

**Escuela de Negocios**

**Tipo de documento:** Tesis de maestría



*EMBA | Executive MBA*

## **Analizar la productividad en el equipo de desarrollo de sistemas de HSBC Argentina**

**Autoría:** Codari, Hernán Pablo

**Año:** 2012

### **¿Cómo citar este trabajo?**

Codari, H. (2012). "Analizar la productividad en el equipo de desarrollo de sistemas de HSBC Argentina". [Tesis de maestría. Universidad Torcuato Di Tella]. Repositorio Digital Universidad Torcuato Di Tella.

<https://repositorio.utdt.edu/handle/20.500.13098/14058>

El presente documento se encuentra alojado en el **Repositorio Digital de la Universidad Torcuato Di Tella** bajo una licencia Creative Commons Atribución-No Comercial-Compartir Igual 4.0 Internacional  
**Dirección:** <https://repositorio.utdt.edu>

**UTDT - ESCUELA DE NEGOCIOS**

**TESIS MBA – HSBC 2010**

**ANALIZAR LA PRODUCTIVIDAD EN EL EQUIPO DE  
DESARROLLO DE SISTEMAS DE HSBC ARGENTINA**

**AUTOR: HERNAN PABLO CODARI**

**TUTOR: GABRIEL PEREYRA**

**JUNIO 2012**

**LUGAR: CIUDAD AUTÓNOMA DE BUENOS AIRES**

## **RESUMEN**

El objetivo del presente trabajo fue analizar los equipos de desarrollo de software de HSBC Argentina, especialmente en lo que respecta a la medición de su productividad.

Asimismo se detallaron los principales roles dentro del área de desarrollo, las motivaciones y factores de éxito de los equipos de trabajo y la forma de remunerarlos.

En relación a las posibles mediciones de la productividad, se incluye una descripción de las principales métricas existentes en la actualidad que tienen como objeto la medición de dichos desarrollos.

Luego se presentarán datos reales de la institución a fin de comprobar cuál es el grado de penetración de dichas métricas y la aceptación de las mismas por parte de los empleados.

La conclusión es que las herramientas de medición de productividad actuales utilizadas en HSBC Argentina no son suficientes por sí solas como una medida de la productividad sino que deben ser complementadas por otros indicadores que reflejen la performance de los equipos en todas las etapas del desarrollo de software.

## **PALABRAS CLAVE**

productividad, sistemas, equipos

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	3
CAPÍTULO 1 – EL PERFIL DE LOS EQUIPOS DE DESARROLLO DE SOFTWARE ..	7
CAPÍTULO 2 – PRODUCTIVIDAD Y EFICIENCIA .....	12
CAPÍTULO 3 – LAS DISTINTAS MÉTRICAS DE PRODUCTIVIDAD UTILIZADAS EN EL DESARROLLO DEL SOFTWARE .....	18
CAPÍTULO 4 – TRABAJO DE INVESTIGACIÓN: ¿CÓMO SON MEDIDOS LOS DESARROLLADORES?.....	25
CAPÍTULO 5 – EL GERENCIAMIENTO DE EQUIPOS DE DESARROLLO Y LA MEDICIÓN DE LOS MISMOS EN HSBC ARGENTINA .....	33
CONCLUSIONES .....	49
BIBLIOGRAFÍA .....	51
GLOSARIO .....	54
ANEXO 1 - Encuesta a Desarrolladores de Software.....	57

## INTRODUCCIÓN

Existen distintos métodos y métricas a fin de lograr la medición de la productividad en equipos de desarrollo de software.

Estos métodos han sido propuestos y analizados por diversos autores y aplicados en distintos proyectos en varias industrias.

Sin embargo, estas métricas no reflejan adecuadamente la productividad y eficiencia de un equipo de desarrolladores de software así como tampoco su performance individual. Esto dificulta la gestión del gerenciamiento de dichos equipos y complica la justificación de los recursos tanto económicos como humanos asignados al equipo en cuestión.

En el área del gerenciamiento de proyectos de software, se aplican métodos muy complejos que son utilizados para las mediciones de dichos proyectos en lo que respecta a las variables de presupuesto y tiempo. Estos métodos abarcan todo el ciclo de vida de los mismos, partiendo de las definiciones iniciales hasta su implementación y control posterior y se mantienen hasta asegurar su total operatividad. De acuerdo a Benediktsson (2006)<sup>1</sup>, “el ciclo de vida representa un camino entre el origen y la finalización de una tarea. La división en fases permite a los gerentes controlar y administrar las actividades en una forma ordenada y metódica”.

El área de aplicación habitual de estos métodos comprende mayormente a ciertas empresas de consultoría (que son habitualmente contratadas para la planificación e implementación de un proyecto específico), las cuales dedican grupos de trabajo a tiempo completo al desarrollo e implementación de un proyecto en particular. Asimismo las fábricas de software (*software factories*) también aplican métodos de estas características.

---

<sup>1</sup> Benediktsson, O., Dalcher, D. & Thorbergsson, H. (2006). Comparison of software development life cycles: a multiproject experiment, IEE Proceedings, 153 3, pp. 88.

Existe la dificultad, sin embargo, de que esta forma de gerenciamiento basada en métodos académicos no puede ser fácilmente replicada en aquellas áreas informáticas que deben utilizar recursos propios de desarrollo de software en una doble función: tanto para la implementación de nuevos proyectos, como para el mantenimiento de incidentes productivos y arreglos menores sobre sistemas ya existentes en sus instalaciones. La coordinación en estos equipos de desarrollo es particularmente compleja según Poile<sup>2</sup>. Adicionalmente, Fenton (1998)<sup>3</sup> agrega que los métodos académicos ya no son aceptados como válidos por su mera publicación sino que el éxito de los mismos se debe basar en resultados empíricos y con la habilidad de repetir dichos resultados.

Particularmente, HSBC Argentina posee una distribución de recursos similar a la mencionada, y por lo tanto no son apropiados únicamente los métodos tradicionales de medición de software que se desarrollarán más adelante. Estos métodos son los más relevantes según algunos autores como Fenton (1998), Mills (1988) y Mc Connell (2008) y pueden verse afectados en sus variables por factores exógenos, como ser las problemáticas de la actividad cotidiana, la emisión de nuevas regulaciones, la rotación de personal, el cambio de prioridades en los negocios (que a su vez implican alterar la planificación de las áreas de desarrollo), etc.

La situación mencionada precedentemente obliga a cuestionar la utilidad de los métodos conocidos así como también la efectividad de los mismos en un entorno cambiante como el de HSBC Argentina. Por último, se plantea el problema de comprender la interacción de los mismos y determinar las mejores opciones disponibles.

Se plantea la hipótesis de que las distintas formas de medir la eficiencia en los desarrollos de software aplicables en HSBC Argentina, no son consideradas desde una visión abarcadora y generalizada como para que resulten efectivas en el gerenciamiento de

---

<sup>2</sup> Poile, C., Nachiappan Nagappan, A. & Layman, L. (2009). Coordination in Large-Scale Software Development: Helpful and Unhelpful Behaviors, *Microsoft Research*, 135, pp 12.

<sup>3</sup> Fenton, N. & Neil, M. (1998). Software Metrics: Successes, Failures and New Directions, *Journal of Systems and Software*, July/August 2000, pp 4.

equipos de trabajo y por lo tanto no deben ser consideradas como medición de productividad de los mismos.

Se trata entonces de comprender los criterios utilizados por los distintos métodos de medición, pero principalmente de observar su aplicación práctica en el área de desarrollo de software de HSBC Argentina S.A. integrada por 170 empleados y funcionarios, apuntando a la interacción entre el nivel gerencial, como encargado de la medición, y los propios desarrolladores.

El método de trabajo consistirá en una exposición inicial de los principales conceptos y prácticas que son de uso habitual en esta actividad según los autores consultados. Luego se procederá a la realización de una encuesta a un grupo testigo de desarrolladores de HSBC Argentina a fin de comprender su visión en lo que respecta a la consideración que les merecen las distintas técnicas de medición en su forma de ser evaluados y por último se plantean entrevistas presenciales a los niveles gerenciales de HSBC Argentina y de una empresa de consultoría independiente (INDRA).

El resultado mostrará la necesidad de obtener una metodología confiable de medición en HSBC, analizando la factibilidad de aplicación de las métricas en HSBC Argentina, la cual se complementa con la visión de los desarrolladores, y la de los niveles gerenciales.

El trabajo se organiza mediante capítulos. En el Capítulo 1 se brindará un marco antropológico de referencia, describiendo y modelizando el perfil de las personas que trabajan en la Gerencia de Desarrollo de software de HSBC, enfocándose en los principales roles presentes en dicha gerencia. Se describen brevemente las técnicas de liderazgo y motivación utilizadas así como los factores más relevantes que hacen al éxito de un grupo de trabajo.

En el Capítulo 2 se introducen los conceptos de productividad y eficiencia según distintos autores y se exponen algunas dificultades que aparecen al tratar de medir la productividad de tareas fundamentalmente intelectuales, no completamente mecanizadas o enmarcadas en

un proceso único. También se menciona el concepto de la remuneración variable en relación a la productividad generada por cada equipo y cada individuo.

En el Capítulo 3 se relevan y revisan las principales métricas utilizadas en la actualidad y que son consideradas el ‘estado del arte’.

Este análisis proveerá la base necesaria para desarrollar la investigación acerca de la posible interrelación y la necesaria cohesión de las distintas métricas y de su utilidad como herramienta integradora dentro del marco de HSBC Argentina.

El Capítulo 4 se basa en un trabajo de campo, donde se lleva adelante una encuesta al grupo de desarrolladores integrado por 170 personas, con el objeto de comprender su conocimiento acerca de las métricas que pueden ser utilizadas y su opinión personal ante las distintas formas de evaluación.

Por último, el Capítulo 5 es una recopilación del pensamiento del Gerente de Desarrollo de Sistemas de HSBC La Buenos Aires Seguros y del Gerente de Desarrollo de Canales de HSBC Argentina. Este capítulo se focaliza en la responsabilidad última sobre los equipos de desarrollo de HSBC Argentina que tiene el nivel gerencial y por lo tanto su necesidad de medir la eficiencia y productividad de los mismos. Asimismo se incorpora la visión global de HSBC a través de una entrevista con el Gerente de Arquitectura y a efectos de contrastar la información obtenida, se agregan conceptos vertidos por el Gerente de Entregas de una empresa de consultoría de primer nivel (Indra SI).

Para finalizar, se incorpora la conclusión del trabajo, proveyendo una visión del problema planteado, a fin de demostrar la necesidad de contar con una metodología de medición y de relacionamiento entre las variables aplicadas para poder implementar un eficiente método de medición de la productividad del equipo de desarrollo de HSBC Argentina.

## CAPÍTULO 1 – EL PERFIL DE LOS EQUIPOS DE DESARROLLO DE SOFTWARE

En su conjunto, un equipo de desarrollo de software debe contar con ciertas características que puedan asegurar el éxito del mismo, el cual se evidencia a través de la calidad del software desarrollado, así como también de la productividad de cada desarrollador. De acuerdo con Lafasto y Larlson<sup>4</sup>, además de otros autores como Reel, algunos de los factores críticos de éxito más reconocidos son los siguientes:

- **La capacidad individual** - En este factor interviene la competencia del individuo y su familiaridad con el área de aplicación.
- **La comunicación entre los miembros del equipo** - Es un factor importante, ya que el trabajo en la mayor parte de las ocasiones no es individual y debe integrarse con el que ha sido desarrollado por otros miembros del equipo.
- **La complejidad del producto** - Este factor depende del tipo de aplicación a desarrollar y es de difícil estimación, ya que muchas veces hasta la fase de desarrollo no es posible comprender en toda su amplitud las complicaciones que conlleva su realización.
- **Utilización de una notación adecuada en el desarrollo de los requerimientos** - Este factor es de gran importancia para facilitar la comunicación entre las partes involucradas (incluido el usuario).
- **Empleo de métodos sistemáticos durante las distintas etapas del desarrollo** - Es importante que se empleen técnicas que sean de amplio consenso y bien conocidas por los integrantes del equipo de desarrollo de la aplicación. También es fundamental que estas técnicas se empleen de manera sistemática para todas las aplicaciones de características semejantes con objeto de facilitar el análisis de costo y tiempo, y también para poder observar la trayectoria profesional de los miembros del equipo.

---

<sup>4</sup> LaFasto, Frank M. J.; Larson, Carl (August 2001), When Teams Work Best, Sage Publications Inc., Thousand Oaks, CA.

- **Conocimiento de los plazos disponibles** - Este factor está vinculado a otros anteriores, ya que es básico conocer el tiempo que puede aportar cada miembro del equipo y en qué plazos, sobre todo en función de las tareas a realizar y de la mayor o menor productividad de determinados integrantes en cada una de ellas.
- **Existencia de facilidades y recursos externos** - Este factor es determinante en la medida en que se puedan utilizar productos o herramientas (automáticas o no) que faciliten las labores de desarrollo e integración de la aplicación. Este aporte es de la mayor significación cuando se conocen, y resulta factible recurrir a aplicaciones similares de fácil transportabilidad y modificación que puedan servir de base a la que hay que realizar.
- **La importancia de la planificación** - Como en el resto de las actividades industriales, comerciales y administrativas, en el desarrollo de software también es importante elaborar una buena planificación del trabajo que facilite una mejor asignación de recursos a los distintos miembros del equipo. Una mala planificación termina en una aplicación disfuncional o en una aplicación terminada a destiempo, o fuera del presupuesto asignado, todo lo cual supone un fracaso.

