Medición de la Productividad de los Puestos de Trabajo en Ingeniería del Software

Adrián Hernández-López, Ricardo Colomo Palacios, Ángel García Crespo

Dpto. Informática

Universidad Carlos III de Madrid

Madrid - España

adrianhernandezlopez@gmail.com; {ricardo.colomo, angel.garcia}@uc3m.es

Abstract: *Productivity is a key element in organizational management. Although it can be measured at different levels (country, sector, organization...) this article focuses on productivity at job position level. Additionally, the area of activity of these jobs is Software Engineering (SE), which is a part of computer engineering, and focuses on the development and maintenance of software. The aim of this paper is to provide a vision of the state of the art about productivity measurement, inputs and outputs of the production process used for this measurement at job position level in SE. In addition, this article raises points that should be addressed to develop new productivity measures, along with some general difficulties in carrying out that task.*

Resumen: *La productividad es un elemento clave en la gestión organizacional. Aunque puede ser medida a diferentes niveles (país, sector, organización...) este artículo se centra en la productividad a nivel de puesto de trabajo. Además, el área de actividad de estos puestos de trabajo es la Ingeniería del Software (IS) que se enmarca dentro de la ingeniería informática, y se centra en el desarrollo y mantenimiento de software. El objetivo de este artículo es disponer de una visión del estado de la cuestión sobre la medición de la productividad, y de las entradas y salidas del proceso productivo utilizadas para dicha medición a nivel de puesto de trabajo en IS. Además, se plantean los puntos que deben ser tratados para elaborar nuevas medidas de productividad, junto con las dificultades generales para llevar a cabo dicha tarea.*

Keywords: *Software Engineering, Productivity, Measurement, Job*

1. Introducción

La gestión de la productividad continúa siendo un reto dentro de la gestión de proyectos de las Tecnologías de la Información y Comunicación (TIC). Mientras que en la industria de fabricación se han diseñado y probado métodos para determinar la productividad, la industria TIC está un paso por detrás en términos de disponer de métodos para evaluar las salidas y predecir el esfuerzo necesario para completar los proyectos [1]. La dificultad de la gestión de la productividad en este sector se debe principalmente a que el factor clave del mismo son las actividades con gran necesidad de capital humano [2]. Dentro de la industria TIC, la Ingeniería del Software (IS) tiene como principal actividad el desarrollo y mantenimiento de aplicaciones software.

Las medidas de productividad en los proyectos de software están principalmente basadas en ratios entre el tamaño de software entregado y el esfuerzo realizado para obtenerlo [3]. Siguiendo esta orientación, las dos formas más utilizadas de medir el tamaño de software entregado: los Puntos Función (FP, *Function Points*) y las líneas de código (SLOC, *Source Lines of Code*), pese a ser muy utilizadas, son medidas de dudosa fiabilidad ya que el desarrollo general puede ser improductivo incluso si la productividad a nivel de desarrollo crece [4]. Además de estas formas de medir la salida producida, hay otras propuestas recientes tales como la medición basada en puntos función post mórtem [5], la utilización de medidas de tamaño múltiple [6], y métodos específicos para metodologías específicas tales como la Orientación a Objetos [7]. De forma adicional hay que destacar que las medidas basadas en SLOC sólo representan la productividad de parte de las actividades de programación, mientras que otras actividades como la gestión, el análisis, y el diseño se quedan fuera del alcance de estas medidas. De igual forma, las medidas basadas en FP sólo representan la funcionalidad ofrecida por el software desarrollado pero no tiene en cuenta la parte no funcional del desarrollo, ni otros elementos tales como la reutilización o la accesibilidad.

Los factores que afectan a la productividad TIC están aceptados y reconocidos por la mayoría de las organizaciones: el incremento de las restricciones de almacenamiento, las limitaciones de tiempo, los requisitos de fiabilidad, los lenguajes de alto nivel, el tamaño del equipo de desarrollo, la volatilidad de los requisitos, las habilidades con herramientas de personal, la disponibilidad de personal, la participación de los clientes, y la duración del proyecto (ver por ejemplo [8]). A pesar de su aceptación y reconocimiento, la influencia de algunos de estos factores, tanto de forma positiva, como negativa, en la productividad TIC no es clara y puede variar en función de otros factores externos tales como el sector de negocio [8] y el ratio de *outsourcing* [9].

Finalmente, la medición de la productividad está relacionada con las prácticas de gestión, y más en concreto con la cultura organizacional y su estructura. En organizaciones en las cuales la cultura de informar y comparar está implantada, las medidas de productividad pueden ser consideradas como un posible indicador útil para medir el efecto de cualquier mejora en la productividad. Por un lado, en organizaciones cuya gestión esté basada en estilo de mando directivo, donde las ordenes vienen de arriba hacia abajo, las medidas de productividad serán aceptadas como una forma de comunicar a los niveles superiores cómo de bien se están ejecutando los proyectos. Por otro lado, en organizaciones con una cultura de innovación y creatividad, las medidas de productividad podrían ser vistas como una herramienta de control que no sirve para la consecución de los objetivos. Adicionalmente, la unión entre los objetivos organizacionales y la productividad podría no estar clara en estas organizaciones, y esto puede llevar a una falta de comprensión sobre el qué, el cómo y porqué es importante evaluar, y por tanto, medir la productividad.

