000 de IBM con Unix.

La Dirección de Informática está compuesta por cuatro gerencias de desarrollo:

1. Gerencia de Desarrollo Tarjetas.
2. Gerencia de Desarrollo Banca.
3. Gerencia de Desarrollo Lotus Notes
4. Gerencia de Desarrollo Internet.

Como toda institución financiera, la tecnología de información es un componente importante para mantener ventaja en un mercado muy competitivo. La disponibilidad y la calidad de todos los servicios financieros proporcionados a los clientes dependen directamente de una o más aplicaciones del software. Por lo tanto, la calidad del software tiene un efecto directo en la calidad de los servicios que el banco ofrece a sus trescientos mil clientes.

Debido a esto, en el año 2000 CREDOMATIC inició un proceso de preparación para certificarse bajo la norma ISO 9000 [3],[4], la que se obtuvo en junio de 2001. El área de informática ya había iniciado ese proceso de manera informal.

3. Descripción del Problema

A principios del año 2000 se definieron las primeras estadísticas para revisar tendencias y se hizo un esfuerzo para crear las primeras métricas. No obstante, éstas carecían de profundidad y detalle. Normalmente, eran confeccionadas para solventar la falta de información para la toma de decisiones en el nivel de la dirección como un todo. Posteriormente, se llevaron al ámbito de tres de las gerencias de desarrollo (Tarjeta, Banca y Comercio electrónico), pero aún se carece de profundidad, pues para empezar un proceso de mejoramiento continuo se requiere llevar las métricas para grupos de trabajo e incluso para uso de analistas y programadores individuales.

Este primer intento adolecía de varios problemas:

1. Las gerencias de desarrollo y los grupos de trabajo no contaban con una herramienta que les facilitara el acceso y control de su gestión de una forma detallada, que les permitiera fijar metas y objetivos cuantificables para cada elemento involucrado en el proceso de desarrollo de sistemas de información.
2. En el año 2000 solo se contaba con métricas resumidas a nivel de gerencia de desarrollo y no se podían conocer los datos a nivel de grupo de trabajo ni por programadores. Esto provocaba que los grupos de trabajo y programadores no sintieran responsabilidad por las metas de la dirección. El problema radicaba en lo grueso de la métrica.
3. En el pasado, la productividad se medía por los proyectos salientes de cada gerencia de desarrollo, pero se carecía de la cantidad de ítemes de software producidos o modificados por gerencia, grupo o programador; lo cual provocaba un sentimiento de injusticia en cuanto a las métricas en ese momento.

4. Faltaba profundidad y detalle en las métricas; por lo tanto, no permitía crear un modelo de mejoramiento continuo en ese momento. Al no tener detalle, es difícil medir con precisión. Si no se sabe cómo se está trabajando, no se puede mejorar la gestión de cada gerencia, grupo o programador.
5. Se contaba con datos valiosos de cada proyecto, de sus ítemes de software involucrados, de los tiempos reportados en cada actividad o tarea; pero no teníamos la herramienta que reuniera y procesara los datos citados para convertirlos en información valiosa a propósito de la toma de decisiones.
6. No contábamos con un indicador de productividad; por lo tanto, determinar el desempeño de las unidades, grupos y programadores se hacía difícil.

Durante el año 2000, se habían contabilizado 336 proyectos instalados en producción por la gerencia de desarrollo de tarjetas, de los cuales 51 correspondieron a errores; es decir, un 15% como indicador de calidad de error. En nuestra organización, un error se define como una falla o defecto de un producto de software instalado en producción, el cual no cumple con lo especificado en la solicitud de requerimientos (el producto es catalogado como un producto no conforme). Para establecer si un proyecto es error, se toman en cuenta aspectos tales como si afectó clientes externos o internos, si produjo pérdida monetaria, la cantidad de países afectados, y el costo de reparar el error.

Existen 4 tipos de proyectos:

1. Desarrollo: construcción de sistemas nuevos.
2. Mantenimiento: mejoras en la funcionalidad de sistemas existentes.
3. Correctivos: arreglo de defectos en sistemas en producción.
4. Error: corrección de errores cometidos durante una corrección o mantenimiento.

A finales del año 2000, se contaba con estadísticas y métricas globales en el ámbito de la dirección de informática y en las distintas gerencias de desarrollo. Por ejemplo, la Figura 1 muestra las estadísticas del número de proyectos terminados por mes durante el año 2000 clasificados por tipo de proyecto.

Cantidad de Proyectos Finalizados cada Mes por Tipo: TARJETAS						
	Desarrollo	Mantenim.	Correctivos	Error	Total	Indice Error
Enero 2000	1	10	8	3	22	14%
Febrero 2000	1	5	9	4	19	21%
Marzo 2000	6	9	7	5	27	19%
Abril 2000	2	11	6	3	22	14%
Mayo 2000	2	7	6	3	18	17%
Junio 2000	2	12	6	3	23	13%
Julio 2000	2	10	7	5	24	21%
Agosto 2000	0	5	9	4	18	22%
Septiembre 2000	2	4	9	9	24	38%
Octubre 2000	11	11	10	10	42	24%
Noviembre 2000	9	15	11	7	42	17%
Diciembre 2000	3	4	4	3	14	21%
Promedio	3	9	8	5	25	20%
Total	41	103	92	59	295	

Figura 1. Proyectos Finalizados por mes y tipo

donde el índice de error se calcula según la siguiente fórmula:

$$\text{Índice de error} = \frac{\text{Proyectos con error por mes}}{\text{Total proyectos finalizados por mes}} \times 100\%$$

La pregunta primordial es: ¿Qué se ha hecho para que no se vuelvan a repetir esos errores? Es evidente que se carece de una metodología de calidad que se retroalimente de esta información y se tomen las acciones correctivas para mejorar continuamente [8],[9]. ¿Cómo mejorar los tiempos de entrega, la eficiencia de la planificación y los tiempos promedios en general si se cuenta con la herramienta y el suficiente detalle para sentar responsabilidad a nivel de gerencias, grupos y programadores? Al no contar con métricas y estadísticas con suficiente profundidad y detalle, no se obtiene retroalimentación y consecuentemente no se pueden tomar acciones correctivas de mejoramiento continuo de la calidad.