### **Liderazgo y motivación**

Uno de las principales fortalezas de un equipo es la motivación que puedan tener sus integrantes, la cual debería ser provocada y acompañada por un fuerte liderazgo del encargado del equipo.

Hesselbein y Goldsmith (2006) mencionan características de un líder como ser:

- Exhibir altos niveles de integridad en su moral y conducta ética
- Enfocarse en los resultados
- Construir en las fortalezas propias y ajenas

En relación al campo informático en particular, Klusmann (2009) opina que muchas veces los propios desarrolladores son promovidos a posiciones de liderazgo sin contar con

algunas de estas cualidades ya que poseen una educación técnica y no están preparados para liderar equipos de trabajo.

Sin embargo, muchas características de un líder son fácilmente trasladables a cualquier dominio, especialmente aquellas que hacen a lograr la motivación de los empleados. Overby (2007) describe que es tarea del líder de equipo poder mostrar de una forma comprensible a sus subordinados, las ventajas que otorga cada uno de los desarrollos a los usuarios finales. Puede ser muy frustrante para un desarrollador no saber cuál es el destino de su trabajo y ni siquiera, si va a ser utilizado o no. Esto se inicia desde el CIO<sup>5</sup> y es propagado por todos los niveles del departamento informático. “Se debe crear una cultura que permita a los empleados dar un paso al frente y tomar el liderazgo como parte de su rol”.<sup>6</sup>

Asproni (2004) y Boehm (1981) señalan como factores motivacionales para los desarrolladores de software, los logros obtenidos, donde la programación realizada se convierte luego en una aplicación de negocios que muchas veces brinda la oportunidad a la empresa de incrementar sus ganancias y la base de clientes. Otros factores mencionados por estos autores son el trabajo en sí mismo, el reconocimiento logrado, la supervisión técnica, la responsabilidad y el trabajo bajo presión. Casi siempre existen restricciones de fechas que obligan a los programadores a trabajar arduamente y sin límites horarios para cumplir las metas. Estas restricciones, que terminan ejerciendo una presión sobre los programadores, pueden ser convertidas en una herramienta motivacional, dada por la satisfacción del deber cumplido en tiempo y forma, favoreciendo además la natural competitividad.

Existen numerosas variantes posibles acerca de la composición de un equipo de desarrollo de software. En este trabajo el énfasis se coloca específicamente sobre HSBC Argentina, el cual posee un departamento de desarrollo de sistemas integrado por 170 personas.

En esta instalación, los recursos son parte integrante de la empresa (se dividen en recursos internos y externos), y por lo tanto están bajo responsabilidad directa de los gerentes también pertenecientes a la institución.

---

<sup>5</sup> CIO (*Chief Information Officer*): Máximo responsable de un departamento informático

<sup>6</sup> Overby, S. (2007). How to Turn Your Employees Into Leaders, *www.cio.com*.

Los roles presentes en el equipo son los siguientes:

### **Gerente de Desarrollo**

Tiene a su cargo la administración de todo el departamento de desarrollo, asume responsabilidad sobre la formulación y el cumplimiento del presupuesto del área, el cual está formado casi en su totalidad por los recursos humanos; y es el encargado de fijar las prioridades del equipo en conjunto con las áreas de negocio relevantes.

### **Líder de Equipo**

Está a cargo de determinados proyectos y del mantenimiento de sistemas existentes, tiene una responsabilidad directa sobre los desarrolladores y debe implementar las estrategias definidas por la gerencia, interactuando con todos los actores involucrados.

Tanto el gerente de desarrollo como el líder de equipo son parte integrante del *management* (considerando sus niveles jerárquicos relativos) ya que agregan a su rol específico, habilidades blandas de comunicación, liderazgo, *empowerment* y facilitación del trabajo en equipo.

Por el contrario, se puede hablar genéricamente de ‘desarrollador de software’ a la conjunción de los roles mencionados a continuación, que comprenden a los empleados que diseñan, analizan, codifican y prueban los desarrollos a ser implementados.

Para este tipo de tareas las habilidades más buscadas son la capacidad técnica, la aceptación de trabajar bajo presión y la predisposición para incorporarse a un equipo. Esto adquiere singular relevancia por cuanto se necesita de los demás para poder llevar a cabo un proyecto de forma exitosa.

### **Arquitecto/Diseñador**

Diseña los componentes de software, analiza las interacciones entre los nuevos componentes y los existentes, y tiene a su cargo la definición y mantenimiento del mapa de aplicaciones y la estrategia de implementación, considerando las nuevas tecnologías que incorpora el mercado para el eventual reemplazo de las obsoletas.

### **Analista de Sistemas**

Es la persona que tiene el conocimiento detallado de un sistema en particular. Realiza el análisis de los requerimientos previamente definidos y procede a su interpretación a fin de que los mismos puedan ser codificados. De acuerdo a su experiencia puede ser clasificado como *senior* o *junior*. Un puesto similar en jerarquía es el de Analista Funcional, cuya orientación está un poco más sesgada hacia el entendimiento de los requerimientos del usuario (áreas comerciales, créditos, comercio exterior, finanzas, operaciones, etc.) y a la interacción de éstos con el departamento de sistemas.

### **Analista Programador**

Es el encargado de efectuar los desarrollos propiamente dichos, utilizando el lenguaje de programación y metodologías definidas para ese proyecto o sistema en particular. Recibe las especificaciones del analista de sistemas y efectúa la codificación, que luego de ser testeada, se implementará en producción.

### **Tester**

Una vez efectuado el desarrollo, el tester, en conjunto con el usuario final, tiene bajo su responsabilidad la confección y realización de un plan de pruebas relevante al módulo desarrollado a fin de asegurar la calidad y plena operatividad del producto que va a ser implementado.

## CAPÍTULO 2 – PRODUCTIVIDAD Y EFICIENCIA

En primer lugar conviene precisar la terminología a la que se apela habitualmente para describir la resultante de la productividad de una tarea que oportunamente fuera encomendada al equipo que tomará a su cargo la planificación y ejecución de la misma.

Para ello es posible recurrir a algunas de las definiciones de la Real Academia Española sobre el concepto “producir”. Las que interesan para el tema en cuestión son “elaborar cosas útiles” y “presentar a la vista y examen aquellas razones y motivos o las pruebas que puedan apoyar su justicia o el derecho que tiene en pretensión”. Es decir que cualquier referencia o empleo que se haga de la palabra productividad finalmente busca evaluar la calidad de lo producido.

Es necesario entonces, avanzar un paso ya que se debe calibrar si el resultado obtenido se ajusta a los parámetros y especificaciones que se fijaron para llevar adelante el desarrollo encomendado, y que, finalmente, el rendimiento o utilidad se ajusta al propósito perseguido.

Jayamaha y Mula (2011), en efecto, proveen una definición funcional de productividad relacionando las salidas y las entradas (*outputs* e *inputs*) de un proceso en particular. Por lo tanto hacen especial énfasis en no sólo lograr un resultado (*output*) sino también en la cantidad de recursos (*input*) necesarios para lograrlo.

Los autores mencionados introducen el concepto de funcionalidad para juzgar la correspondencia entre el propósito y el logro, pero aún se debe recorrer otro trecho más para comprender cabalmente al sentido que se le otorga a la productividad en el ámbito del *management*.

En efecto, no es aceptable para dicho entorno que para obtener un resultado análogo, se hubieran aplicado recursos cuya magnitud y cuantía sean disímiles. Esta diversidad incide directamente en la calificación de la gestión y en el gerenciamiento de la tarea.

Coelli (1998) menciona que en la práctica cotidiana se apela a algunos conceptos que suelen utilizarse indistintamente, casi como sinónimos, en la mayor parte de los casos a pesar de que su significado en muchos casos sea precisamente el opuesto. Sin embargo, reconocer la circunstancia en la que se debe aplicar uno u otro, implica un hecho esencial para entender con plenitud los métodos que conducen a la optimización de la productividad. Resulta necesario entonces explorar la imprecisa frontera que separa la eficiencia de la eficacia, términos éstos que se aplican usualmente para valorar un resultado, satisfactorio o no, de una actividad, procedimiento o producto.

Sólo así se puede comenzar a comprender y utilizar las herramientas de medición que posibilitarán el seguimiento y control de la ejecución y, por consiguiente, la consideración que recibirá el producto terminado de todos aquellos que deberán juzgarlo.

Se desprende de los párrafos precedentes que se recurre a calificar un proyecto como eficiente si se establece una relación adecuada y conveniente entre los costos materiales y humanos que insume una actividad y si estos están en consonancia con los logros obtenidos. Es decir que aumenta la eficiencia cuando se utilizan menos recursos para obtener el objetivo fijado. También se aplica cuando se pueden lograr objetivos adicionales con los mismos recursos.

En cambio, eficacia apunta directamente a la consecución satisfactoria de un objetivo, con prescindencia de los recursos necesarios para lograrlo. Coelli (1998) brinda un compendio muy interesante de los términos en cuestión. Específicamente, muestra la diferencia entre los términos “productividad” y “eficiencia”, pese a reconocer que pueden llegar a ser utilizados indistintamente. Coelli explica que si existe una manera diferente de producir un mejor o el mismo resultado, y esa manera no utiliza más recursos de entrada, entonces el producto final se puede producir de una forma más eficiente.

Evidentemente, esto último es la tendencia imperante en el mundo de los negocios, cuyas prácticas se han debido subordinar a la presión que exige cada vez un mayor nivel de rentabilidad y competitividad.

Resumiendo, la eficiencia apela a términos propios de los recursos (dinero, tiempo, esfuerzo), mientras que la eficacia se mide en términos del resultado final (calidad, perfección, excelencia).

### **Productividad ¿cuantitativa o cualitativa?**

Los procesos a los que aportan la logística y la fuerza laboral de un establecimiento industrial de una dimensión equiparable a la de la entidad financiera sujeta a este análisis, desembocan finalmente en una máquina que, en forma repetitiva, manufactura un tipo determinado de productos, idénticos unos de otros.

Ebert (2006) agrega que entonces es posible medir la productividad recurriendo a fórmulas sencillas, tales como “total de productos fabricados / insumos necesarios”, o “cantidad de productos elaborados / estándar del modelo”.

Cada uno de los términos de estas fórmulas puede ser desagregado, lo que permite analizar, y eventualmente incidir para lograr una mejoría; se trata de factores tales como mano de obra, energía, proveedores, equipamiento, y tantos otros. A su vez, los estándares pueden ser varios ya que es posible obtener datos y estadísticas históricas de la propia actividad, del respectivo sector económico o, directamente, apelar al *benchmarking* como ejemplo con el cual compararse. Por último, Ebert menciona la influencia de los factores ambientales (complejidad, restricciones de calidad y de tiempo, capacidad de los procesos, etc.) como parte integrante de la ecuación.

El autor cita sin embargo la problemática de la medición de la productividad en áreas orientadas al desarrollo (software): “El crecimiento pasado ha sido logrado mayormente por el hardware, mientras que el software aún es una industria artesanal”.<sup>7</sup>

---

<sup>7</sup> Alcatel (2006). *Measuring and Improving Productivity*. Zurich: Ebert, C.

Por su parte, Poile (2009) y Overby (2007) mencionan que un factor adicional de complicación es el que surge del carácter individual de cada una de las personas que integran el equipo, que presentan rasgos disímiles en cuanto a experiencia, remuneración, motivación, colaboración, y tantos otros factores que resultan ya competencia de la gestión de los recursos humanos. Adicionalmente, la labor de todos esos individuos, cada uno con sus particularidades, debe ser coordinada en pos de un logro común.

Los autores implican que la medición de un equipo de trabajo no puede reducirse únicamente a la aplicación de un método formal y mecanicista, sino que cada uno de estos equipos, al igual que cada uno de los individuos que lo componen, deben ser evaluados de acuerdo a su idiosincrasia.

Otro tema de importancia es mencionado por Beatty (2005), el cual resalta como evidente la dificultad que se le presenta a cualquier nivel gerencial superior del área de informática, que se haya visto en el trance de tener que dar cuenta de su trabajo a un grupo directivo carente de una formación específica, o siquiera general en la materia, o cuyo *background* tenga un origen proveniente de la contabilidad o la ingeniería. Deberá entonces afrontar preguntas y aún severos cuestionamientos, derivados de la aplicación de conceptos y fórmulas contables o industriales, que revelan una clara incomprensión de las características propias de la actividad cuyo resultante se ha logrado mediante el exclusivo empleo del intelecto.

A su vez, el *CIO* ha debido relegar sus conocimientos especializados para seguir las tendencias del mercado, que requieren su compenetración con las estrategias comerciales enfocadas en la rentabilidad que rigen hoy en el mercado y en las directrices emanadas de los accionistas. Por lo tanto, es posible el diálogo y comprensión de las necesidades mutuas.