En este escenario, la definición de nuevos modelos, medidas y métodos de evaluación pueden dilucidar la productividad en los proyectos TIC, especialmente en los proyectos de desarrollo de software. A su vez, las organizaciones necesitan establecer sus propias medidas de productividad, además de las medidas del sector, para realizar un *benchmarking* (evaluación comparativa) de sus datos [8]. Antes de definir nuevos modelos, medidas y métodos para evaluar la productividad en los proyectos de desarrollo de software, es necesario disponer del estado de la cuestión para tener una visión de la situación actual. Así pues, el presente artículo presenta el estado de la cuestión sobre la medición de la productividad a nivel de puesto de trabajo en la IS.

El resto de este artículo se organiza de la siguiente manera: en primer lugar, se presenta un resumen de la productividad en IS desde el punto de vista de su medición; en segundo lugar, se introducen los elementos necesarios para medir la productividad; en tercer lugar, se comenta la definición de puesto de trabajo en IS; en cuarto lugar, se resume el estado actual de la medición de la productividad a nivel de puesto de trabajo en IS; finalmente, se presentan las conclusiones y algunas de las líneas de investigación posibles en esta área de conocimiento.

2. Productividad en Ingeniería del Software

Los orígenes del término “productividad” se remontan al siglo XVIII [10]; sin embargo, hasta mediados del siglo pasado, las definiciones eran confusas. Tradicionalmente, la productividad ha sido definida como el ratio de salidas producidas por unidad de entrada; es decir, como sinónimo de la eficiencia en el uso de la entrada para producir unas salidas [11]. Esta definición encaja bien en el paradigma de la fabricación porque está basado en cantidades de unidades de medida estandarizadas y claramente identificadas, pero no encaja en nuevos

entornos tales como la IS, en los cuales hay activos intangibles, junto con tangibles. Es por ello que en las industrias de servicios, la productividad debe ser vista como un componente del rendimiento, no como un sinónimo de éste [12]. Por otro lado, la productividad tiene por objeto ser comparada con anteriores medidas de productividad, y no como una unidad de resultado, es decir, las medidas de productividad deben ser utilizadas para compararlas sobre el tiempo, mientras que el rendimiento representa una medida temporal. Siguiendo esta puntualización, el valor de las medidas de productividad se encuentra en la capacidad de gestionar y controlar para alcanzar un uso más eficiente de los recursos [13]. Además el objetivo principal de la medición de la productividad es la mejora de la productividad [14]. Tal y como Anselmo y Ledgard indicaron [15] siguiendo la afirmación de Lord Kelvin - *"When you can measure what you are speaking about, ... you know something about it; but when you cannot measure it, ... your knowledge is of a meager and unsatisfactory kind..."*. Por ello, la mejora de la productividad en IS no puede ser realizada sin una medición de la productividad. De este modo, una medida de productividad apropiada aporta una herramienta de pronóstico adecuada para lograr esa mejora de la productividad [14].

En IS, el interés científico por la medición de la productividad puede decirse que comenzó a finales de los años 80. En sus orígenes, los objetos de estudio fueron principalmente las actividades de programación [16] y los proyectos de software [17]. Además, en esa época se empezaron a considerar los factores que afectan a la productividad [18-19]. Con el paso del tiempo, y dado que la medición de la productividad tiene como uno de sus objetivos comparar productividades, fueron apareciendo investigaciones que realizaban comparaciones a diferentes niveles [20-21]. De igual forma, el estudio de los factores continuó en las siguientes décadas [22]. Por otro lado, se fueron incluyendo elementos a la medición de la productividad tales como la reutilización [23-24] o la orientación a objetos [25]. No obstante, la utilización de estos elementos no parece haber sido aceptada ampliamente ya que las medidas de productividad continúan utilizando los estudios de finales de los años 80 como marco de trabajo [26-28]. La finalidad que persiguen estas medidas es medir la eficiencia con la que se termina un proyecto software, bien mediante la medición de las líneas de código desarrolladas (SLO) o de la funcionalidad entregada (FP). Esta utilización además permite emplear la medición para estimar futuros proyectos software utilizando alguno de los métodos disponibles [29]. Así pues, y dado que uno de los retos clásicos de la industria del software es la entrega en tiempo de los proyectos [30], la utilización de medidas de productividad que tengan por objeto medir la eficiencia de entrega no ha de resultar extraño [31]. Sin embargo, estas medidas de productividad no miden todas las actividades de IS, y, por lo tanto, no miden la productividad a nivel de puesto de trabajo. De este modo, es necesario elaborar medidas de productividad para dicho nivel.

Antes de cualquier intento de medir la productividad, es necesario identificar qué debe ser capturado en dicha medida [32]. Así pues, teniendo en cuenta estas contribuciones, y con el objetivo de crear una medida de productividad en IS, es necesario distinguir los factores, entradas y salidas susceptibles de ser medidas. En primer lugar, los factores que afectan a la productividad son de diversa tipología y pueden organizarse en: factores técnicos (producto, proceso, entorno de desarrollo) y factores *soft* (cultura organizacional, cultura de equipo, capacidades y experiencias, entorno, proyecto) [33]. Aunque muchos de estos factores son ampliamente conocidos desde hace años - por ejemplo, la influencia de la experiencia de los trabajadores [34-35] y el lenguaje de programación (entorno de desarrollo) - no está claro que su importancia realmente sea la que hace años se le adjudicó ya que las prácticas y procesos de trabajo así como las herramientas han

evolucionado considerablemente [36]. Además, evaluar el grado de influencia y la valencia del factor es difícil dado que no siempre será la misma y dependerá en gran medida de las características del proyecto y del entorno de trabajo [37]. Lo que sí parece estar claro es la existencia de dos grandes grupos de factores: “factores de personas” (*people factors*) y “factores tecnológicos” (*technical factors*).