4. Diseño del Sistema de Métricas

Hace tres años, la Gerencia de Desarrollo de Sistemas decidió implementar un proyecto a mediano para diseñar e implementar un conjunto de métricas de software detalladas que serán utilizadas por la Dirección Regional de Informática, Gerencias de Desarrollo, grupos de trabajo y analistas programadores; que permitan medir la calidad y productividad de la gestión de desarrollo y que señale cuáles son las fortalezas y debilidades de la gestión, con miras a mejorar la productividad y calidad de los productos de software desarrollados a través del mejoramiento continuo y la evaluación del desempeño de los integrantes del proceso de desarrollo. Esta iniciativa se enmarcó siempre dentro del sistema de administración de la calidad de software (SQMS) basado en la norma ISO 9000:2000 [3], buscando alcanzar cinco objetivos específicos:

1. Crear una aplicación que recolecte y resuma las métricas de productividad y calidad.
2. Implementar el uso de la nueva aplicación y de las métricas en toda la Dirección Regional de Informática
3. Implementar métricas más detalladas y con mayor profundidad (no solo a nivel de proyectos).
4. Establecer objetivos cuantitativos que permitan evaluar la gestión.
5. Establecer ciclos de mejoramiento continuo que permitan el seguimiento y la retroalimentación en toda la Dirección.

4.1 Fuentes de datos para la recolección de métricas

La Dirección de Informática cuenta con herramientas automatizadas que brindan seguimiento y control al proceso de desarrollo de sistemas. Estas registran las fases del ciclo de desarrollo y constituyen la fuente de datos e información requerida para construir las estadísticas y métricas planteadas. Las herramientas son las siguientes:

1. Sistema de análisis de requerimientos de Credomatic (SARCOM). Esta herramienta recopila los requerimientos planteados por los usuarios y, en términos generales, es un flujo de trabajo automatizado que va desde la inclusión del requerimiento hasta la puesta en producción del producto de software requerido. El instrumento fue desarrollado en la aplicación Lotus Notes y contempla todo un ciclo de revisión y aprobación por las distintas áreas funcionales de la organización. A cada solicitud de programación se le asigna un número único y universal, el cual sirve como referencia para las demás herramientas donde se registran datos relacionados con esta. Una solicitud de programación es sinónimo de proyecto; por lo tanto, para este documento se tratan de igual forma.
2. Herramienta para la confección de Planes de Trabajo (PLT). Esta herramienta fue desarrollada en Lotus Notes y es un flujo de trabajo automatizado o Workflow, donde los analistas programadores y los líderes de proyectos indican cómo se realizará el trabajo. Para cada solicitud de programación se confecciona un Plan de Trabajo y se usa el mismo número de referencia el cual los vincula.
3. Herramienta para la administración de la configuración (PDE). Esta herramienta administra la configuración de los ítemes de software, la cual posibilita manejar las versiones del software. Esta herramienta agrupa los ítemes de software necesarios para desarrollar el proyecto. Está conformado por fuentes y objetos asociados a las diferentes aplicaciones del sistema que están sujetas a modificar o a crear. Para cada proyecto incluido en SARCOM, se asocian los ítemes de software requeridos. La herramienta fue comprada a un proveedor y está desarrollada en el AS/400.
4. Herramienta para registro de tiempos a los proyectos (Team Project). Esta herramienta registra los tiempos consumidos en proceso de desarrollo. Para cada proyecto se confecciona un cronograma donde se detallan las diferentes actividades y tareas para llevarlo a cabo. Tanto el líder de proyecto y los analistas programadores que desarrollan el proyecto deben registrar el tiempo ocupado en cada una de las etapas de desarrollo. La herramienta es un workflow desarrollado en Lotus Notes

4.2 Proceso de recolección de datos

Las solicitudes de programación se registran en la herramienta SARCOM, la cual está desarrollada en Lotus Notes; los planes de trabajo se registran en la herramienta PLT Planes de Trabajo; estos abarcan el análisis y diseño de un proyecto o solicitud de programación y fueron desarrollados en la herramienta Lotus Notes; los ítemes de software están almacenados y administrados en la herramienta PDE y radica en el AS/400; los tiempos reportados para cada proyecto se registran en la herramienta Team Project, la cual está desarrollada en Lotus Notes. La solución planteada une los datos de las cuatro fuentes y convergen en una base de datos relacional en Access. En la Figura 2

se muestra la convergencia de las herramientas desde el punto de vista de hardware y la Figura 3 muestra la convergencia de los datos en una sola base de datos consolidada.