Resumiendo, se trata de una combinación de factores de apreciación cualitativa y no cuantitativa. Queda entonces abierto el camino de la aproximación a través de metodologías utilizando las métricas que mejor se adapten al control del equipo en cuestión.

## La productividad y la remuneración

Por último, se puede ir un poco más allá en el concepto de productividad, considerando la posibilidad de un pago variable relacionado justamente con la productividad alcanzada.

Este concepto no es utilizado actualmente por HSBC, donde en el área de sistemas se retribuye el trabajo con sueldos fijos (los cuales obviamente varían dentro de rangos sugeridos según la especialización y la experiencia de los profesionales), y también en algunos casos el pago se realiza por hora trabajada, aunque el valor de la misma es fijo, teniendo en cuenta la consideración anterior.

Bikker (2009) expresa algunas posibles mediciones dentro del sector de servicios financieros acordes a la remuneración variable. Por ejemplo, este esquema se utiliza con la fuerza de ventas, donde además de un componente fijo, el salario incorpora una porción variable (en muchos casos de una gran incidencia), de acuerdo al cumplimiento y la cuantía de los negocios logrados o de las ventas realizadas.

Llevar adelante esta modalidad remunerativa a nivel organizacional puede resultar una ardua tarea pues se deberán resolver algunas cuestiones plagadas de dificultades.

Atchison y Belcher (1987) mencionan que la forma de retribución debe integrarse a la política de remuneraciones de la respectiva entidad. Sin embargo coinciden en afirmar las ventajas de una estrategia en la que los beneficios laborales se encuentren atados directamente al aporte que cada individuo hace a la rentabilidad de su organización.

Se obtiene así una mayor flexibilidad en la administración de las compensaciones por el trabajo llevado a cabo, ya sean monetarias o como beneficios sociales. Además, desde el punto de vista patronal aumenta el *engagement*, fortalece la equidad de la estructura salarial, al mismo tiempo que depura la dotación de personal no idóneo.

Según los autores, los principales factores a tener en cuenta para implementar un plan de estas características son los siguientes:

- Posibilidad de medir el trabajo realizado (*output*).
- La relación existente entre el trabajo realizado y el esfuerzo demandado.
- La estandarización presente en la tarea.
- Requerimientos de calidad y cantidad.
- Las condiciones competitivas.

Por todo ello, cualquier intento por introducir esta modalidad requiere su total integración a la cultura laboral de la entidad en cuestión. Otra dificultad es inherente a la tarea en sí, en el sentido (como se verá más adelante), de que es prácticamente imposible fijar una unidad de medida objetiva y aceptada sin reparos, para evaluar la productividad del equipo de desarrolladores. Si fuera imperioso contar con una métrica en particular o de una combinación de ellas, la estructura de beneficios deberá ser consensuada por los integrantes del grupo o, como mínimo, por sus líderes de equipo; asimismo deberá ser fácilmente comprensible y difundida ampliamente.

En resumen, la remuneración variable, una temática muy compleja que excede los límites de este trabajo, presenta ventajas pero también inconvenientes, por lo que no resulta fácil conciliarlos, adaptarlos a la cultura de la entidad y, por último, lograr una armoniosa y pacífica implementación.

## CAPÍTULO 3 – LAS DISTINTAS MÉTRICAS DE PRODUCTIVIDAD UTILIZADAS EN EL DESARROLLO DEL SOFTWARE

Existe una gran cantidad de autores que han definido y analizado distintas hipótesis sobre el tema bajo análisis, arribando a diversas y, en algunos casos divergentes conclusiones.

De los mismos se rescatan aquellos conceptos válidos tanto por su validez conceptual como por su utilidad en la práctica cotidiana.

Uno de los principales autores que se ha dedicado al tema es Capers Jones. Su visión acerca de la forma de encarar los desarrollos de software se puede resumir citando: “El software de alta calidad no es caro; en realidad es más barato y más rápido de construir que el software de baja calidad, desde los desarrollos iniciales hasta el costo total de propiedad del mismo”.<sup>8</sup>

Seguidamente se muestra una síntesis de las principales métricas de productividad en el desarrollo del software. Algunos de los autores consultados son Mc Connell (2008), Fowler (2003), Fenton (1998), Mills (1988) y Boehm (1981, 1998).

### Líneas de Código

Esta métrica denominada LOC (*Lines of Code*) es una de las más utilizadas ya que se caracteriza por la simplicidad de su aplicación. Consiste en poder efectuar una mensura del tamaño de un programa de software mediante el recuento de la cantidad de líneas que componen el código fuente de dicho programa.

Este procedimiento es usado tanto para estimar el esfuerzo que será requerido para elaborar un programa, como así también para estimar la productividad del trabajo de programación o de mantenimiento del software, una vez que el desarrollo ya ha sido producido.

---

<sup>8</sup> Jones, C. & Bonsignour, O. (2011), *The Economics of Software Quality*, pp.520

Mills (1998)<sup>9</sup> comenta que LOC fue una de las métricas utilizadas desde los principios de la era informática, ya que los lenguajes de programación utilizados en esos momentos (*Assembler, FORTRAN, COBOL*) son fácilmente medibles por presentar una estructura fija de sentencias por línea como objetos discretos que pueden ser fácilmente contados.

Martin Fowler (2003)<sup>10</sup> sugiere que la implementación exitosa de esta técnica debe tener en cuenta la aplicación de un orden de magnitud cuando se procede a la medición de las líneas de código. Se considera mucho más útil apelar a este método para comparar un desarrollo de 20.000 líneas con uno de 200.000 líneas (orden 10x de magnitud) que para cotejar ese mismo desarrollo de 20.000 líneas con uno de 21.000 o 22.000 líneas.

El autor agrega que no considera válida la aplicación de este método debido a la diferencia en cantidad de líneas que trae aparejado el uso de distintos lenguajes de programación. Es así que se le plantea la posibilidad a un ingeniero de software de desarrollar un mismo módulo utilizando una cantidad mayor o menor de líneas, de acuerdo a la estructura del programa utilizada o a la reusabilidad de código que pudiera emplearse basándose en un programa ya desarrollado anteriormente.

Otras falencias atribuibles a este método, son que no tiene en cuenta la experiencia del desarrollador, lo cual constituye es un factor fundamental en este escenario, ya que un buen programador que tenga una experiencia considerable, muchas veces puede codificar una misma funcionalidad utilizando menos líneas de código que un programador *junior*.

También se puede mencionar que en muchos entornos modernos de programación, el código a ser ejecutado es automáticamente generado a partir de instrucciones básicas ingresadas por el desarrollador y por lo tanto no es factible tomar como medida la cantidad de líneas presentes en dicho código, que es el utilizado cuando el sistema está en el ambiente de producción.

---

<sup>9</sup> Mills, E. (1988). *Software Metrics*, *Software Engineering Institute*, Seattle University.

<sup>10</sup>Fowler, M. (2003). Cannot Measure Productivity, <http://martinfowler.com/bliki/index.html>

Actualmente, se consideran mayores las desventajas de este método que sus posibles beneficios, por lo que solamente se utiliza en casos muy puntuales y siempre acompañado de otras métricas.

## **Puntos de Función**

El método de Puntos de Función (*Function Points*) fue definido por Albrecht (1979) en IBM<sup>11</sup>, aún con diversas modificaciones introducidas posteriormente, sigue siendo utilizado en la actualidad.

Ebert (2006)<sup>12</sup> menciona que dicho método está basado en la medición de la complejidad de la funcionalidad de negocios que cada programa o módulo desarrollado entrega al usuario final. Para ello, se tipificaron los requerimientos funcionales del usuario en cinco tipos:

- Salidas (*outputs*)
- Entradas (*inputs*)
- Consultas (*inquiries*)
- Archivos internos (*internal files*)
- Interfases externas (*external interfaces*)

Cada requerimiento es clasificado y también ponderado con respecto a su complejidad. En las últimas versiones del modelo, se utilizan variables como ser la complejidad del programa, el uso de objetos predefinidos y la cantidad de cálculos y algoritmos que contiene el programa, a fin de poder realizar una mejor ponderación de los puntos de función relevantes.

---

<sup>11</sup> A. J. Albrecht, "Measuring Application Development Productivity," Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, October 14–17, IBM Corporation (1979), pp. 83–92.

<sup>12</sup> Alcatel (2006). *Measuring and Improving Productivity*. Zurich: Ebert, C.

Por último, el modelo asigna un puntaje final a cada módulo a efectos de ser utilizado como métrica de comparación entre distintos desarrollos efectuados por los mismos o diferentes equipos de trabajo.

Este modelo ha sido reconocido bajo las siguientes normas ISO:

- COSMIC-FFP: ISO/IEC 19761:2011 *Software engineering. A functional size measurement method.*
- FiSMA FSM: ISO/IEC 29881:2008 *Information technology - Software and systems engineering - FiSMA 1.1 functional size measurement method.*
- IFPUG FSM Method: ISO/IEC 20926:2009 *Software and systems engineering - Software measurement - IFPUG functional size measurement method.*
- Mk II Function Point Analysis: ISO/IEC 20968:2002 *Software engineering - Mk II Function Point Analysis - Counting Practices Manual.*
- NESMA FPA Method: ISO/IEC 24570:2005 *Software engineering - NESMA function size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis.*

### **Cantidad de errores en los programas y su repetición**

El error en un programa de software tiene una definición formal, ya que puede ser visto como “cualquier desviación sobre el comportamiento especificado del programa”.<sup>13</sup>

Una forma muy utilizada para poder mensurar esta métrica es la denominada densidad de defectos del sistema. Según McConnell<sup>14</sup>, es la que se establece como la relación entre la cantidad de defectos encontrados y el tamaño del sistema en cuestión.

---

<sup>13</sup> Fenton N. & Pfleeger S. (1998). *Software Metrics: A Rigorous & Practical Approach*, pp 23

<sup>14</sup> McConnell, S. (1997). *Gauging Software Readiness with Defect Tracking*, *IEEE Software*, 14, 3, pp 136.

Para poder implementar una aplicación práctica de este modelo en forma satisfactoria, es importante definir claramente las distintas etapas del desarrollo, utilizando conceptos de gerenciamiento de proyectos (*project management*) para así poder mensurar la cantidad de defectos en cada ciclo o etapa del desarrollo.

De acuerdo con la metodología estandarizada por el *Project Management Institute* (PMI), los defectos encontrados en las fases más tempranas son de más fácil y barata resolución. Luego de superadas las etapas de definición de requerimientos, diseño y análisis detallado, se comienza la etapa de programación en la cual se complica progresivamente la localización del error y su consecuente solución.

El código resultante de esta tarea, será ejecutado y testeado en diferentes etapas, las cuales típicamente son el testeo unitario (*unit testing*), testeo de aceptación del sistema (*system acceptance testing*), testeo de aceptación del usuario (*user acceptance testing*) así como también algunos testeos más técnicos orientados a medir la performance del programa bajo condiciones extremas (*stress testing*) y su inserción dentro del ambiente operativo en el cual va a ser instalado (*operations acceptance testing*).

Usualmente, los defectos y errores encontrados en cada uno de los pasos de prueba utilizados son registrados a fin de asegurar su seguimiento y resolución. Este registro finalmente es utilizado para la confección de la métrica ya que posee información inherente al defecto en sí, pero también datos que se refieren a su reincidencia, tiempo de resolución, impacto en el producto final, etc.

### **COCOMO (*Constructive Cost Model*)**

Este método fue desarrollado por Barry Boehm<sup>15</sup> en el año 1981 y se basa en un modelo matemático para la estimación de costos en un desarrollo de software.

---

<sup>15</sup> Boehm, B (1981) *Software Engineering Economics*, Prentice Hall, New Jersey

Boehm también había desarrollado el método Delphi, utilizando expertos que pudieran aportar información relevante a la estimación, pero la introducción de COCOMO significó un avance en este campo, al poder emplear parámetros objetivos en el proceso de estimación.

El método provee una estimación iterativa de 3 capas (Básico, Intermedio y Detallado) donde en cada iteración se agregan atributos y se refinan los previamente existentes.

Algunas de los atributos definidos por Boehm son: tamaño de las bases de datos, complejidad del producto, limitaciones de memoria y hardware, experiencia de los programadores, uso de herramientas, etc.

Este modelo fue estudiado por numerosos autores e investigadores entre los cuales se pueden citar a Abts, Clark y Horowitz. En el año 1998, estos autores participaron junto con Boehm en el desarrollo de una nueva versión del modelo, denominada COCOMO II<sup>16</sup> la cual introduce mejoras sustanciales orientadas a las nuevas tecnologías y lenguajes de programación, a la vez que mantiene los principios fundamentales del modelo original.

Este modelo es más integrador que los mencionados anteriormente ya que incorpora datos de líneas de código y puntos de función así como conceptos de reusabilidad de código y mantenimiento de aplicaciones, que previamente eran utilizados en forma individual.

## **Conclusión**

Un buen análisis de las distintas métricas mencionadas es el realizado por Ebert (2006) que menciona que “No existe ninguna metodología, herramienta o estrategia de mejoras de procesos que por sí sola logre mejoras revolucionarias en eficiencia de procesos”.