En segundo lugar, las entradas del proceso productivo susceptibles de ser medidas utilizadas hasta ahora se centran en unidades de tiempo, principalmente horas y días [38]. La utilización del tiempo, como única entrada, se puede deber a dos intereses: por un lado, la entrega de un proyecto en el tiempo planificado, así como la puesta en el mercado del mismo, elementos que son un factor importante para el éxito de un proyecto; y, por otro lado, el coste o esfuerzo de personal es el mayor coste en las organizaciones productoras de software por lo que el tiempo que se dedica a desarrollar software es el principal coste [39]. Ambos intereses en emplear esta entrada son de índole económica, sin embargo la productividad no es sólo un indicador económico. Además, el empleo de esta entrada plantea algunas cuestiones sin resolver: ¿realmente sólo se necesita tiempo para producir software?, y si sólo se mide el tiempo ¿qué tiempo debería medirse: el empleado por el trabajador para llevar a cabo sus tareas o el pagado por el empleador? De forma paralela y con el avance del tiempo, el coste de los equipos hardware se ha ido reduciendo y el coste de estos recursos es casi irrelevante en proyectos de gran tamaño. Así pues, cualquier recurso de interés para la medición de la productividad no estará relacionado con este recurso sino que estará directa o indirectamente relacionado con la mano de obra empleada. El tiempo es, sin lugar a duda, una de ellas ya que a mayor tiempo mayor empleo de mano de obra, sin embargo no parece ser la única entrada cuando todavía se piensa en la IS como una actividad artesanal [30].

En tercer lugar, las salidas del proceso productivo han seguido la misma filosofía que las entradas de entrega en tiempo por lo que se han medido principalmente en SLOC y/o funcionalidad [40]. Estas medidas de la salida, pese a sus limitaciones, tienen gran aceptación en las organizaciones. Por un lado, los FP son una forma de medir la funcionalidad a desarrollar y, por lo tanto, una forma de medir qué se entrega al cliente midiendo la funcionalidad como salida. Por otro lado, las SLOC son las instrucciones que codifican la funcionalidad desarrollada, de modo que son una medida de bajo nivel por lo que es una medida más tangible. No obstante, estos dos tipos de salida no son los únicos que se producen en la IS. Por ejemplo, una salida importante, que queda fuera de las salidas habitualmente consideradas, es la calidad, la cual afecta a la salida producida (eficacia) y al proceso productivo en sí mismo (eficiencia) [41]. Pero hay otras salidas que se producen, entre ellas existen salidas de tipo organización del trabajo o de tipo, que no siempre son medidas cuando se habla de productividad. Además, hay características de las salidas que pueden afectar a la productividad, tanto por su creación inicial como por su posterior uso, entre ellas destacan la reutilización [15] y la documentación [42].

De forma adicional, el nivel de análisis debe ser tenido en cuenta cuando se mide la productividad en IS ya que no todas las medidas son útiles en todos los niveles. Así pues, debe haber una clara especificación del objetivo de la medida y las audiencias interesadas en los resultados de dichas medidas; esta información debe estar definida antes que cualquier otra información sobre productividad [12]. A nivel de sector y organización, la granularidad de los factores, entradas y salidas envueltas no es suficiente para una medida precisa que pueda ser utilizada como fuente de información en la mejora de ésta; de modo que las medidas a este nivel constituyen sólo un elemento para comparar entre periodos. A niveles más bajos de medición es necesario un amplio análisis de los elementos que deben ser medidos, lo que requiere un proceso establecido (ver por

ejemplo [12]). Llegado este punto, es necesario preguntarse si las prácticas actuales de medición de productividad en IS siguen este tipo de procesos; y, si los factores, las entradas y las salidas están claramente identificados en cada uno de los niveles de medición. Y por otro lado, si factores ampliamente aceptados como la reutilización o la calidad están incluidos (de forma directa o indirecta) en las medidas de productividad empleadas.

3. Elementos necesarios para la medición de la productividad: factores, entradas y salidas

Antes de elaborar una medida de productividad es necesario identificar qué va a ser medido y qué influye en lo que va a ser medido. En concreto, es necesario identificar las entradas y salidas (qué va a ser medido) y los factores (qué influye). Es conocido que en IS, la calidad de las salidas y las entradas del proceso productivo pueden afectar la productividad, por lo tanto, la calidad y la productividad deben ser medidas como conceptos interrelacionados [43]. Tal y como Gummesson [44] observó, la calidad, la productividad y la rentabilidad forman una triplete en la que todas las partes están relacionadas con el mismo fenómeno: el resultado económico de la organización. La calidad de los productos puede ser significativa y positivamente afectada por la capacidad del personal, los factores del proceso de desarrollo software, y los recursos de despliegue en las etapas iniciales de desarrollo del producto, especialmente en el diseño [45].