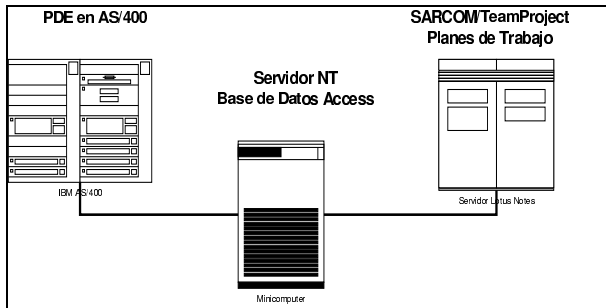


Figura 2. Convergencia de las herramientas

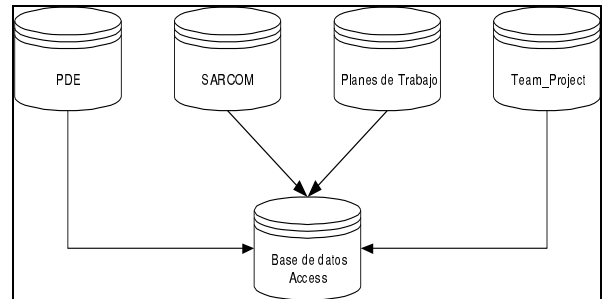


Figura 3. Convergencia de los datos

4.3 Profundidad de las métricas

Se desarrolló una estructura de métricas que abarque todos los niveles de la dirección de informática. La idea es que cada nivel pueda consultar las estadísticas y métricas para su mejoramiento continuo y que sirvan de apoyo a toda la organización del área de informática regional. La Figura 4 muestra la profundidad de las métricas propuestas.

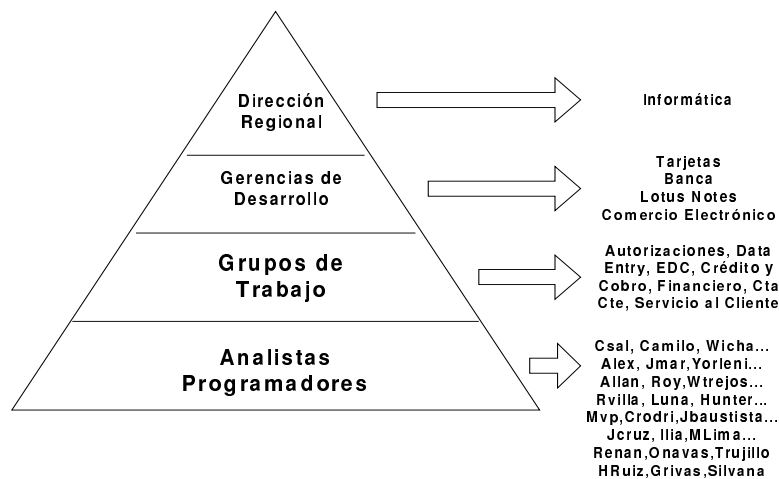


Figura 4: Profundidad de las métricas

Actualmente, cada unidad organizativa o gerencia de desarrollo tiene grupos de trabajo que atienden las diferentes aplicaciones del sistema actual. Cada grupo está conformado por un líder y un conjunto de analistas programadores, quienes tienen a su cargo suplir las necesidades de sistemas de información de las aplicaciones.

4.4 Conformación de las métricas

Las métricas están conformadas por cuatro herramientas que convergen en una base de datos relacional. Las solicitudes de programación que se capturan con la herramienta SARCOM contienen los proyectos que solicitan todas las fuerzas de la organización; en ella se detallan y se analizan los requerimientos de los sistemas de información. De las solicitudes de programación se derivan los Planes de Trabajo, en los cuales se detalla el análisis y el diseño de los sistemas que se confeccionan. Estos son construidos por los líderes de proyectos y los analistas programadores; además, se capturan en la herramienta de Planes de Trabajo o PLT. De los Planes de Trabajo surgen dos componentes importantes para las métricas: a) los ítemes de *software* por retener y b) los cronogramas de análisis, diseño y ejecución que sirven para registrar tiempos a los proyectos.

Las solicitudes de programación y Planes de Trabajo son la base para las métricas orientadas a los proyectos; los ítemes de *software* retenidos y pasados a producción son la base para las métricas orientadas a ítemes de *software* y los tiempos reportados en Team Project son la base para las métricas orientadas al tiempo o duración promedio. La Figura 5 muestra la conformación de las métricas.

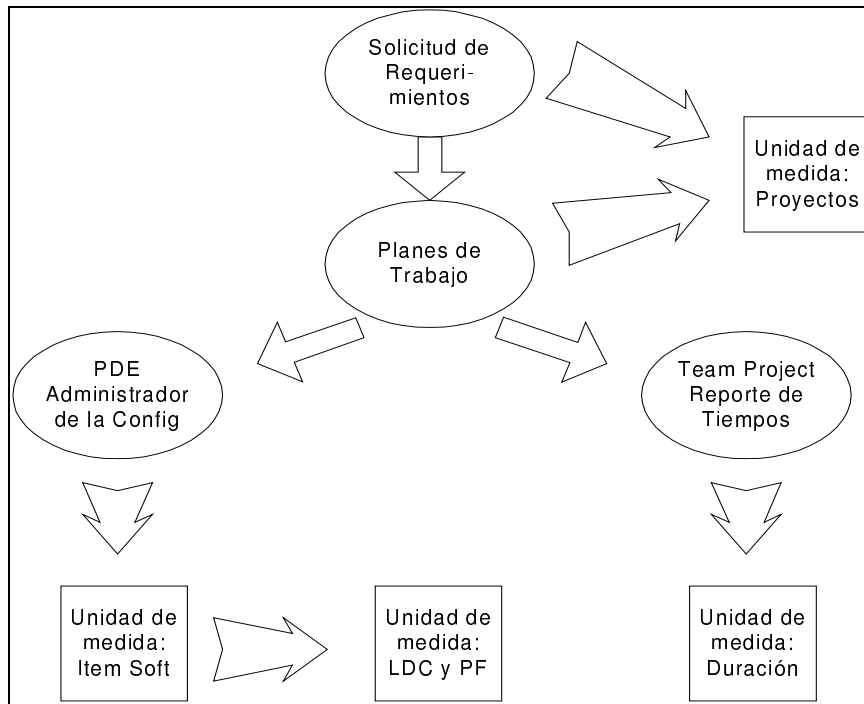


Figura 5: Conformación de las métricas

Las fuentes que conforman se relacionan con la profundidad de las métricas; es decir, la información recolectada sirve como base para construir las métricas en los diferentes niveles de la organización de la dirección de informática. La Figura 6 muestra la relación de ambos conceptos:



Figura 6. Relación en las fuentes de métricas y niveles de la organización.