---

<sup>16</sup> Boehm, B, Abts, C, Devnani-Chulani, S et al (1998) COCOMO II Model Definition Manual, Center for Software Engineering, USC.

Sin embargo, Ebert menciona una conjunción de factores que apuntan a la confiabilidad de las métricas entre los que se pueden mencionar disciplina de procesos, estándares (lenguajes, formularios), diseños de calidad y costo y conceptos de agilidad.

Por su parte, Fenton y Neil (1998) coinciden en que la profusión de métricas de software e investigaciones al respecto debe considerarse como un área y un tema en expansión. Asimismo remarcan la necesidad de profundizar las aplicaciones reales de dichas técnicas en instalaciones de desarrollo a fin de mejorar la efectividad de las mismas como herramientas de toma de decisión para los niveles gerenciales, a la vez que puedan administrar de mejor manera una combinación de factores conteniendo altos niveles de incertidumbre.

## **CAPÍTULO 4 – TRABAJO DE INVESTIGACIÓN: ¿CÓMO SON MEDIDOS LOS DESARROLLADORES?**

En este capítulo se desarrolla un trabajo de campo, realizado en los meses de abril y mayo de 2012. Se diseñó una encuesta utilizando la herramienta online *Survey Monkey*.

La encuesta incorpora 3 preguntas para recabar datos demográficos, como ser la edad, años de experiencia de cada desarrollador y educación alcanzada, pero no incursiona en otros que no resultan relevantes para la investigación.

A continuación se incluyen 6 preguntas con el objetivo de comprobar el grado de conocimiento de los integrantes de equipos de desarrollo de software acerca de las métricas que pueden ser utilizadas para medir su productividad como así también su reacción ante las distintas formas de evaluación.

Por último se realizó una última pregunta abierta a fin de que los encuestados pudieran desarrollar sus ideas sobre el tema específico de la remuneración variable.

Luego de diseñada la encuesta, la misma fue evaluada y comentada a modo de prueba piloto por 4 integrantes de un equipo de desarrollo, además de 2 personas de nivel jerárquico.

Una vez realizados los ajustes correspondientes, la invitación optativa a contestar la encuesta fue enviada vía mail a un universo de 170 integrantes de los equipos de desarrollo de HSBC Argentina, recibándose en un plazo de 24 horas la cantidad de 96 respuestas (56%).

En modo informal se recibieron comentarios positivos acerca de la encuesta, evidenciándose un notable interés por el tema planteado.

Se incluyen en el Anexo 1 las preguntas utilizadas y sus correspondientes respuestas.

## Resultados obtenidos

Las respuestas obtenidas muestran una interesante disparidad entre los encuestados, destacándose como un factor importante la experiencia en el área de desarrollo de sistemas.

En efecto, casi el 60% de los encuestados posee más de 15 años de experiencia, con solamente un 5% con una experiencia inferior a 5 años en un área de desarrollo.

En lo que respecta a la edad de los encuestados, el 38% tiene entre 31 y 40 años, con un porcentaje muy similar distribuido entre los 41 a 50 años.

Esta información contradice la creencia popular de que la tecnología se encuentra en manos de la juventud. Por el contrario evidencia un notorio corrimiento hacia la franja etaria que llega hasta los 50 años, incluso con un porcentaje no despreciable (15%) que la ha superado. La conclusión es obvia, pero es relevante destacarla. La tarea de los desarrolladores (analistas de sistemas y programadores) requiere un grado de pericia que sólo muchos años de experiencia pueden brindar.

Es evidente que ese factor ostenta tal solidez (valorada consecuentemente por la entidad) que ha podido resistir las sucesivas oleadas racionalizadoras del sistema financiero argentino que se han caracterizado por drásticos *downsizings*, por la informatización de la totalidad de la actividad y por la reducción de los costos operativos.

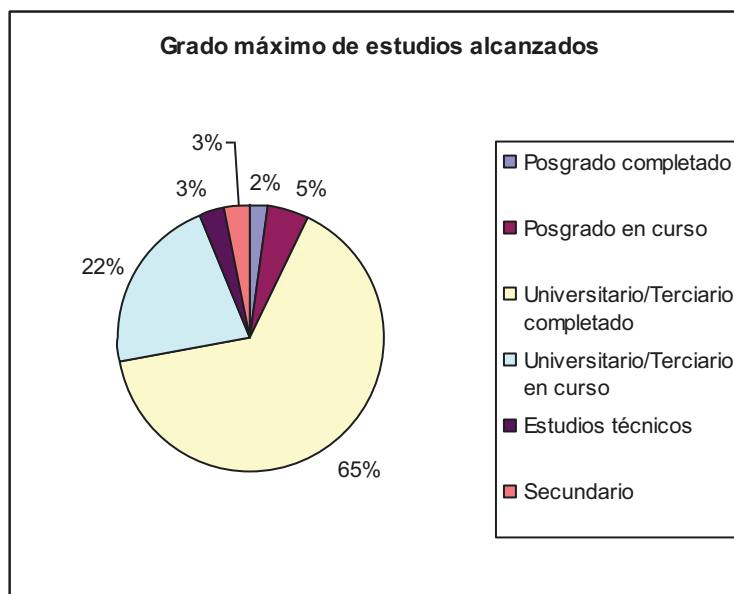
Así, en la mayoría de las áreas de las entidades financieras, se ha desplazado a los empleados y funcionarios de mayor edad, de conocimientos generalistas y con una formación conservadora, reemplazándolos por una generación que percibe menores sueldos, acepta fácilmente las novedades y cambios y se adaptan a las tareas informáticas por ser nativos informáticos, que crecieron y se educaron en el uso de las nuevas tecnologías.

Esto, un axioma indiscutible en el ámbito bancario, se contrapone a los reveladores resultados de la encuesta que descubre una situación diferente en el área de desarrollo lo cual conduce a una conclusión ambivalente; por un lado la combinación de edad,

experiencia y profesionalidad facilita la supervisión, pero por el otro delimita un tipo de dotación que requiere ser administrada a través de una capacidad de liderazgo y motivación con características especiales.

El nivel de estudios alcanzados por la población estudiada es muy relevante, ya que el 72% tienen estudios universitarios completos, y un 22% adicional se encuentra actualmente realizando estudios universitarios o terciarios (ver Gráfico N° 1).

GRÁFICO N° 1



Fuente: investigación propia

En referencia al rol de cada integrante dentro del equipo, se observa una clara mayoría de analistas de sistemas y analistas programadores (92%). Son justamente estas dos funciones las más relacionadas con el tipo de medición que se quiere efectuar, al tener ambos roles la asignación, dentro de la descripción de su posición de trabajo, de la generación de códigos y el desarrollo de programas.

Como hemos visto, la productividad de un desarrollador está íntimamente ligada a su interacción y sinergia con el equipo de trabajo.

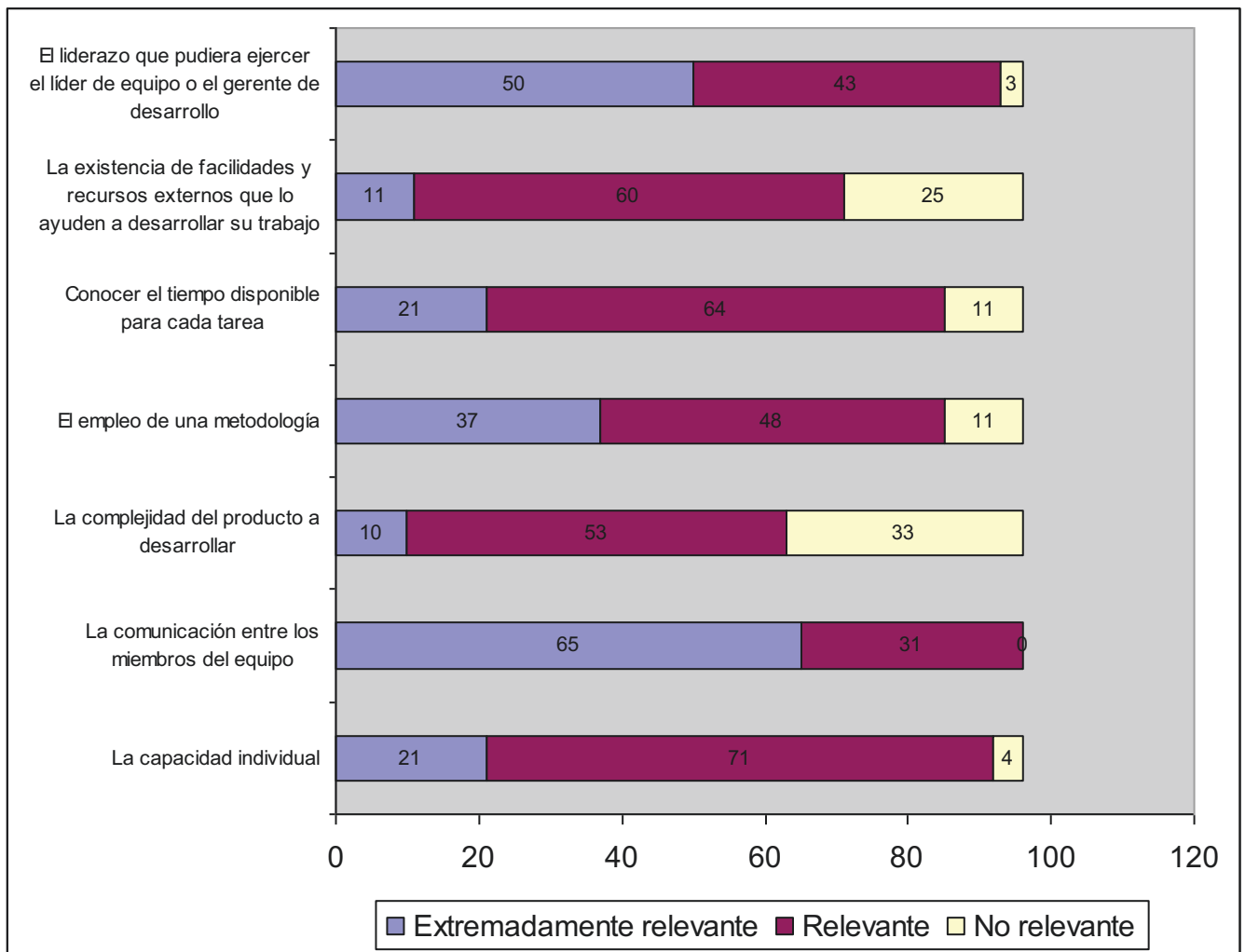
Por este motivo, es importante conocer la opinión de los desarrolladores acerca de este punto y específicamente a cuáles de los posibles factores de éxito dentro de un equipo de desarrollo le asignan una mayor importancia.

Las respuestas obtenidas son de particular importancia para una investigación futura, pudiéndose complementar con la opinión de gerentes a cargo de equipos de desarrollo, como para generar una visión del área que pueda interrelacionar ambas visiones, logrando integrar los intereses de todos los recursos, tanto materiales como humanos, y pudiéndose así lograr equipos exitosos y más efectivos.

Las cifras obtenidas en relación a las apreciaciones precedentes confirman el enfoque que prioriza la capacidad intelectual de los integrantes del grupo, puesta al servicio de la resolución de un proyecto desafiante.

En el Gráfico N° 2 se puede observar que la variable más relevante en lo que respecta a la pertenencia dentro de un equipo es la comunicación entre sus miembros ya que el 68% de los encuestados lo considera un factor extremadamente relevante. Asimismo, el 52% opina que el liderazgo que muestra la persona a cargo del equipo es igualmente importante.

Por el contrario, factores que a priori podrían ser considerados como extremadamente críticos, no lo son tanto en la visión de los desarrolladores ya que les asignan una relevancia menor. Entre estos factores están la capacidad individual, la complejidad del producto a desarrollar, y la existencia de facilidades y recursos externos.

**GRÁFICO N° 2**


Fuente: investigación propia

A continuación se incorporó una pregunta a fin de introducir el tema de la medición de la productividad y también para lograr el interés y compromiso de los encuestados en el campo a ser estudiado. Las respuestas a esta pregunta demuestran claramente la importancia de la productividad que la llevan a ser considerada como un dato clave en la forma de medir el trabajo. El 35% está totalmente de acuerdo con este concepto y el 60% comparte la idea parcialmente.

Se comprueba así que hay un consenso generalizado, propio de la naturaleza humana, así como también de las reglas que regulan cualquier organización, de que el desempeño individual y grupal debe ser evaluado pues de la equidad del juicio será posible discernir el premio o castigo que corresponda.

Luego se efectuaron dos preguntas tendientes a conocer la opinión de los profesionales en lo que respecta a los métodos de medición.