La industria IS en gran medida es un sistema abierto en el cual las partes interesadas, los clientes y los usuarios finales afectan a las entradas y salidas, producen una contribución a la eficiencia interna y externa, y por lo tanto a la productividad del proceso productivo (ver por ejemplo [46]). Por consiguiente, es necesario un enfoque totalmente diferente de la productividad para obtener una medida global que establezca como de bien utilizan las organizaciones IS los recursos para crear salidas con la calidad percibida necesaria y el valor para el cliente requerido [43]. Por lo tanto, las medidas de las entradas y las salidas deben tener en cuenta la cantidad y calidad. Esta importancia se refleja en la premisa que Grönroos y Ojasalo [43] establecieron: “Cuanto mejor es la calidad percibida producida utilizando una cantidad determinada de entradas (entradas de los proveedores de servicios y de los clientes), mejor es la eficiencia externa, dando por resultado una mejora de la productividad del servicio” y “Cuanto más eficientemente la organización de servicios utiliza sus propios recursos como entrada en los procesos y cuanto más la organización puede educar y guiar a los clientes hacia entradas de procesos de apoyo para producir una cantidad determinada de salidas, mejor es la [eficiencia] interna”.

3.1. Factores

Los factores que afectan en la medición de la productividad han sido ampliamente estudiados. Por ejemplo, Anselmo y Ledgard [15] indican algunos de estos factores: independencia de los módulos, comprensibilidad del código y la arquitectura, flexibilidad del proceso de desarrollo software, visibilidad de la arquitectura, y la abstracción del proceso de desarrollo software para ser examinado experimentalmente. Además, la importancia de la arquitectura y el diseño en la productividad del desarrollo de software ha sido contrastada [15]. Éstas son propiedades inherentes al entorno de desarrollo de software, y pueden aumentar o disminuir con el diseño. De igual forma, otros factores afectan a la productividad: restricciones de recursos, volatilidad de los requisitos, utilización de herramientas software, complejidad del programa, lenguajes de programación,

involucración del cliente y el usuario, experiencia y capacidades del personal, estabilidad del personal, tamaño del equipo, ratio de *outsourcing*... [8, 47]. Por otro lado, Scacchi [22] dividió estos factores en tres listas de atributos: entorno de desarrollo software, producto del sistema software, y atributos del personal del proyecto. Además, existen revisiones sistemáticas de la literatura sobre este objeto de estudio [33, 36].

Desde el punto de vista de los autores del presente artículo, los factores de personal requieren una especial atención (ver por ejemplo [48]). Las actividades de IS son intensivas en capital humano, por lo que el factor humano tiene que ser analizado en cualquier práctica de gestión con el objetivo de obtener resultados más ajustados a la realidad. En el contexto de la medición de la productividad, es conocido que factores relacionados con el personal tales como las capacidades técnicas y no técnicas, las habilidades, y la experiencia (en diversas áreas: programación, lenguajes de programación, diseño, análisis...) afectan directamente a la productividad pese a que no existe una literatura que trate este asunto [28]. Además de estos factores, y considerando la falta de literatura relacionada con esta área, se propone que otros factores como la motivación [49], la prácticas vinculadas con los resultados y la evaluación dentro de la gestión del desempeño, los sistemas de compensación y beneficios, el clima organizacional y la felicidad de los trabajadores podrían influir en los resultados de productividad; pero no está claro como pueden influenciar y como pueden ser introducidos en la medición de la productividad. Por lo tanto, se presenta un amplio abanico de posibilidades de investigación al combinar el conocimiento de la gestión de recursos humanos con la gestión de la productividad; de modo que esta combinación, puede llevar a una transferencia de conocimiento con un propósito común de investigación [50].

3.2. Entradas y salidas

En IS las fronteras entre las entradas y las salidas son difusas [51]; por ejemplo, el conocimiento utilizado en los procesos de producción de software puede ser considerado como entrada y como salida, y puede sufrir una transformación durante estos. Un ejemplo de estas fronteras difusas sucede durante la fase de mantenimiento, en la que todos los productos de los procesos de producción de software son utilizados como entradas y son transformados para adaptarlos a las necesidades del cliente, o para solucionar los *bugs* (errores) encontrados. Así pues, debería tenerse en cuenta la realización de un análisis profundo de las entradas ya que el uso de entradas *proxies* (medidas indirectas) introduce sesgos en la medición de la productividad. Además, esta fuente de error es una de las más difíciles de superar, y los posibles métodos para tratar con ello no son todos obvios [52].

Además, definir “una unidad de ingeniería del software” es una tarea difícil, aunque hay ciertas medidas universales de entradas y salidas [29]. Desde el punto de vista de las entradas, algunas medidas de productividad utilizan las horas-hombre como una medida de esfuerzo, y las capacidades y competencias técnicas y no técnicas del personal como un factor correctivo para el esfuerzo. Desde el punto de vista de las salidas, algunas medidas de productividad utilizan las SLOC como medida de cuantificación de la salida y el ratio de errores como un factor correctivo. Otras medidas ampliamente utilizadas son los FP completados y los productos terminados. Estos parámetros son medidas físicas, que pueden ser transformados en unidades monetarias para obtener medidas financieras de productividad. Sin embargo, se debe tener en cuenta que hay algunos problemas con las medidas financieras cuando se utilizan. Por ejemplo, los ingresos no son siempre

una buena medida de la salida, ya que los precios no siempre reflejan la calidad percibida del servicio ofrecido [43].