4.5 Unidades de medidas de las métricas

Las métricas y estadísticas desarrolladas usan cuatro unidades de medidas. La idea de este proyecto es ir de la medida más global a la medida más específica. Uno de los problemas que se tienen con las métricas previas es que, en algunos casos, la unidad de medida es tan amplia que se vuelve “injusta” o poco representativa; sin embargo, en vista de que ya algunas estaban implementadas, no se vio justificación para eliminarlas. Las unidades de medida serán usadas en todos los niveles de la organización y son obtenidas de las distintas fuentes que conforman las métricas.

En la Figura 7 se muestra la relación entre las fuentes de conformación de las métricas y las unidades de medidas que se utilizan.

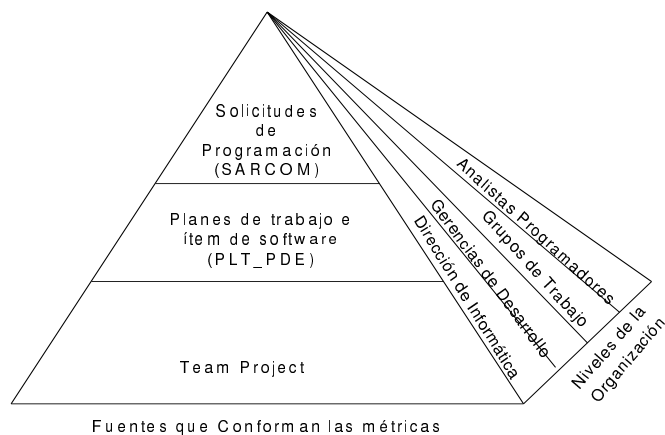


Figura 7. Relación de fuentes que conforman las métricas y las unidades de medidas

Por otro lado, la Figura 8 se muestra la relación entre los niveles de la organización y las unidades de medida que se utilizan.

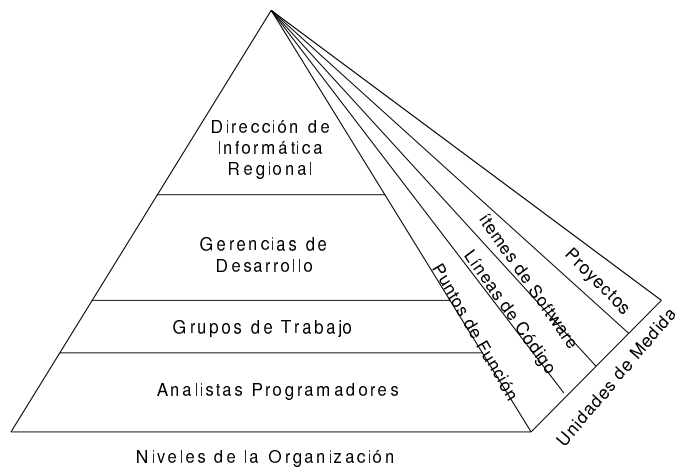


Figura 8. Relación entre los niveles de la organización y las unidades de medida

4.6 Descripción de las Estadísticas y Métricas

A continuación La Tabla 1 detalla el nombre, descripción, forma de cálculo, e interpretación de cada métrica que se propone.

Tabla 1. Descripción detallada de las métricas de calidad y productividad.

Nombre	Tipo	Definición	Cálculo	Interpretación
Cantidad de proyectos entrantes (totales, por tipo, y por prioridad)	Productividad	Indica la cantidad de proyectos que entraron y que fueron aceptados por el Comité de Desarrollo, en un mes para toda la dirección y sus dependencias.	Σ Proyectos Entrantes en el mes	Por sí sola, la métrica nos indica poco, pero combinada con los proyectos finalizados nos señala la razón de E/S o la diferencia.
Cantidad de proyectos finalizados (totales, por tipo, y por prioridad)	Productividad	Indica la cantidad de proyectos finalizados y que fueron instalados en producción, en un mes para toda la dirección y sus dependencias.	Σ Proyectos Finalizados en el mes	Por sí sola, la métrica nos indica poco, pero combinada con los proyectos entrantes nos señala la razón de E/S o la diferencia.
Diferencia entre los proyectos entrantes y salientes (totales, por tipo, y por prioridad)	Productividad	Indica la cantidad de proyectos entrantes menos la cantidad de proyectos finalizados, en un mes para toda la dirección y sus dependencias.	Σ Proyectos Entrantes en el mes MENOS Σ Proyectos Finalizados en el mes	Durante el 2001 la diferencia fue de 203 proyectos negativos, implica que la oferta en programación no alcanzó para cubrir la demanda de ese año.
Porcentaje de proyectos Finalizados por tipo o por prioridad	Productividad	Indica el porcentaje de proyectos que se finalizaron de Desarrollo, Mantenimiento, y Corrección para toda la dirección y sus dependencias en un mes.	Σ Proyectos Finalizados de Desarrollo, Mantenimiento y de Corrección en el mes DIVIDIDO Total de Proyectos Finalizados en el mes. Se calcula un porcentaje para cada tipo.	Si el porcentaje de corrección supera el 20%, se debe hacer un análisis detallado de las causas por las cuales se tuvieron que hacer las correcciones.
Índice de error	Calidad	Indica el porcentaje de proyectos finalizados de Error generados en relación con el total de proyectos finalizados para toda la dirección y sus dependencias en un mes.	\square Proyectos Finalizados de error en el mes DIVIDIDO Total de Proyectos Finalizados en el mes	Si el porcentaje de error supera el 10%, se debe hacer un análisis detallado de las causas por las cuales se cometieron los errores.
Duración promedio de proyectos finalizados por tipo o por prioridad	Productividad	Duración promedio en días de los proyectos finalizados de Desarrollo, Mantenimiento, Corrección y de error para toda la dirección y sus dependencias en un mes.	Σ Duración en días de proyectos finalizados por tipo en el mes DIVIDIDO entre el Total de Proyectos finalizados por tipo en el mes	En general, se pretende que todos los proyectos duren lo menos, pero hay muchas variables externas que inciden en esta métrica.
Duración promedio por etapa de proyectos finalizados	Productividad	Duración promedio en días de los proyectos finalizados en la etapa de análisis de requerimientos, en etapa de espera o cola y en la etapa de desarrollo para toda la dirección y sus dependencias en un mes.	Σ Duración en días de las etapas para proyectos finalizados en el mes DIVIDIDO entre el Total de Proyectos finalizados en el mes	En general, se pretende que todos los proyectos duren lo menos en cada etapa, pero hay muchas variables externas que inciden en esta métrica. La etapa de análisis de requerimientos se ve impactada por el tiempo de espera para aclarar los requerimientos con los usuarios; el tiempo de espera o en cola se ve influida por el cambio de prioridades y el desarrollo por la productividad de los desarrolladores o por mala planificación.
Índice de planeación (total, por tipo o por prioridad)	Calidad	Indica el porcentaje de eficiencia en la planificación en los proyectos finalizados en general para toda la dirección y sus dependencias en un mes.	Σ duración promedio estimada en horas en el mes DIVIDIDO entre Σ duración promedio real en horas en el mes.	En general, se pretende que todos los proyectos se finalicen en el tiempo estimado.. De todos aquellos proyectos que estén por encima o por debajo del 20%, se revisan las causas de las desviaciones y se formulan acciones que corrigen esas diferencias.