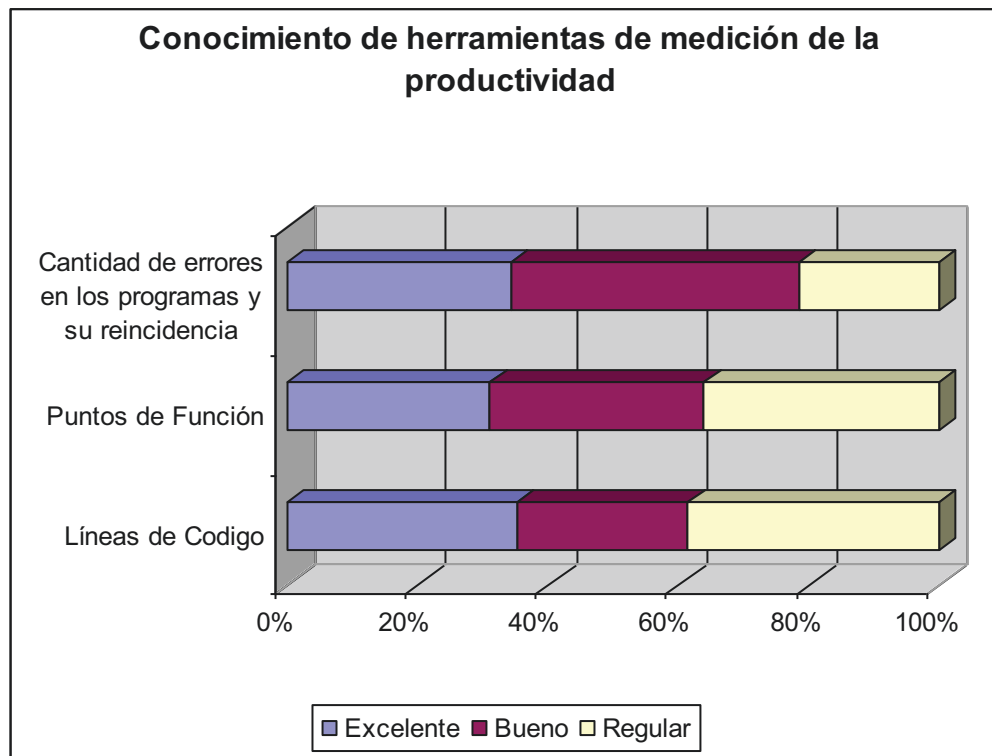
Primeramente se buscó determinar el conocimiento del mercado y las técnicas que pudieran existir. En este punto, solamente el 12% está totalmente de acuerdo en que dichas herramientas existen y están disponibles en el mercado, con un 63% opinando en el mismo sentido pero no de forma categórica.

En segundo lugar se midió el conocimiento específico de algunas de las métricas más conocidas y utilizadas en la práctica, que son las que se han descrito en este trabajo. Pese a la importante experiencia y estudios alcanzados que posee la población analizada, casi un 70% considera tener un regular o buen conocimiento de dichas métricas dejando sólo al 30% restante con un conocimiento excelente.

Por lo tanto, en cuanto a las alternativas posibles para alcanzar una meta de medición apropiada, los encuestados presentan sus reparos, confirmando empíricamente uno de los postulados de este trabajo.

En el Gráfico N° 3 se pueden apreciar estos resultados.

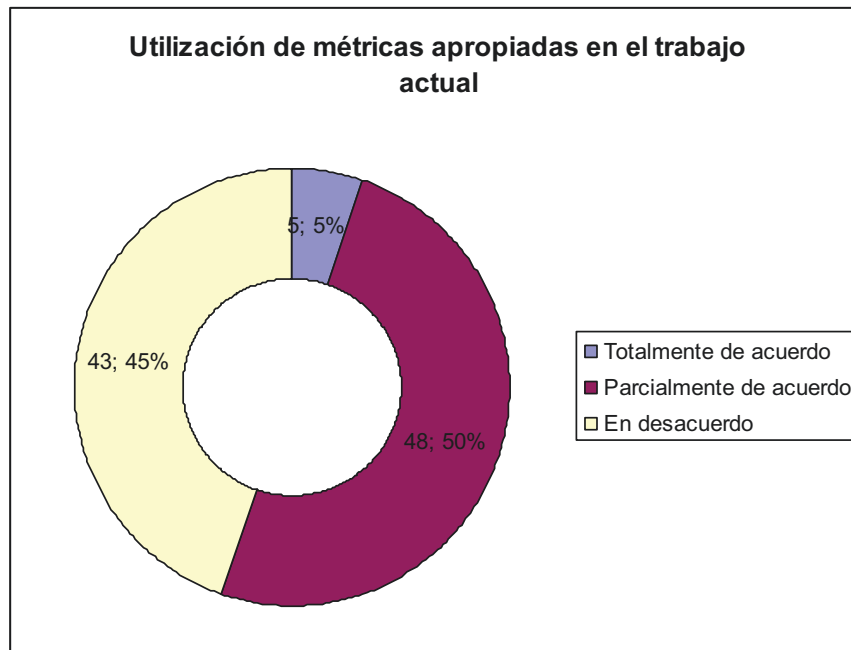
GRÁFICO N° 3



Fuente: investigación propia

En relación al trabajo puntual que cada persona lleva a cabo en la actualidad, se preguntó si justamente las herramientas mencionadas eran utilizadas efectivamente para poder medir la productividad. En el Gráfico N° 4 se puede observar que el 45% (43 respuestas) considera directamente que no se utilizan efectivamente dichas herramientas y el 50% opinó que se utilizan solamente en parte o en forma incompleta. La realidad de todos los días se refleja en estas respuestas, al ser mayoritaria la opinión de que no se están aplicando las herramientas adecuadas. Esto constituye un desafío que no puede ser dejado de lado y evidencia la necesidad de profundizar el tema por parte de los niveles gerenciales a fin de lograr una mayor penetración y comprensión de las métricas utilizadas.

GRÁFICO N° 4



Fuente: investigación propia

La última pregunta apuntaba a obtener la conclusión de cada encuestado acerca de la efectividad de las métricas para medir la productividad, al relacionar este tema directamente con la posibilidad de cambiar el esquema de remuneración actual a un esquema variable.

Esta pregunta abierta permitió obtener diversas opiniones, las cuales se encuentran muy divididas. Un 54% de los encuestados no cree correcto la aplicación de una remuneración variable mientras que un 46% está de acuerdo con este concepto, aunque en la mayoría de los casos se expresan distintas consideraciones tanto a favor como en contra.

Dada la riqueza de algunas respuestas, conviene reproducir las más significativas. A tal efecto, en el Anexo 1 se citan las más representativas de cada posición.

## **CAPÍTULO 5 – EL GERENCIAMIENTO DE EQUIPOS DE DESARROLLO Y LA MEDICIÓN DE LOS MISMOS EN HSBC ARGENTINA**

El trabajo de investigación se completa mediante cuatro entrevistas. Las entrevistas se desarrollaron en dos etapas entre los meses de abril y junio de 2012, en las instalaciones de HSBC Bank Argentina S.A. y de Indra SI S.A., ciudad de Buenos Aires, y el método utilizado consistió en preguntas abiertas a fin de que cada tema pueda ser desarrollado en profundidad por los entrevistados.

En primer lugar, Luis Ferreyra, Gerente de Desarrollo de Sistemas de HSBC La Buenos Aires Seguros, explica la composición de su equipo de trabajo compuesto por 23 personas así como también las métricas utilizadas en la medición de la productividad del mismo y brinda una visión general de la forma de calificación por objetivos que se utiliza en su entidad. Ferreyra cuenta con 25 años de experiencia en desarrollo de sistemas y durante los últimos 12 años se ha desempeñado a cargo de equipos de trabajo.

A continuación se entrevistó a Fabián Bernard, Gerente Senior de Desarrollo de Canales de HSBC Argentina, quien se inclina por un estilo de liderazgo motivacional, brindando ideas que puedan ser desarrolladas más ampliamente con el objetivo de optimizar las métricas y fomentar su utilización. Bernard trabaja hace 20 años en el área de desarrollo y posee 13 años como líder de distintos equipos. En la actualidad está a cargo de 45 personas.

Los niveles gerenciales representados por Ferreyra y Bernard son los que poseen la responsabilidad última sobre los equipos de desarrolladores; el desempeño de sus áreas como un todo, es evaluado por la eficiencia y productividad de los mismos.

La guía de preguntas utilizadas fue la siguiente:

- ¿Cómo está formado su equipo de trabajo? ¿Cuáles son las características del mismo en base a la experiencia de sus miembros y a las tecnologías utilizadas?
- ¿Utiliza métricas de medición de la productividad?
- ¿Qué opina de los métodos tradicionales de medición como ser líneas de código, puntos de función, etc.?
- ¿Cree que el desempeño de los integrantes de su equipo tiene que ser medido en forma ‘dura’ u objetiva o que deben existir componentes subjetivos en esa medición?
- ¿Aceptaría que como incentivo, una parte del sueldo de sus equipos sea calculada en forma variable en base a la productividad de cada integrante del mismo?

Con el objetivo de conocer más en profundidad la aplicación de métricas en HSBC a nivel global, se entrevistó a Juan Sorroche, Gerente de Arquitectura IT y Business Intelligence de HSBC Argentina. Sorroche está a cargo del liderazgo en lo referente a la adopción en HSBC Argentina de prácticas y tendencias tecnológicas que se utilizan en otros centros de HSBC a nivel mundial. La entrevista a Juan se basó principalmente en poder comprender qué tipo de métricas ya son utilizadas en otras partes de HSBC y cómo se podrán aplicar a HSBC Argentina.

Por último se incorpora la visión de Marcelo Farías, Gerente de Entrega de Indra SI S.A., con el objetivo de poder contrastar las prácticas utilizadas en HSBC Argentina con aquellas que son de aplicación en una empresa de primer nivel de desarrollo de servicios. Farías explica el proceso metodológico bajo el cual se miden los proyectos encarados por la compañía.

## Entrevista Nro. 1: Luis Ferreyra

El equipo de trabajo a cargo de Ferreyra involucra varias tecnologías, ya que administra sistemas bajo plataforma *i-series*<sup>17</sup> y sistemas bajo arquitectura distribuida<sup>18</sup>. Dentro de ésta última existen sistemas basados en la Web y aplicaciones desarrolladas utilizando los lenguajes de programación *Visual Basic*, *C++* y *JAVA*.

Los desarrolladores de la plataforma *i-series (AS400)* cuentan con muchos años de experiencia en programación, mientras que los recursos que trabajan con arquitecturas distribuidas tienen una experiencia media y algunos son más jóvenes. Uno de los principales objetivos de la gerencia es que se forme una buena química de trabajo entre las dos generaciones (existen recursos de 25 años y otros de casi 60 años de edad).

Ferreyra comentó que en su área se utilizan métricas para medir la productividad, específicamente las orientadas a determinar los errores de los programas, las cuales se dividen en los siguientes métodos de medición:

### Cancelaciones de los procesos (*batch*<sup>19</sup>)

Se producen al interrumpirse la ejecución de un programa que se encuentra preparado para no necesitar intervención humana (típicamente cálculo de intereses, generación de reportes, etc.). Las mediciones son relevantes para determinar la calidad de la programación que se ejecuta en el ámbito de producción. En el Gráfico N° 5 se puede ver la evolución de esta métrica.

---

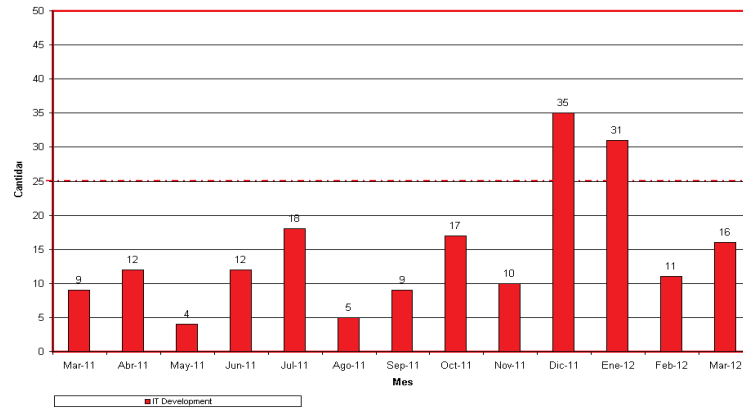
<sup>17</sup> *i-series* es una tecnología desarrollada por IBM que apunta a empresas de envergadura. Su complejidad es intermedia y los equipos utilizados en su implementación se denominan AS400.

<sup>18</sup> Se denomina arquitectura distribuida a aquellos sistemas desarrollados sobre varias capas en distintas plataformas como ser Windows, Linux, etc.

<sup>19</sup> Son aquellos programas que se ejecutan típicamente de noche y no requieren de la presencia del usuario para operarlos (en contraposición a los programas *online*)

GRAFICO N° 5

### Batch Disruption – IT Software Delivery



Fuente: HSBC

### Incidentes productivos (*online*)

Estos incidentes (ver Gráfico N° 6) son reportados por los usuarios durante la operatoria normal del sistema. Los motivos que surgen como generadores de este tipo de incidentes son un proceso de testing deficiente, problemas en las definiciones emitidas por el usuario, errores de interpretación o directamente defectos en la programación.