Una vez que las entradas, salidas, y factores que afectan a la medición de la productividad han sido definidos, es posible formular una medida de productividad. Desde el punto de vista de los autores de este artículo, es sorprendente que a pesar de la controversia existente sobre la validez de algunas medidas de entradas y salidas para medir la productividad en IS se continúen utilizando dichas medidas. Ejemplo de ello son las medidas derivadas de la utilización de las SLOC como salida y de las Horas-Hombre como entrada [42], o la utilización de los FP como salida. Estas medidas continúan en uso en la actualidad en numerosas organizaciones pese a dicha controversia; probablemente bajo la ley que DeMarco y Lister [53] versionan de los principios de la capacidad de ser medido de Gilbs: “Todo lo que necesitas cuantificar puede ser medido de alguna forma mejor que no medirla de ninguna forma”.

4. Puestos de trabajo en Ingeniería del Software

La estructura de cualquier organización, independientemente de su forma, necesita de la definición de los puestos de trabajo que forman dicha estructura [54]. Estos puestos de trabajo son los engranajes que deben dar como resultado el cumplimiento de los objetivos y misión de la organización. Por ello, disponer de la definición de los puestos de trabajo existentes en la organización, junto con su valoración, es una necesidad si se desea realizar cualquier cambio organizacional o si se está creando una nueva organización [55]. En concreto, la definición del puesto de trabajo puede contener la siguiente información: la función del puesto, la situación del mismo en el organigrama organizacional, las tareas que se realizan en el mismo (qué, cómo, con qué, para qué (lo hace), frecuencia, tiempo, autonomía, relaciones con otros puestos...), flujo de trabajo, esfuerzos (físicos, intelectuales), riesgos, condiciones de trabajo, supervisión, conocimientos y competencias necesarios, valores, acceso al puesto y destinos [54]. Como se puede apreciar, el puesto de trabajo no es el sitio físico en el que se realizan las actividades de un trabajador, sino la definición integral de una necesidad organizacional. Además, hay que tener en cuenta que una misma persona puede desempeñar más de un puesto de trabajo en la misma organización; y que varias personas pueden desempeñar el mismo puesto de trabajo.

En IS, existen varios puestos de trabajo más o menos reconocidos universalmente, aunque su definición varía en cada organización y están en constante actualización [56]. Por ejemplo, puestos tales como jefe de proyecto, programador o analista son puestos referenciados tanto en la academia como en la industria desde hace años [57] pero su definición es difusa [58]. Una fuente de consulta de estos puestos de trabajo es el proyecto O*Net creado por el departamento de trabajo de los EE.UU (<http://www.onetonline.org/>). En él es posible consultar descripciones genéricas de puestos de trabajo muy útiles a la hora de crear nuevas o actualizar definiciones de puestos de trabajo existentes. Pese a esta variedad, los puestos de trabajo en IS se pueden agrupar entorno a lo que se denomina trabajadores de cuello blanco (*white-collar worker*), también llamados trabajadores del conocimiento [2, 59]. Estos puestos de trabajo se caracterizan principalmente por la utilización de capital humano, intelectual y relacional para desempeñar las tareas y cumplir los objetivos de los puestos de trabajo [60]. Estas características los diferencian considerablemente de los puestos de trabajos de otras industrias en los que la presencia de la mano de obra desde un punto de vista físico, en lugar de intelectual, es el principal recurso necesario. Además, los recursos que necesitan los puestos de trabajo en IS

son, en gran medida, intangibles (por ejemplo, conocimiento y experiencia) frente a los recursos tangibles de las industrias clásicas [61]. Adicionalmente, la configuración del entorno de trabajo tiene diversas vinculaciones con el puesto de trabajo. Por ejemplo, gracias a la combinación de espacios individuales, para el trabajo técnico con el objetivo de evitar distracciones, con espacios compartidos, para comunicarse y realizar reuniones, es posible aumentar la productividad de las actividades de desarrollo de software [37], pero ¿cuál es la configuración ideal del espacio de trabajo para aumentar la productividad? [62]. Esto hace que la configuración del entorno de trabajo sea un aspecto a tener en cuenta desde la definición de los puestos de trabajo.

Finalmente, y a diferencia de otros sectores, los puestos de trabajo en IS están muy influenciados por la forma en la que el trabajo se estructura. En la mayoría de las organizaciones IS, la organización del trabajo en equipos de trabajo es una realidad [63-65], lo que está contrapuesto con la organización del trabajo en puestos aislados en los que la interacción con otro puesto es recibir una entrada para dar una salida. El trabajo en equipo requiere habilidades y competencias propias de dicha forma de trabajo por lo que su utilización requiere que las definiciones de los puestos de trabajo estén actualizadas de acuerdo con estas nuevas necesidades. Un ejemplo de esta forma de organización del trabajo son las metodologías de desarrollo ágiles [66], cuya utilización va en aumento [67]. Otra de las diferencias es la utilización de una estructura organizacional distribuida a lo largo del planeta, denominada *Global Software Development* (GSD) [68-69]; esta estructura de trabajo tiene como objetivos desarrollar software de forma ininterrumpida y dar mantenimiento continuado. Desde el punto de vista del puesto, esta forma de organización requiere una serie de cambios en los puestos de trabajo: multiculturalidad [70], idiomas [71], gestión de la confianza [72]... Estos cambios hacen que los puestos de trabajo en IS difieran entre cada organización y sea necesario disponer de una definición actualizada de cada puesto de trabajo, máxime cuando se quiere medir la productividad a nivel de puesto de trabajo.