Cantidad de proyectos en espera o cola por tipo o por prioridad	Productividad	Indica la cantidad de proyectos en espera o cola de Desarrollo, Mantenimiento, Corrección y de error, que fueron aceptados por el Comité de Desarrollo para toda la dirección y sus dependencias en un mes.	Σ Proyectos en espera o cola de desarrollo, mantenimiento, corrección y error en el mes (se calcula un total por cada uno de los tipos)	Si los proyectos de corrección y error superan a los de desarrollo y de mantenimiento, nos da un indicio de incremento en el retrabajo.
Duración promedio para evacuar proyectos en cola por tipo o por prioridad	Productividad	Tiempo promedio para evacuar la cola de proyecto de desarrollo, mantenimiento, corrección y error basados en el promedio de programación para toda la dirección y sus dependencias en un mes.	Cantidad de proyectos por tipo en cola en el mes MULTIPLICADO por el tiempo promedio utilizado en la etapa de programación para proyectos desarrollo, mantenimiento, corrección y error finalizados en el mes.	Esta métrica nos indica cuánto tiempo requerimos para evacuar la cola, basados en los promedio de los proyectos de su mismo tipo. Si se agregan los recursos disponibles, sabremos cuándo se termina la cola o podemos derivar cuántos recursos faltan.
Cantidad de ítemes de software pasados a producción.	Productividad	Cantidad de ítemes de software pasados a producción de los proyectos finalizados para toda la dirección y sus dependencias en un mes.	Σ ítemes de software pasados a producción en el mes	Por sí sola, la métrica nos indica poco, pero combinada con los ítemes pasados a producción con error nos da una tasa de error por ítemes.
Cantidad de ítemes de software con error	Calidad	Cantidad de ítemes de software con error de los proyectos finalizados para toda la dirección y sus dependencias en un mes.	Σ ítemes de software con error pasados a producción en el mes.	Por sí sola, la métrica nos indica poco, pero combinada con los ítemes pasados a producción nos da una tasa de error por ítemes.
Índice de error por ítemes de software.	Calidad	Indica el porcentaje de ítemes de software con Error en relación al total de ítemes pasados a producción para toda la dirección y sus dependencias en un mes.	Σ ítemes de software con error en el mes ENTRE Σ ítemes de software pasados a producción en el mes.	Si el porcentaje de error supera el 5%, se debe hacer un análisis detallado de las causas por las cuales se cometieron los errores.
Cantidad de ítemes de software pasados a producción por tipo o por prioridad.	Productividad	Indica la cantidad de ítemes de software de desarrollo, mantenimiento, corrección y error, pasados a producción de los proyectos finalizados para toda la dirección y sus dependencias en un mes.	Σ ítemes de software pasados a producción por tipo en el mes Se da un total por cada uno de los tipos.	Por sí sola, la métrica nos indica poco, pero combinada con los ítemes pasados a producción con error nos da una tasa de error por ítemes.
Cantidad de Puntos de Función Marginal por tipo o por prioridad	Productividad	Indica la cantidad de Puntos de Función Marginal de desarrollo, mantenimiento, corrección y error, correspondientes a proyectos finalizados para toda la dirección y sus dependencias en un mes.	Σ Puntos de Función Marginal desarrollados por tipo o por prioridad en el mes Se da un total por cada uno de los tipos	Si se combinada con los Puntos de Función con error, nos da una tasa de error por Puntos de Función. También no describe cuál es la productividad real de la dirección, gerencia, grupo o analista programador.
Índice de error por Puntos de Función Marginal	Calidad	Indica el porcentaje de puntos de función con Error en relación con el total de puntos de función pasados a producción para toda la dirección y sus dependencias en un mes.	Σ Puntos de Función Marginal con error en el mes ENTRE Σ Total de Puntos de Función Marginal pasados a producción en el mes	Si el porcentaje de error supera el 5%, se debe hacer un análisis detallado de las causas por las cuales se cometieron los errores.
Productividad Incremental de Puntos de Función	Productividad	Indica un factor de productividad basado en los puntos de función generados y el esfuerzo para llevarlos a cabo para toda la dirección y sus dependencias en un mes.	Σ Puntos de función pasados a producción en el mes DIVIDIDO entre la duración en horas de desarrollo en el mes.	Cuanto mayor sea el índice más productivos somos, o sea, se producen mayor cantidad de puntos de función en relación con el esfuerzo gastado.