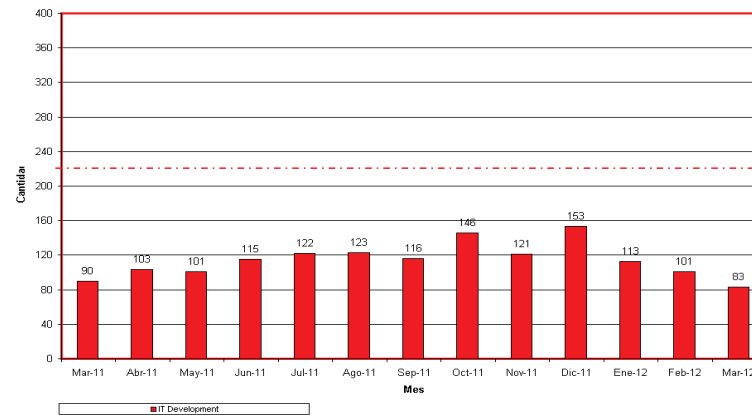
Al haberse implementado consistentemente este método se han podido detectar inclusive algunos incidentes repetitivos que se retrotraen a la implementación de un sistema (en este caso el *core*<sup>20</sup> de Seguros) hace varios años y que no habían sido detectados previamente.

Estos errores de diseño del sistema pudieron ser detectados al haberse podido medir todos los incidentes reportados y luego de identificar un patrón de repetición en algunos de ellos.

<sup>20</sup> Se denomina *core* de un sistema a los principales programas que permiten el funcionamiento del mismo

## GRÁFICO N° 6

### Incidents – IT Software Delivery



Fuente: HSBC

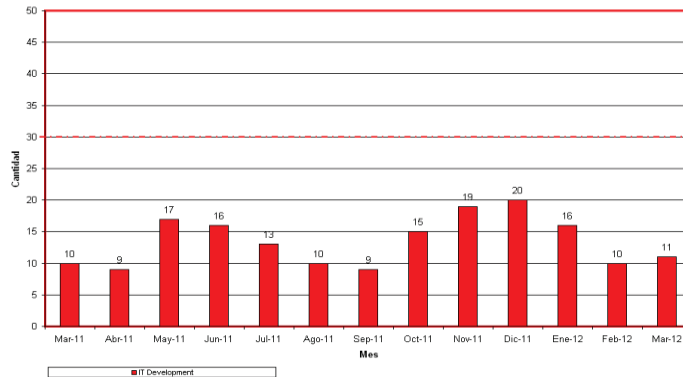
### Claves de Emergencia

Esta herramienta consiste en obtener un usuario especial que permite modificar datos o un programa directamente en producción, para corregir un problema productivo o en base a una solicitud regulatoria; aunque este motivo no se asocia luego con la productividad.

En el Gráfico N° 7 se puede observar la evolución del uso de esta herramienta en el período Marzo 2011 / Marzo 2012.

## GRAFICO N° 7

Emergency Key – IT Software Delivery



Fuente: HSBC

### Implementaciones de Emergencia

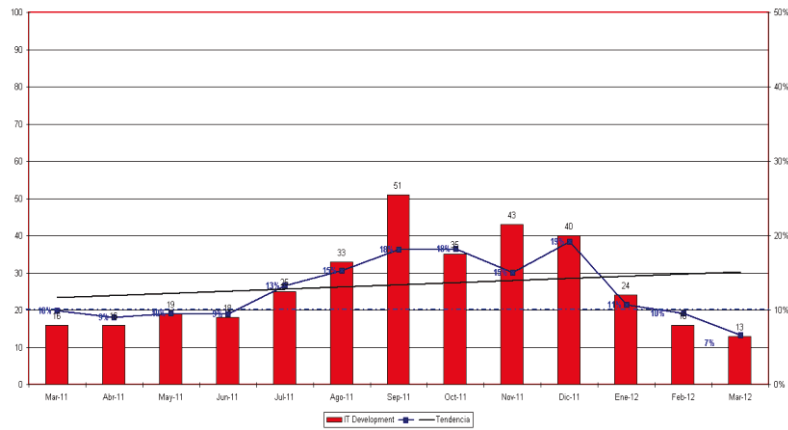
Se producen por alguna necesidad urgente solicitada por el usuario, por un requerimiento regulatorio o por un defecto productivo originado por alguna implementación o error en los programas.

Este último es el motivo al cual se le presta la mayor atención ya que el objetivo es poder referenciar dicho error a la persona que haya codificado el programa en cuestión y de esa manera identificar la causa raíz del mismo.

Se observa en el gráfico N° 8 la cantidad de estas implementaciones realizadas en un período de 13 meses.

## GRAFICO N° 8

### Emergency Implementations – IT Software Delivery



Fuente: HSBC

### Líneas de Código

Ferreya no utiliza este método al considerarse que no brinda parámetros fiables debido a la distinta forma de programar y a la distinta experiencia y habilidades de los integrantes del equipo.

### Puntos de Función

En el área bajo análisis se utiliza esta métrica de la siguiente manera: en una primera instancia se efectúa una estimación en tiempo del trabajo a realizar en base a la complejidad de los programas a ser desarrollados, los cuales se tipificaron de 5 formas:

- Muy simple, menos de 1 día
- Simples, 1 a 2 días
- Medios, 3 días

- Complejos, 5 días
- Muy complejos, 10 o más días.

Esta estimación la realiza el líder de equipo, aunque luego es acordada con el desarrollador (analista de sistemas o analista programador) ya que esta participación mejora el compromiso e involucramiento del programador, a la vez que transparenta la planificación.

Este punto fue uno de los cambios más importantes en la forma de implementar esta métrica ya que anteriormente los desarrolladores no eran consultados acerca de la complejidad de cada programa y por lo tanto no se tenía en cuenta su opinión, siendo justamente ellos los que más conocen el diseño y complejidad de cada sistema a diferencia del líder de equipo, que posee una visión menos detallada. Mediante encuestas de *engagement*, se pudo detectar y corregir este punto.

A continuación se realiza la medición del tiempo efectivamente empleado, el cual se contrasta con la estimación realizada como primer paso. Es decir que si hay que desarrollar una determinada cantidad de programas, se clasifica a cada uno de ellos en base a la complejidad y luego se compara el plazo en el cual el desarrollador entrega cada programa contra la planificación.

Una importante aclaración es que la estimación que se realiza inicialmente está basada en un nivel de conocimientos y experiencia estándar para el puesto. Esta estimación es luego ajustada según el nivel particular de cada programador para poder tener en consideración aquellas particularidades de cada persona. En efecto y por ejemplo, un programa medio que debería desarrollarse en 3 días, si es desarrollado por una persona con una gran capacidad, podrá ajustarse a 2,5 días. Por el contrario si la persona asignada para el desarrollo es un integrante *junior* del equipo, el tiempo podrá ser ajustado a 3,5 o 4 días.

Con referencia al concepto de remuneración variable en base a la productividad obtenida, Ferreyra opina que es una idea a desarrollar, pero no para su aplicación inmediata. Actualmente trata de realizar ajustes de sueldo a aquellas personas con una mayor productividad, pero el componente del sueldo sigue siendo fijo. Para ello existe un

esquema de ranking dentro de cada equipo a efectos de poder mensurar a cada integrante del mismo.

Sin embargo, un problema de esta metodología es lograr el compromiso o *engagement* de los empleados, ya que hay varios de ellos (especialmente los que tienen un desempeño apenas estándar o por debajo de sus pares) que no sienten que se los está midiendo objetivamente. El mensaje que se transmite a los integrantes del equipo siempre es un mensaje claro, aunque esto también trae algún inconveniente, “a veces la verdad no ofende, pero duele”.

Ferreira cree que pese a que su forma de gerenciar trata de ser lo más justa y objetiva posible, las herramientas de medición con las que cuenta no están completamente automatizadas y si existiera alguna herramienta abarcadora de la problemática sería bien recibida.

Por último, menciona que los esquemas de medición se simplifican mucho cuando se trata de medir a un equipo dedicado por tiempo completo a un proyecto en particular ya que en éstos cada etapa está claramente definida y tiene sus propias formas de medición y control. El proceso de medición del equipo es más dificultoso cuando las prioridades del negocio y por lo tanto las tareas asignadas son modificadas de continuo, ya sea por requerimientos cambiantes, nuevas regulaciones o incidentes que deban ser resueltos con urgencia.

En los equipos que comparten el desarrollo con el curso normal de los negocios no existe una separación clara entre un equipo orientado al desarrollo de proyectos y un equipo orientado al mantenimiento diario.

Por lo tanto una posible solución a este tema es realizar una estimación del porcentaje de tiempo dedicado a proyectos y del porcentaje dedicado a mantenimiento. Una vez efectuada esta estimación, se asignan determinadas personas para cada tema (algunos recursos pasarán al desarrollo del proyecto y otros al mantenimiento) a fin de poder clarificar las prioridades al nivel del desarrollador y poder lograr su completo compromiso con la tarea asignada. Lo que se logra de esta forma es una separación fáctica de la

estructura entre recursos asignados al desarrollo de nuevos proyectos y recursos asignados al mantenimiento, aunque esta estructura no es reflejada formalmente en el organigrama. Es tarea del líder de equipo buscar el equilibrio a fin de poder asignar a cada recurso la tarea más conveniente, así como también efectuar una rotación de los recursos asignados a proyectos importantes del negocio ya que estos son los más motivantes y reconocidos.

En conclusión, las mediciones que se efectúan son un camino a la calidad, aunque muchas veces sólo reflejan los síntomas y no las causas de los problemas ya que considera que no existe una herramienta integradora que resuelva la problemática. En parte la utilización del *balanced scorecard* apunta en este sentido.

Actualmente las métricas mencionadas precedentemente son incluidas en dicha herramienta, la cual asimismo contiene una serie de objetivos heredados del *management* global y local de la organización.

## **Entrevista Nro. 2: Fabián Bernard**

El equipo de trabajo a cargo de Bernard mantiene y desarrolla aquellas aplicaciones que brindan soporte a los canales de atención al público de HSBC. Específicamente las terminales de autoservicio, la banca por Internet y el *call center*. También posee responsabilidad sobre las tecnologías de integración entre dichos canales y los sistemas centrales o *core*, denominadas *middleware*.

Bernard comentó que conoce en detalle las métricas utilizadas por el mercado e inclusive trabajó en su aplicación en distintos equipos de trabajo. Su opinión al respecto es que ninguna de dichas métricas es fácil de aplicar por lo que no las utiliza de forma sistemática.

Considera que no existe ninguna métrica de desarrollo que pueda medir la productividad de una forma comparable a como se puede medir la productividad de una fuerza de ventas.

Particularmente comentó que la métrica que mide Líneas de Código tiene muchas falencias, ya que un mal programa puede tener más líneas de código que un buen programa y por lo tanto no sería más productivo. Esta desventaja también está presente en la métrica de Puntos de Función.

Bernard opina que podría investigarse la posibilidad de implementar una métrica basada en el resultado financiero que la programación entrega a la empresa. Sin embargo, este concepto denominado ROI (*Return on Investment*), muy utilizado en el área financiera, podría adolecer de su dependencia a factores exógenos al área de desarrollo. En particular, si se desarrolla un muy buen programa de ventas, pero la fuerza de ventas no es de las mejores del mercado, la rentabilidad obtenida no será un indicativo de la productividad del área de desarrollo.

Debido a las dificultades que se observan para medir la productividad, es tarea del líder de equipo poder agrupar todas las variables y en definitiva evaluar la productividad de cada colaborador. Es fundamental que existan buenos líderes de equipo, que conozcan mucho a los integrantes de sus equipos y que tengan una objetividad que les permita medir equilibradamente a cada persona. También son factores de éxito la comunicación dentro del equipo y el compromiso o *engagement* que se logra.

Pese a que una crítica a la gerencia suele ser que “no se puede gestionar lo que no se puede medir” esto no es aplicable por ejemplo a un estudio de abogados, donde no se mide la productividad de una forma convencional: pueden existir muchos casos menores que demanden un gran trabajo, o determinados casos de fácil resolución que brinden una mayor rentabilidad.

Tampoco existen en el mercado métricas que sean de aplicación práctica; la productividad pasa por la motivación del equipo, la comunicación, la capacidad de reusar diseños, códigos, etc. Un equipo aislado donde cada persona hace lo suyo y se comporta como compartimientos estancos o burbujas, nunca va a lograr una buena productividad. Sin embargo, esta medición tampoco es tangible.

Según Bernard, al ser los desarrolladores evaluados y medidos por una persona, la comunicación es clave ya que los métodos formales existentes son imperfectos pues no engloban todas las variables.

Las mediciones que se basan en tener en cuenta los errores del sistema, también tienen problemas, como ser definir qué se puede considerar un error o defecto, ya que éstos pueden deberse a malas definiciones de los usuarios o a una mala interpretación de dichas definiciones. ¿Eso se considera un error?

En base a lo mencionado precedentemente, Bernard expone que no sería oportuno poner en práctica en la actualidad un esquema de remuneración variable ya que no sería justo ni se podría explicar claramente a los empleados.

Reconoce que existen variables en los lenguajes utilizados y en los entornos de programación; también que las nuevas tecnologías ayudan a los programadores a ser más productivos, pero siguen adoleciendo de una comparación apropiada entre las distintas herramientas utilizadas.