5. Medición de la productividad a nivel de puesto de trabajo en Ingeniería del Software

La medición de la productividad a nivel de puesto de trabajo en IS tiene como origen los trabajos publicados a finales de los años 70 y comienzos de los 80 [19, 35, 73]. En sus comienzos, la medición de la productividad, al igual que otros tipos de medidas, se centraba en la actividad de programación [16]. Por otro lado, en esos años, otros autores indicaban la importancia del factor humano en el desarrollo software [74-75]. En aquel contexto se crearon medidas de productividad basadas en la filosofía ingenieril que utiliza la productividad como sinónimo de eficiencia de recursos. De este modo, se utilizaron medidas basadas en SLOC y posteriormente, tras su definición y creación [40], basadas en FP. Estas medidas no reflejan toda la actividad desarrollada por cada puesto de trabajo, sino que son medidas de productividad que tienen por objetivo medir la eficiencia de la entrega del proyecto.

Con respecto al nivel de medición, la investigación se ha centrado en el sector (por ejemplo, a nivel de país [9]), la organización [76] y proyecto [6] dentro de las mismas de modo que las medidas de productividad a niveles inferiores (equipos de trabajo, puestos de trabajo) no han tenido el mismo interés, quizá por la dificultad de medir a dicho nivel [2]. No obstante, hay que destacar que las medidas empleadas a nivel de puesto de trabajo son las mismas que las empleadas a niveles superiores de medición [77]. Esto representa un problema ya que

los recursos empleados (entradas), los productos y/o servicios generados (salidas) y los factores que afectan a la productividad a este nivel no son los mismos que a niveles superiores, de modo que las medidas empleadas son de dudosa validez para este nivel de medición [78]. Por ello es necesario elaborar nuevas medidas que tengan validez para medir la productividad a nivel de puesto de trabajo. Con el objetivo de establecer los puntos de partida para elaborar estas nuevas medidas se presentan a continuación los puntos clave a tener en cuenta en dicha tarea desde el punto de vista de los factores, de las entradas y las salidas, y de las medidas.

5.1. Factores que afectan a la productividad a nivel de puesto de trabajo en IS

A nivel de puesto de trabajo, los factores que afectan directamente a la productividad, siguiendo la clasificación presentada por Santillo [79] son los referentes al personal del proyecto junto con los relativos a las condiciones de trabajo definidas en el puesto de trabajo [54]. No obstante, de forma indirecta los factores de proceso, producto, y tecnológicos presentados por Santillo deben ser considerados ya que en el caso de disponer de una correcta definición de puesto de trabajo, como la planteada en [54], el proceso corresponde al cómo lo hace, el producto a qué hace, y los tecnológicos al con qué lo hace. Así pues, se observa claramente que la definición de los puestos de trabajo está vinculada con la medición de la productividad ya que aporta la información de partida para plantear una posible medida.

Por otro lado, tal y como se ha planteado anteriormente, los puestos de trabajo en IS son puestos con un alto uso de capital humano por lo que los factores que más pueden influir en la productividad deberían ser los relativos al capital humano. De este modo, factores tales como la motivación, tratada inicialmente en los años 20 por Elton Mayo, y en IS tenida en cuenta desde hace años [49, 80] influyen en la productividad. Sin embargo, no está claro qué motiva a los ingenieros software, cómo están motivados, ni las salidas y beneficios de dicha motivación [49]. Pese a ello, la conclusión general es disponer de trabajadores motivados como fuente de buenos resultados para los proyectos de desarrollo software [81]. Además, existen factores, incluidos también en otras medidas como las relacionadas con la estimación, que influyen en la productividad; por ejemplo, la experiencia en las tareas (análisis, diseño, programación...) o en los lenguajes de programación empleados [34]. Pese a que existe un amplio número de estudios que tratan los factores que afectan a este tipo de puestos de trabajo, el signo y el grado con el que influyen en la productividad, y cómo se afectan entre sí, es una tarea difícil pero necesaria [33].

5.2. Entradas y salidas a nivel de puesto de trabajo en IS

De igual forma que para los factores, las entradas que se utilizan y las salidas que se producen a nivel de puesto de trabajo dependen de cada puesto de trabajo, y deben estar de explícitas en la definición de cada puesto [54]. Si por ejemplo se consideran las salidas clásicas utilizadas (SLOC, FP), y los puestos de trabajo perfilados en Métrica 3 (directivo, jefe de proyecto, consultor, analista, programador) [57], faltarían por definir salidas, generalmente productos y servicios intermedios antes de la puesta en servicio del software. Poniendo el caso del puesto de trabajo analista, para el cual Métrica 3 define su responsabilidad como *“[...] elaborar un catálogo detallado de requisitos que permita describir con precisión el sistema de información, para lo cual mantendrán entrevistas y sesiones de trabajo con los responsables de la organización y usuarios, actuando de el interlocutor entre éstos y el equipo de proyecto en lo que a requerimientos se refiere. Estos requisitos permiten a los analistas elaborar los distintos modelos que sirven de base para el diseño, obteniendo los modelos de datos y de procesos en el caso del análisis estructurado y los modelos de clases e interacción de objetos en análisis*