4.7 Proceso de recolección y análisis de Métricas

En este punto se tiene los tres elementos esenciales para procesar las métricas para uso y consulta de toda la Dirección Regional de Informática. Las tres fuentes de información convergen en la base de datos Access, donde se relacionan y se procesan. A continuación se describen los elementos:

1. Base de datos de SARCOM en Access: aquí tenemos todos los proyectos o solicitudes de programación para un período determinado.
2. Base de datos de PDE en Access: en esta se almacenan todos los ítemes de *software* y puntos de función relacionados con cada ítem, que corresponden a los proyectos o solicitudes de programación instalados en producción.
3. Base de datos de Team Project en Access: se almacenan los tiempos reportados a los proyectos y las tareas de estos, que corresponden a los proyectos o solicitudes de programación puestos en producción.

El procesamiento de las métricas como un todo consta de siete pasos que se muestran en la Figura 7.

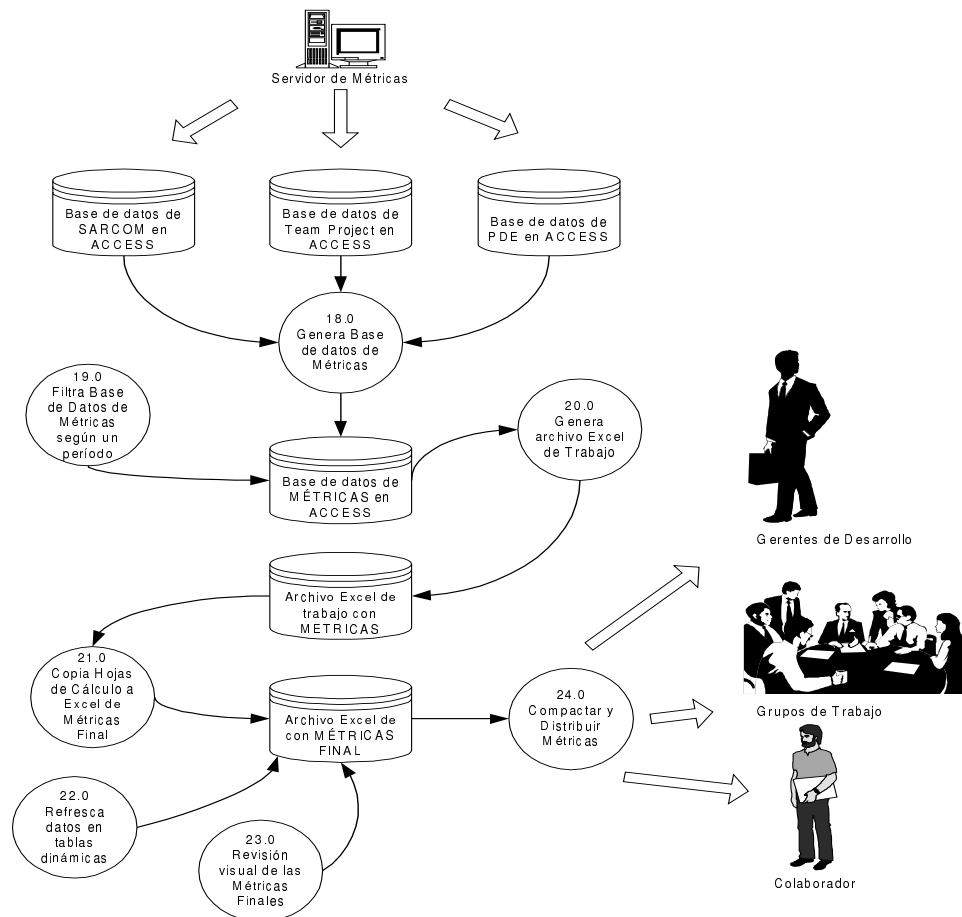


Figura 9. Generación de las métricas para uso y consulta.

4.8 Implementación de las Métricas

En esta sección se describen los resultados que se han obtenido de la experiencia de implementar las métricas arriba descritas.

4.8.1 Proyectos finalizados e índice de error

Esta es una de las métricas de mayor uso por parte de la Dirección de Informática Regional, debido a que forma parte de indicadores de gestión de calidad. La Figura 10 muestra la cantidad de ítemes pasados a producción y la Figura 11 muestra los datos estadísticos donde se refleja la cantidad total de proyectos finalizados cada mes, el desglose de los proyectos finalizados por tipo (Desarrollo, Mantenimiento, Corrección y Errores) y el índice de error por proyecto. En el año 2001 se finalizaron 440 proyectos con un promedio de 36,7 proyectos por mes. Se distribuyen en 70 proyectos de desarrollo con un promedio 5,83 proyectos por mes, 182 de mantenimiento con un promedio de 15,17 proyectos al mes, 146 de corrección, con un promedio de 12.17 proyectos al mes, y 42 de error con un promedio al mes de 3,5 proyectos. Así, para el año 2001 se fijó como meta no sobrepasar este índice en un 10%. Revisando los datos estadísticos en la Figura 11, el índice de error promedio en el año 2001 fue de un 10%; es decir, se cumplió con la meta..