Por su parte, las métricas que miden proyectos de acuerdo a la metodología empleada en base a las estimaciones previas se basan directamente en la experiencia, en proyectos similares desarrollados en el pasado, pero justamente las variaciones que poseen invalidan el método y lo vuelven no determinístico, al entender que tienen un margen de error.

En conclusión, el objetivo debería ser un método transparente, replicable y común a todas las personas de un mismo rol y experiencia, pero este es un tema que hay que seguir trabajando ya que no parece haber una solución generalizadora en el futuro cercano.

### Entrevista Nro. 3: Juan Sorroche

Sorroche mencionó que el HSBC cuenta con un plantel de 21.000 desarrolladores de software situados en todo el mundo lo cual lo hace una de las más grandes del mundo en el sistema financiero y también comparable a empresas líderes en el desarrollo de software como ser Microsoft, Oracle o HP.

Esta estructura se divide en equipos que brindan soporte a negocios locales (como es HSBC Argentina) y también en fábricas de desarrollo (*software factories*) en varios lugares del mundo cuya tarea es el desarrollo de aplicaciones globales que serán posteriormente implementadas en los distintos países donde HSBC tiene presencia.

Sorroche comenta que en los centros mundiales de desarrollo, se aplican diversas técnicas y métricas de medición de la productividad así como también aquellas que están orientadas a la calidad y a la entrega de los proyectos cumplimentando variables de tiempo y costo.

Algunas de las métricas más utilizadas (ver Gráfico N° 9) son las siguientes:

- Productividad: Es medida en base a la cantidad de puntos de función desarrollados por cada individuo.
- Costo de Puntos de Función: se asocia al costo monetario de cada punto de función entregado.
- Densidad de defectos: Es la cantidad de defectos encontrados a partir de las primeras etapas de testeo, comparando dicho número con los puntos de función desarrollados.
- Varianza de cronograma: Mide la diferencia entre el plan original del proyecto (*baseline*) y el plan real.
- Varianza de esfuerzo: Es similar al anterior, pero considerando la variable del costo asociado al proyecto.

**GRAFICO N° 9**
**SWD Summary - Build + Build & Deploy Projects – YTD (Dec'11 – Feb'12)**

Category	Metrics	Measurement Criteria	SWD			Peer Group Figures
			YTD ( Q1 Data )	Baseline	Trend w.r.t Baseline	
Scope (# Projects) -->			29	86	NA	NA
Productivity Metrics	<b>Productivity</b>	Number of function points delivered per FTE	168	156	Improved	288
	<b>Cost per Function Point</b>	Cost incurred per Function Point	\$537	\$612	Improved	\$823
Quality Metrics	<b>Defect Density</b>	Number of defects detected SAT or SIT onwards per thousand Function Points	80	88	Improved	32
Delivery Metrics	<b>Schedule Variance</b>	Difference between the baseline and actual completion schedules	0.0%	0.0%	=	28.0%
	<b>Effort Variance</b>	Difference between the baseline project effort and the actual effort	-1.6%	-1.6%	=	20.0%

Fuente: HSBC

Sorroche, junto con sus pares en otros lugares del mundo, trabaja en determinar la mejor forma de aplicación de estas métricas de acuerdo a las particularidades de cada equipo local. Particularmente en Argentina, se prevé una implementación gradual de cada una de las métricas comenzando a fines de 2012.

Como conclusión, destaca la importancia de su aplicación a efectos de poder medir no sólo la evolución de las variables a través del tiempo, sino también la variabilidad de las mismas en relación a otras empresas de similar envergadura (*benchmarking*), constituyéndose así en una poderosa herramienta de gestión.

#### **Entrevista Nro. 4: Marcelo Farías**

Farías posee 15 años de experiencia en gestión de proyectos, incluyendo entre los equipos que supervisa a integrantes del área de desarrollo así como también otras áreas necesarias para cumplimentar cada uno de ellos como ser marketing, recursos humanos, etc.

Particularmente, en la actualidad se dedica a la gestión de la entrega de servicios y proyectos incluyendo dentro de dichos servicios al mantenimiento de relaciones con clientes existentes y también el desarrollo de nuevos proyectos.

Existe una relación muy estrecha entre su área y las fábricas de desarrollo de Indra, la cual se administra con un diseño de ciclo de vida de software claramente definido por la organización incluyendo todas las fases necesarias para la completitud del proyecto.

La medición de productividad se realiza con herramientas corporativas que soportan el ciclo de vida. Estas herramientas permiten definir ciertos parámetros asociados con cada persona integrante de los equipos (como ser la experiencia, nivel de programación, lenguajes que domina, etc.) y son apropiadas para efectuar una comparación de los distintos equipos no sólo a nivel local sino también con instalaciones de Indra en otras partes del mundo.

Asimismo se ingresan datos relacionados con los requerimientos de negocio, pudiéndose efectuar una catalogación detallada de los mismos (complejidad, tecnología a utilizar, etc.). Una vez ingresados los datos reales de facturación de tiempo e incidencias resueltas por cada desarrollador, la herramienta cruza dicha información con datos de estimación cargados previamente para poder determinar niveles de productividad a nivel de cada desarrollador. Se obtiene un nivel de rendimiento comparable con la media corporativa.

Farías agrega que la metodología de puntos de casos de uso es muy utilizada para realizar las estimaciones iniciales. Esta metodología fue propuesta por Karner<sup>21</sup> en 1993 y se basa

---

<sup>21</sup> Cohn, M (2005), Estimating with Use Case Points, Methods & Tools, 13, 3 pp. 3-13.

en el estudio de los puntos de función para calcular el esfuerzo que demandará desarrollar cada proyecto.

Con respecto a la performance de cada integrante del equipo, la misma es evaluada anualmente basándose en los indicadores entregados por las herramientas de gestión y también por la consideración y el criterio de los jefes de equipo. Esta evaluación no se utiliza para efectuar un pago de una remuneración variable, pero sí para poder determinar variaciones en el salario, potencialidad de crecimiento y asignación a nuevos proyectos.

En conclusión, la gestión del área administrada por Farías se basa en realizar una muy buena presupuestación del esfuerzo, mediante la metodología de casos de uso, y a su vez la utilización de un sistema de registración. De esta forma se puede conocer el desvío y ajustar las variables de eficiencia que los modelos administran para que cada vez se obtengan valores más representativos y ajustados de la realidad.

## CONCLUSIONES

Todo lo expresado hasta aquí adquiere una singular relevancia al dimensionar el universo en el cual estos conceptos deben emplearse. HSBC Argentina es una empresa líder en el mercado financiero y la cantidad de personas que integran el área de desarrollo de software revela en toda su magnitud la necesidad de resolver, por un lado la dificultad de medir la productividad de dotaciones numéricamente importantes, y por el otro, la ineludible necesidad de contar con un instrumento fiable y cierto para lograrlo.

Sin embargo, en base a las observaciones realizadas y a los datos relevados se puede afirmar que las métricas basadas en el tamaño del código o en la complejidad de los programas no son efectivas por sí solas, lo cual dificulta su utilización sistemática. Tampoco la cantidad de defectos encontrados es relevante como única métrica que permita determinar los posibles errores futuros de un módulo o sistema desarrollado.

Se evidencia que las teorías y criterios académicos sobre la productividad de los equipos de desarrollo de software sólo resultan aplicables directamente a la realidad cotidiana mediante la utilización de metodologías y herramientas estandarizadas. Igualmente, al considerarse el desarrollo como una actividad fundamentalmente intelectual se presentan dificultades que obstaculizan una medición precisa de la productividad, por lo que se puede recomendar avanzar con estudios más abarcativos y completos que puedan optimizar las observaciones empíricas comparándolas entre sí y confrontándolas con las teorías aceptadas y elaborar así mejores instrumentos que deberán ser testeados en base a su utilidad en un ambiente productivo.

Es posible y deseable seguir investigando en modelos de mensura para el desarrollo de software y el proceso de testeo que tomen en consideración conceptos cruciales que no están presentes en las aproximaciones tradicionales basadas en estadísticas. La necesidad de lograr herramientas de medición confiables es dada por la exigencia que los niveles directivos demandan, especialmente para asignar recursos.

Específicamente estos modelos deben administrar:

- Procesos diversos y evidencia de los productos entregados por los mismos.
- Relaciones genuinas de causa y efecto, alineando los defectos encontrados con las funcionalidades entregadas en el módulo final.
- Factores de incertidumbre e información incompleta.
- Estimaciones iniciales de esfuerzo y comparación con los resultados obtenidos.

Se observa entonces que se abre un campo de investigación importante relacionado con la optimización de los modelos y métricas actuales, y también una penetración desigual en distintos ámbitos. Por lo tanto un objetivo futuro para HSBC Argentina es poder implementar herramientas de soporte de decisiones que se basen en las métricas simples ya conocidas, pero que combinen los distintos aspectos del desarrollo de software y del testeo correspondiente, considerando las particularidades de los equipos de desarrollo en Argentina.

Esas herramientas permitirán a los gerentes de HSBC no solamente medir la productividad de sus equipos sino también realizar distintos tipos de predicciones, evaluaciones y compromisos con todos los interesados. El objetivo último es poder administrar aquellos factores que usualmente no contemplan los métodos tradicionales, específicamente de los que derivan evidencias no certeras y con datos incompletos o subjetivos.

## BIBLIOGRAFÍA

- Albrecht, A (1979). Measuring Application Development Productivity, Proceedings of the Joint Share, Guide, and IBM Application Development Symposium, Monterey, California, 10, 14–17
- Alcatel (2006). *Measuring and Improving Productivity*. Zurich: Ebert, C.
- Asproni, G. (2004). Motivation, Teamwork and Agile Development, *Agile Times*, 2004, 4, 1-9
- Atchison, T, Belcher, D & Thomsen, D (1987). *Compensation Administration*, Prentice Hall, NY
- Balling, M, Gnan, E, Lierman F et al (2009), Productivity in the Financial Services Sector, Suerf, Vienna.
- BCRA (2012). *Información de Entidades Financieras*. Buenos Aires
- Beatty, R, Arnett, K & Liu, C (2005). CIO/CTO Job Roles: An Emerging Organizational Model, *Communications of the IIMA*, 2005, 5, 2.
- Benediktsson, O., Dalcher, D. & Thorbergsson, H. (2006). Comparison of software development life cycles: a multiproject experiment, *IEE Proceedings*, 153 3, 87-101
- Bloom, N, Dorgan, S, Dowdy, J et al (2007). Management Practice & Productivity: Why they matter, *Economic and Social Research Council*, November 2007, 1-10.
- Boehm, B. (1981). *Software Engineering Economics*, Prentice Hall, New Jersey.
- Boehm, B, Abts, C, Devnani-Chulani, S et al (1998) *COCOMO II Model Definition Manual*, Center for Software Engineering, USC
- Coelli, T, Rao, D & Baltese, G (1998). *An Introduction to Efficiency and Productivity Analysis*, Kluwer Academic Publishers, Norwell, Ma.
- Cohn, M (2005), Estimating with Use Case Points, *Methods & Tools*, 13, 3.
- Dulye, L, Grossman, D, Quinlan, T et al. (2007). *Essential Techniques for Employee Engagement*, Melcrum Publishing Limited.
- Ebert, C., Dumke, R et al (2004). *Best Practices in Software Measurement*, Springer. New York, NY.

- Fenton, N. & Neil, M. (1998). Software Metrics: Successes, Failures and New Directions, *Journal of Systems and Software*, July/August 2000,1-19
- Fenton N. & Pfleeger S. (1998). Software Metrics: A Rigorous & Practical Approach (2nd Edition), PWS Publishing Co. Boston, MA.
- Fowler, M. (2003). Cannot Measure Productivity, Extraído el 17 de Noviembre de 2011 de <http://martinfowler.com/bliki/index.html>
- Fried, H., Knox Novell, C. & Schmidt, S. (2008). *The Measurement of Productive Efficiency and Productivity Growth*, Oxford University Press.
- Hesselbein, F. & Goldsmith, M. (2006). *The Leader of the Future 2*, Jossey Bass. San Francisco, CA
- Jayamaha, A. & Mula, J. (2011). Productivity and Efficiency Measurement Techniques, 2<sup>nd</sup> International Conference on Business Information, Kelaniya, Sri Lanka.
- Jones, C. (2009). Software Engineering Best Practices, McGraw-Hill.
- Jones, C. & Bonsignour, O. (2011), The Economics of Software Quality, Addison-Wesley Longman
- Kitchenham, B. & Mendes, E. (2004), Software Productivity Measurement Using Multiple Size Measures, *IEEE Transactions of Software Engineering*, 30, 12, 1023-1035
- Klussman, W. (2009), Philosophy of Leadership – *Driving employee engagement in integrated management systems*, Diplomita Verlag, Hamburg.
- LaFasto, Frank M. J.; Larson, Carl (August 2001), When Teams Work Best, Sage Publications Inc., Thousand Oaks, CA.
- Lee, G. & Xia, W. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility, *MIS Quarterly*, 34 1, 87-114
- Mc Connell, S. (2008). Measuring Productivity of Individual Programmers, Extraído el 17 de Noviembre de 2011 de <http://forums.construx.com/blogs/>
- McConnell, S. (1997). Gauging Software Readiness with Defect Tracking, *IEEE Software*, 14, 3

- Mills, E. (1988). *Software Metrics*, Software Engineering Institute, Seattle University.
- Overby, S. (2007). How to Turn Your Employees Into Leaders, Extraído el 21 de Diciembre de 2011 de <http://www.cio.com>
- Poile, C., Nachiappan Nagappan, A. & Layman, L. (2009). Coordination in Large-Scale Software Development: Helpful and Unhelpful Behaviors, *Microsoft Research*, 135, 1-17
- Reel, John (1999). Critical Success Factors in Software Projects. *IEEE Software*, 3, 18-23.
- Rising, L. & Janoff, N. (2000). The Scrum Software Development Process for Small Teams. *IEEE Software*, 4, 26-32.