orientado a objeto. Así mismo realizan la especificación de las interfaces entre el sistema y el usuario.” Teniendo en cuenta esta definición, puede decirse que las salidas del analista son catálogos de requisitos, modelos de datos y procesos o de clases e interacción, y especificaciones de interfaces; por otro lado, las entradas son el conocimiento de los usuarios y organización para la cual se construye el proyecto software y los catálogos de requisitos. Este ejemplo ilustra a la perfección la problemática ya que ni genera las salidas clásicamente utilizadas para medir la productividad (SLOC o FP), la frontera entre las entradas y salidas es difusa (genera un catálogo de requisitos que a su vez sirve como entrada para realizar los modelos) [51], y como entradas no sólo utiliza el tiempo sino que emplea conocimiento y el catálogo de requisitos. Además, para generar las salidas interactúa con otros puestos de trabajo y con personas externas a la organización lo que puede ser visto como otra entrada [54]. Además, otro elemento a tener en cuenta debe ser la calidad (tanto de las entradas utilizadas, como de las salidas generadas) [82]. Así pues, parece claro que las entradas y salidas utilizadas en la medición de la productividad a nivel de puesto de trabajo en IS deben estar ligadas al puesto de trabajo y no a la salida final entregada al cliente.

5.3. Medidas de productividad a nivel de puesto de trabajo en IS

Una vez han sido tratados los factores, y las entradas y salidas, es posible abordar las medidas de productividad a este nivel de medición. Teniendo en cuenta la problemática descrita en los apartados anteriores, tiene sentido plantearse si las medidas de productividad empleadas actualmente en IS son de utilidad para medir la productividad de cada uno de los puestos de trabajo. La respuesta será afirmativa si la medida mide lo que el puesto de trabajo debería aportar y lo que debería utilizar para realizar dicho aporte [83]. De este modo, las medidas clásicas (SLOC/t o FP/t) no son de utilidad para todos los puestos de trabajo en IS. Puede que sean de gran utilidad en puestos de generación de código (programador) pero actualmente no todos los desarrollos emplean código nuevo [84], sino que se reutiliza código y por lo tanto emplear medidas basadas en las SLOC producidas puede carecer de utilidad incluso para dichos puestos [85]. Por otro lado, y utilizando el mismo puesto de trabajo (programador), hay que tener en cuenta las tareas de mantenimiento de software en las que las líneas de código generadas no es tanto la finalidad del puesto sino la solución de errores o la inclusión de nueva funcionalidad [86].

Considerando las observaciones anteriores, y volviendo al ejemplo del analista definido en Métrica 3, la medida de productividad a emplear para dicho puesto sería del tipo:

$$p_1 = f(s_1(\text{catálogo de requisitos}), e_1(\text{conocimiento}), e_2(\text{usuarios}), \text{factores})$$

$$p_2 = f(s_2(\text{modelos}), e_3(\text{catálogo de requisitos}), e_4(\text{usuarios}), e_5(\text{otros puestos}), \text{factores})$$

$$pp = g(p_1, p_2, \text{factores})$$

donde

f es una medida que emplea las entradas y salidas para dar un resultado de productividad,

p_x es una función que emplea como medidas de entradas (e_x) y salidas (s_x) y devuelve un resultado de productividad para una tarea del puesto de trabajo (p_x),

g es una función que emplea como entradas las productividades de cada actividad del puesto y devuelve un resultado de productividad para el puesto de trabajo (pp),

s_x es una medida de una salida,

e_x es una medida de una entrada,

factores son todos los factores que afectan a la productividad en dicha función.

Pese a que la definición textual es sencilla, la dificultad radica en establecer unidades de medición para cada elemento medido y en definir una unidad de productividad que permita incluir las medidas de cada elemento en cada función de productividad. Por otro los factores, incluida la calidad, deben ser evaluados para ver su importancia y signo en la función de productividad.

6. Conclusiones y líneas futuras de investigación

Los proyectos de IS están creciendo en tamaño y está siendo cada vez más y más complicado cumplir con los requisitos, mientras que la entrega es cada vez más urgente [87]. Por lo tanto, la necesidad de disponer de indicadores sobre la eficiencia y eficacia a diversos niveles es una necesidad diaria en las organizaciones del sector TIC. En esta línea, las medidas de productividad representan un indicador de cuán eficiente es el proceso productivo ejecutado a un nivel específico de análisis en una organización. Por otra parte, la investigación sobre medidas de productividad en IS está principalmente centrada en la productividad a nivel de proyecto o a niveles superiores (organización, sector, industria...), y existe poca literatura sobre medidas de productividad a niveles más bajos (a nivel equipos o a individual) [88]. Existen algunos estudios que reflejan la importancia del diseño en los procesos de producción de software (por ejemplo, [15]), pero continua sin estar claro cómo medir la productividad de los diseñadores de software e incluso las tareas de diseño de software. De este modo, las medidas de productividad a nivel de puesto de trabajo en IS es una necesidad tangible [2].

Por un lado, hay que tener en cuenta que este artículo es un breve estado de la cuestión realizado utilizando una metodología clásica de búsqueda revisión de literatura. Por el contrario, otra forma de construirlo es realizar una revisión sistemática de la literatura [89]. Con este tipo de revisión se consigue que el estudio pueda ser replicado ya que se informa de todos los pasos realizados para construirlo; todo lo contrario que con una revisión literaria en la que no hay un proceso de trabajo ordenado y documentado. Así pues, la realización de una revisión sistemática de la literatura sobre medición de productividad a nivel de puesto de trabajo es una línea futura de investigación.