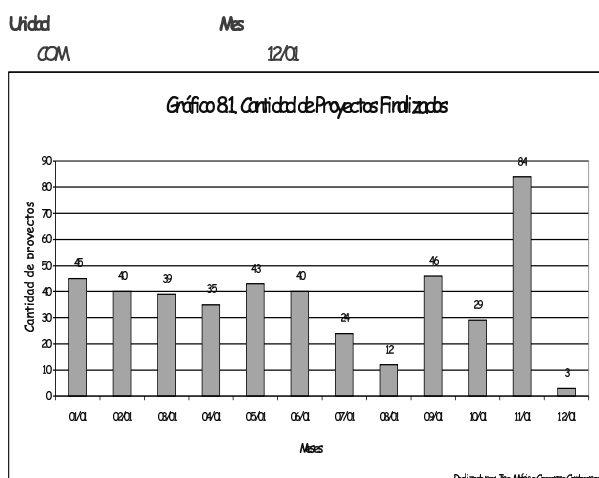


Figura 10. Cantidad de proyectos finalizados por mes.

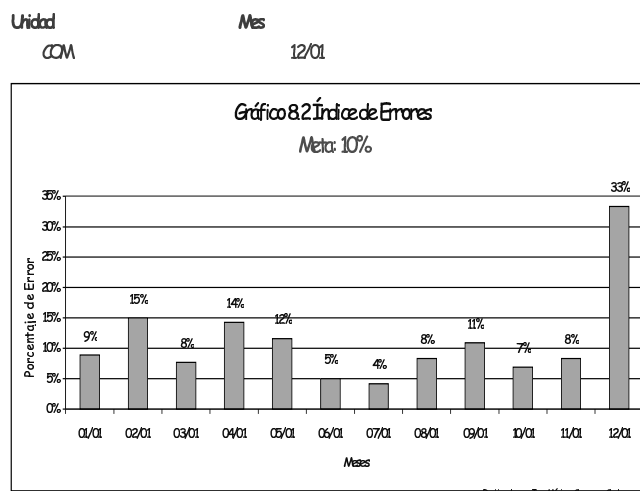


Figura 11. Índice de error de proyectos por mes.

4.8.2 Duración promedio real/estimada e índice de planeación

Esta métrica ha tomado mucha importancia en la Dirección de Informática Regional y está orientada a medir cuan bien se estima la duración de los proyectos y no el cumplimiento en fechas de entrega. La duración estimada se toma de los cronogramas que se adjuntan en los planes de trabajo y la duración real de Team Project, específicamente de los tiempos reportados a las tareas de los proyecto. La Figura 12 muestra el Tiempo promedio de duración por proyectos y la Figura 13 el Índice de planificación de proyectos.

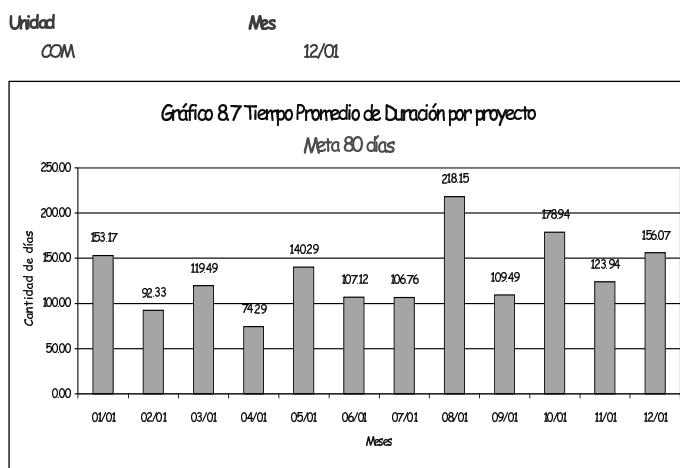


Figura 12. Tiempo promedio de duración de proyectos.

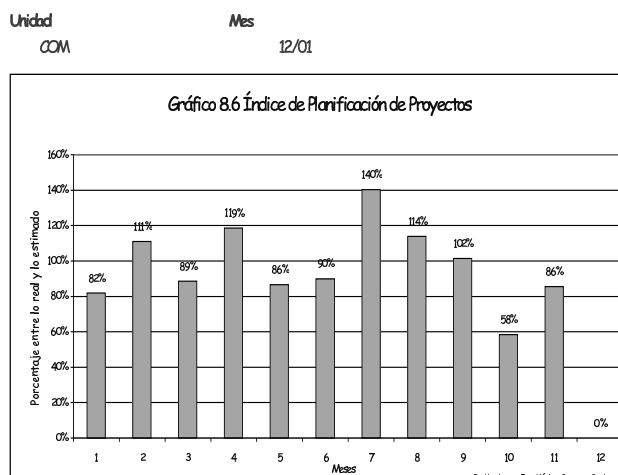


Figura 13. Índice de planificación de proyectos.

4.8.3 Otras métricas

Otras métrica importantes son los tamaños de las colas o presas de trabajo. La Figura 14, muestra la cantidad de proyectos en cola por unidad y la Figura 15 el tiempo promedio en días para evacuar la cola