## OTRAS FUENTES DE INFORMACIÓN

- Banco Central de la República Argentina - <http://www.bcra.gov.ar/>
- CEPA – The Center for Efficiency and Productivity Analysis - <http://www.uq.edu.au/economics/cepa>
- International Function Point Users Group - <http://www.ifpug.org/>
- International Organization for Standardization - <http://www.iso.org/>
- Project Management Institute - <http://www.pmi.org/>
- Real Academia Española - <http://www.rae.es/>
- Survey Monkey - <http://es.surveymonkey.com/home/>
- WIKIPEDIA - <http://www.wikipedia.com/>

## GLOSARIO

### **Productividad:**

Es la relación que existe entre la salida (*output*) de un proceso y su entrada (*input*) considerando factores relevantes como ser complejidad, restricciones de tiempo y presupuesto, distribución de los equipos, etc. Particularmente se consideran entradas a los recursos utilizados (recursos humanos, tiempo, etc.) y salidas al valor entregado (producto de software, documentación).

Algunas definiciones complementarias son las siguientes:

- OCDE (Organización para la Cooperación y Desarrollo Económico): Productividad es igual a producción dividida por cada uno de sus elementos de producción.
- OIT (Organización Internacional del Trabajo): Los productos son fabricados como resultados de la integración de cuatro elementos principales: tierra, capital, trabajo y organización. La relación de estos elementos a la producción es una medida de la productividad.
- EPA (Agencia Europea de Productividad): Productividad es el grado de utilización efectiva de cada elemento de producción. Es sobre todo una actitud mental. Busca la constante mejora de lo que existe ya. Está basada sobre la convicción de que uno puede hacer las cosas mejor hoy que ayer, y mejor mañana que hoy. Requiere esfuerzos continuados para adaptar las actividades económicas a las condiciones cambiantes y aplicar nuevas técnicas y métodos. Es la firme creencia del progreso humano.

### ***Engagement:***

Es un término cada vez más utilizado a nivel empresarial en relación a los empleados y se puede traducir como ‘compromiso’. Se refiere a la voluntad de los empleados de la organización en brindar lo máximo de sí mismo para lograr objetivos comunes con el

empleador sintiéndose parte de las metas de negocio y planes estratégicos. Dulye<sup>22</sup> lo considera esencial para las comunicaciones internas a fin de lograr un compromiso de los empleados con las empresas donde trabajan y lo relaciona directamente con el liderazgo individual que cada persona posee y pone en práctica en su trabajo.

### **Métricas:**

Se refiere a una forma de medición del producto desarrollado (software) así como también al proceso por el cual se lo desarrolla. El producto de software debe ser considerado como una entidad abstracta que evoluciona desde un estado inicial de una necesidad (expresada a través de una definición) hacia el estado final de una aplicación o sistema instalado en un entorno informático y considerado productivo para el solicitante.

### **Líneas de Código:**

Son las instrucciones que forman el código fuente de los programas las cuales se escriben utilizando la sintaxis determinada por el lenguaje de programación utilizado. Estas líneas deben respetar la secuencia necesaria para mantener la lógica del programa, y son las que, una vez finalizada la programación, se ejecutan para lograr el funcionamiento en producción de los programas. En la actualidad existen algunos lenguajes de programación (llamados meta-lenguajes) que pueden generar automáticamente líneas de código, mediante el ingreso de instrucciones de alto nivel por parte del programador.

---

<sup>22</sup> Dulye, L, Grossman, D, Quinlan, T et al. (2007). Essential Techniques for Employee Engagement, Melcrum Publishing Limited, pp. 6-7

### *Balanced Scorecard*

Es un marco de medición de desempeño de alto nivel donde es posible incluir e integrar métricas específicas, y en base al cumplimiento o no de determinados objetivos prefijados, poder efectuar una medición y calificación de cada individuo.

El método permite una especificación de objetivos orientados en cuatro áreas:

- Procesos Internos
- Clientes
- Desarrollo y Aprendizaje
- Finanzas

Cada una de estas áreas recibe una ponderación (hasta totalizar el 100%) y en cada una de ellas se asignan objetivos claros y mensurables.

## ANEXO 1 - Encuesta a Desarrolladores de Software

Preguntas utilizadas y respuestas obtenidas.

Casos: 96

<b>1. ¿Podría indicar su edad?</b>	<b>Respuestas</b>	<b>Porcentaje</b>
20 a 30 años	10	10,4%
31 a 40 años	<b>37</b>	<b>38,5%</b>
41 a 50 años	35	36,5%
51 años o más	14	14,6%

<b>2. ¿Cuál es el grado máximo de estudios que posee?</b>	<b>Respuestas</b>	<b>Porcentaje</b>
Posgrado completado	2	2,1%
Posgrado en curso	5	5,2%
Universitario/Terciario completado	<b>62</b>	<b>64,6%</b>
Universitario/Terciario en curso	21	21,9%
Estudios técnicos	3	3,1%
Secundario	3	3,1%

<b>3. ¿Cuántos años posee de experiencia laboral en el área de Desarrollo de Sistemas?</b>	<b>Respuestas</b>	<b>Porcentaje</b>
Hasta 5 años	5	5,2%
6 a 10 años	13	13,5%
10 a 15 años	21	21,9%
16 a 20 años	26	27,1%
21 años o más	<b>31</b>	<b>32,3%</b>

<b>4. ¿Cuál de los siguientes roles es el que más se aplica a su trabajo actual?</b>	<b>Respuestas</b>	<b>Porcentaje</b>
Arquitecto/Diseñador	2	2,1%
Analista de Sistemas	<b>45</b>	<b>46,9%</b>
Analista Programador	42	43,8%
Tester	3	3,1%
Otro	4	4,2%

**5. ¿Cuáles considera Ud. que son las variables más relevantes a la hora de formar parte de un equipo de desarrollo y que por lo tanto más inciden en su productividad?**

	Extremadame nte relevante	Relevante	No demasiado relevante	Totalmente irrelevante
La capacidad individual	21,9% (21)	<b>74,0% (71)</b>	4,2% (4)	0,0% (0)
La comunicación entre los miembros del equipo	<b>67,7% (65)</b>	32,3% (31)	0,0% (0)	0,0% (0)
La complejidad del producto a desarrollar	10,4% (10)	<b>55,2% (53)</b>	32,3% (31)	2,1% (2)
El empleo de una metodología	38,5% (37)	<b>50,0% (48)</b>	10,4% (10)	1,0% (1)
Conocer el tiempo disponible para cada tarea	21,9% (21)	<b>66,7% (64)</b>	10,4% (10)	1,0% (1)
La existencia de facilidades y recursos externos que lo ayuden a desarrollar su trabajo	11,5% (11)	<b>62,5% (60)</b>	24,0% (23)	2,0% (2)
El liderazgo que pudiera ejercer el líder de equipo o el gerente de desarrollo	<b>52,1% (50)</b>	44,8% (43)	3,1% (3)	0,0% (0)

**6. ¿Considera que su trabajo debe ser medido en base a la productividad del mismo?**

	Respuestas	Porcentaje
Totalmente de acuerdo	34	35,4%
Parcialmente de acuerdo	<b>57</b>	<b>59,4%</b>
Parcialmente en desacuerdo	3	3,1%
Totalmente en desacuerdo	2	2,1%

**7. ¿Considera que existen en el mercado las herramientas apropiadas para efectuar la medición de su trabajo y por lo tanto determinar su productividad?**

	Respuestas	Porcentaje
Totalmente de acuerdo	11	11,5%
Parcialmente de acuerdo	<b>61</b>	<b>63,5%</b>
Parcialmente en desacuerdo	18	18,8%
Totalmente en desacuerdo	6	6,3%

<b>8. Por favor mencione su grado de conocimiento de las siguientes herramientas de medición de la productividad</b>	<b>Excelente (Lo conozco en detalle)</b>	<b>Bueno (He oído hablar del mismo, pero no lo conozco en detalle)</b>	<b>Regular (No lo conozco)</b>
Lineas de Código (Cantidad de líneas producidas por cada desarrollador)	35,4% (34)	<b>39,6% (38)</b>	25,0% (24)
Puntos de Funcion (Medición de la complejidad de la funcionalidad de negocios en cada programa. Se miden inputs, outputs, algoritmos, interfaces, etc.)	26,0% (25)	<b>41,7% (40)</b>	32,3% (31)
Cantidad de errores en los programas y repetición de los mismos (Medición de fallas detectadas tanto en testing como en producción, también la no reincidencia de errores)	38,5% (37)	<b>45,8% (44)</b>	15,6% (15)

<b>9. ¿Considera que en su trabajo actual se están utilizando las herramientas apropiadas para efectuar la medición de su trabajo y poder medir de esa forma su productividad?</b>	<b>Respuestas</b>	<b>Porcentaje</b>
Totalmente de acuerdo (Se utilizan las herramientas adecuadas)	5	5,2%
Parcialmente de acuerdo (Se utilizan algunas herramientas, pero no de la mejor forma)	<b>48</b>	<b>50,0%</b>
En desacuerdo (No se utilizan herramientas, la medición es subjetiva)	43	44,8%

<b>10. Si Ud. trabajara en un lugar en el que se estén utilizando las herramientas apropiadas para efectuar la medición de su trabajo y poder medir de esa forma su productividad, ¿aceptaría que su sueldo sea variable en base a dicha productividad?</b>	<b>Respuestas</b>	<b>Porcentaje</b>
SI - ¿Por qué?	44	45,8%
NO - ¿Por qué?	<b>52</b>	<b>54,2%</b>

### **Pregunta 10 – Citas representativas (Respuesta: ‘SI’)**

‘Sí, lo aceptaría, pero con un sueldo básico porque me siento seguro y responsable efectuando mis tareas por lo cual el sueldo variable no afectaría mi actual sueldo fijo o bien podría mejorarlo.’

‘Sí, aunque es un tema complejo, debe estar perfectamente alineada la herramienta de medición para evitar diferencias en el equipo de trabajo, ya sea en cantidad de trabajo, calidad, continuidad.’

‘Sí, sólo si se consideran otros factores para el cálculo del sueldo además de la productividad. Existen acciones que crean valor económico para una compañía y pueden no ser detectadas por una herramienta de medición de productividad. En consecuencia, la herramienta de medición deberá ser lo suficientemente inteligente como para entender situaciones que puedan representar un aporte de valor a la compañía, por ejemplo la gestación de una idea, la capacitación de un colaborador, la capacitación individual. La productividad individual en un período de tiempo no es un dato suficiente para determinar el aporte de una persona a una organización, inclusive en el área de desarrollo de software. Por lo tanto, el sueldo debe tener un componente ligado a la productividad directa en desarrollo de software y otros relacionados con innovación, liderazgo, trabajo en equipo, etc. Si la herramienta puede entender la productividad que generan estas acciones, entonces sí se podría variabilizar una porción del sueldo asociándolo a una herramienta de medición.’

‘Sí- de esta manera, permitiría diferenciar a cada persona y recompensarla según su esfuerzo’

‘Es la forma más apropiada de lograr un producto final de mayor calidad’

‘Si las definiciones fueran precisas y el error fuese propio, sí’

**Pregunta 10 – Citas representativas (Respuesta: ‘NO’)**

‘Una persona debe tener un sueldo básico constante. Es una medida injusta para aquellos que, por alguna razón, no cuentan con las herramientas/conocimientos necesarios. Los parámetros utilizados, a mi criterio, tienen un sesgo importante. ¿Como se subsana que una persona haga muchas líneas de código de más o realice diseños intrincados para así aumentar la envergadura y complejidad del requerimiento? No son variables absolutas y libres del falseamiento por parte de los individuos.’

‘Porque la productividad es sólo una de las variables a tener en cuenta.’

‘Cualquier herramienta, incluso la más precisa, debería estar acompañada de una evaluación de la parte humana que afecta la productividad. Esto quedaría a criterio de quien evalúa por lo que sería demasiado subjetivo como para permitir que impacte en los sueldos. Creo más adecuados los incentivos por bonus puntuales.’

‘La productividad depende de muchos factores, a veces poco controlables’

‘No – Hasta tanto no conocer cuáles serían las herramientas utilizadas para la medición’

‘El desarrollo de sistemas incluye tareas de análisis y diseño que no son mensurables’