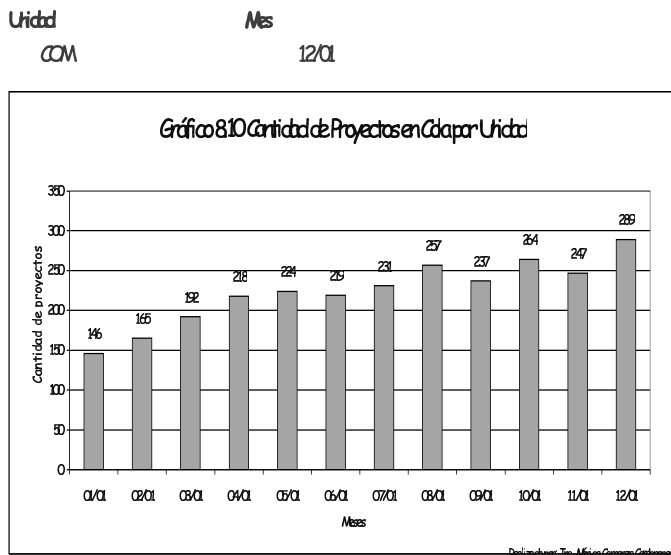


Figura 14. Cantidad de proyectos en cola por unidad por mes

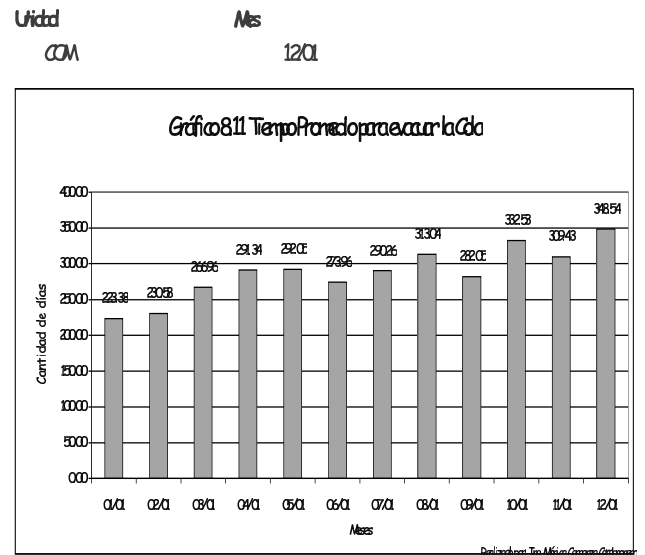


Figura 15. Tiempo Promedio en días para evacuar la cola

7. Conclusiones

La norma ISO 9000:2000 requiere implementar un proceso de medición del proceso para desarrollar y mantener software. La herramienta propuesta permite a la Dirección de Informática Regional recolectar y procesar la información existente. Se estableció cómo se calculan las métricas actuales y se implementó en toda la organización. Se determinaron las fuentes de datos con que se contaban, se consolidó toda la información en una base de datos común, a partir de la cual se calculan las métricas. Por último, las métricas se presentaron en una herramienta de fácil uso y acceso, con lo que se solventó la carencia de información para la toma de decisiones de la toda la Dirección de Informática Regional y se proporciona un instrumento de medición del proceso de desarrollo y mantenimiento de software.

Las métricas no son útiles únicamente para la Dirección de Informática, sino para todos los niveles de la organización. La herramienta brinda información como un todo a la Dirección de Informática Regional, así como particularmente a cada gerencia, grupo de trabajo e individualmente a cada analista programador. De esta manera, se pueden establecer metas u objetivos medibles en todos los niveles de la organización, con el fin de que valoren su rendimiento.

No todas las métricas sirven a todas las gerencias. En Credomatic existen cuatro gerencias de desarrollo; dos de ellas trabajan exclusivamente en AS/400, y las otras dos en sistemas abiertos. El alcance de las métricas abarca a las que trabajan en la primera y parcialmente a las gerencias que trabajan en sistemas abiertos. En una segunda fase que ya inició, se está elaborando el diseño de las métricas propuestas para los sistemas abiertos, de tal forma que los instrumentos de medición sean iguales para todas las plataformas computacionales.

Finalmente, nada se logra con medir y cuantificar el trabajo de la Dirección de Informática si no se cuenta con un proceso de mejora continua y con un sistema de evaluación del desempeño. Estos dos temas ya se están tratando en la organización pero por motivos de espacio quedan fuera de esta explicación.

Referencias

- [1] Grady, R. *Practical Software Metrics For Project Management And Process Improvement*. 1ed, Prentice-Hall, 1992.
- [2] IEEE. *IEEE Standards Collection: Software Engineering*, 1999 edition. IEEE Inc. 1999.
- [3] ISO. *International Standard ISO 9001*. ISO 2000.
- [4] ISO. *Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for their use*. ISO 1991.
- [5] M. Jenkins. Adopting Development Standards to Achieve Process Improvement. *Proceedings Sixth International Conference on Software Quality*, Montreal, Canada, 1996, pages. 111-120.
- [6] G.A. Kaplan. Secrets of Software Quality. *Proceedings Fifth International Conference on Software Quality*, Austin, Texas, 1995, pages. 15-27.
- [7] S.H. Kan. *Metrics and Models in Software Quality Engineering*, Addison-Wesley, 1995.
- [8] Laudon, K y Loudon, J. *Administración de los Sistemas de Información: Organización y Tecnología* 3ed. U.S.A.: Prentice Hall, 1996.
- [9] E. McGuire. *Software Process Improvement: Concepts and Practices*. Idea Group Publishing, 1999.
- [10] J.W. Moore. *Software Engineering Standards: A User's Road Map*. IEEE Inc., 2000.
- [11] M. Paulk, B. Curtis, M.B. Chrissis, C.V. Weber. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley, 1995.
- [12] S. Lawrence Pfleeger, N. Fenton, S. Page. Evaluating Software Engineering Standards. *IEEE Computer*. Sept. 1994, pages. 71-79.
- [13] N.F. Schneidewind, N. Fenton. Do Standards Improve Quality? *IEEE Software*. Jan. 1996, pages. 22-24.
- [14] Schulmeyer G.G., McManus J.I. *Handbook of Software Quality Assurance*. Prentice Hall, 1999.
- [15] Vachette, J. *Mejora Continua de la Calidad – Control estadístico del Proceso (SPC)*. 1ed España: Ediciones CEAC S.A., 1992.