Por otro lado, este artículo no presenta ningún resultado basado en un estudio al tratarse de una revisión de la literatura. De este modo, y con el objetivo de aumentar el conocimiento en el objeto de estudio, es necesario realizar investigaciones y publicar los resultados obtenidos. Considerando la casuística del objeto de estudio, la utilización de metodologías que exploratorias toma ventaja frente a otras metodologías. La principal ventaja de estas metodologías es que la recogida de información está abierta a cualquier nueva entrada frente a las recogidas de información cerradas [90-91]. Una posible línea de investigación es la realización de un estudio cualitativo mediante entrevistas a trabajadores para ver la visión de la medición de la productividad desde los propios trabajadores en IS.

Finalmente, la creación de nuevas medidas de productividad a nivel de puesto en IS permitirá a investigadores de otras áreas de conocimiento elaborar medidas para puestos de otros sectores, principalmente de puestos del conocimiento. Esta tarea continua siendo un reto desde hace una década [59], y continuará siéndolo a no ser que se realicen esfuerzos desde la academia y la industria para elaborar nuevas medidas basadas en las definiciones de los puestos de trabajo y no partiendo de medidas utilizadas a niveles mayores de medición.

Referencias

- [1] Dalcher, D. (2006). Supporting software development: enhancing productivity, management and control. *Software Process: Improvement and Practice* 11(6) 557-559.
- [2] Ramirez, Y.W. & Nembhard, D.A. (2004). Measuring knowledge worker productivity: A taxonomy. *Journal of Intellectual Capital* 5(4) 602-628.
- [3] MacCormack, A., Kemerer, C.F., Cusumano, M. & Crandall, B. (2003). Trade-offs between Productivity and Quality in Selecting Software Development Practices, *IEEE Software* 20(5)78-85.
- [4] Lee, S. & Schmidt, R.C. (1997). Improving application development productivity in Hong Kong. In: *Information technology and challenge for Hong Kong*, B. J.M. & M. B.M. (eds.), Hong Kong University Press.
- [5] Asmild, M., Paradi, J.C. & Kulkarni, A. (2006). Using data envelopment analysis in software development productivity measurement. *Software Process Improvement and Practice* 11(6) 561-572.
- [6] Kitchenham B.A. & Mendes, E. (2004). Software Productivity Measurement Using Multiple Size Measures. *IEEE Transactions on Software Engineering* 30(12) 1023-1035.
- [7] Pendharkar, P.C. (2006). Scale economies and production function estimation for object-oriented software component and source code documentation size," *European Journal of Operational Research* 172(3) 1040-1050.
- [8] Maxwell K.D. & Forselius, P. (2000). Benchmarking Software-Development Productivity. *IEEE Software* 17(1) 80-88.
- [9] Tsunoda, M., Monden, A., Yadohisa, H., Kikuchi, N. & Matsumoto, K. (2009). Software development productivity of Japanese enterprise applications. *Information Technology and Management* 10(4) 193-205.
- [10] Quesnay, F. (1766). Analyse de la formule arithmétique du tableau économique de la distribution des dépenses annuelles d'une nation agricole. *Journal de l'Agriculture, du Commerce & des Finances* 11-41.
- [11] Jefferys, J., Hausberger, S. & Lindblad, G. (1954) *Productivity in the distributive trade in Europe: wholesale and retail aspects*, Organisation for European Economic Co-operation.
- [12] Sink, D.S., Tuttle, T.C. & DeVries, S.J. (1984). Productivity measurement and evaluation: what is available? *National Productivity Review* 3(3) 265-287.
- [13] Fitzgerald L. & Moon, P. (1996). *Performance measurement in service industries: making it work*, Cima.
- [14] Nachum, L. (1999). Measurement of productivity of professional services. *International Journal of Operations and Production Management* 9(9/10) 922-950.
- [15] Anselmo, D. & Ledgard, H. (2003). Measuring productivity in the software industry. *Communications of the ACM* 46(11) 121-125.
- [16] Chrysler, E. (1978). Some basic determinants of computer programming productivity. *Communications of the ACM* 21(6) 472-483.
- [17] Albrecht, A.J. (1979). Measuring application development productivity. *Proc. Joint SHARE/GUIDE/IBM Application Development Symposium*, 83-92.
- [18] Vessey, I. & Weber, R. (1983). Some factors affecting program repair maintenance: an empirical study. *Communications of the ACM* 26(2) 128-134.
- [19] Thadhani, A.J. (1984). Factors affecting programmer productivity during application development. *IBM Systems Journal* 23(1) 19-35.
- [20] Cusumano, M.A. & Kemerer, C.F. (1990). A quantitative analysis of U.S. and Japanese practice and performance in software development. *Management Science* 36(11) 1384-1406.
- [21] Maxwell, K.D., Wassenhove L.V., & Dutta, S. (1996). Software Development Productivity of European Space, Military, and Industrial Applications. *IEEE Transactions on Software Engineering* 22(10) 706-718.
- [22] Scacchi, W. (1994). Understanding Software Productivity. In: *Software Engineering and Knowledge Engineering: Trends for the Next Decade*, Software Engineering and Knowledge Engineering 3, W. D. Hurley, (ed.), pp. 293-321.

- [23] Gaffney, J. (1989). Software reuse--key to enhanced productivity: some quantitative models. *Information and Software Technology* 31(5) 258-267.
- [24] Banker, R.D. & Kauffman, R.J. (1991). Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study. *MIS Quarterly* 15(3) 375-401.
- [25] Puroo, S. & Vaishnavi, V. (2003). Product metrics for object-oriented systems. *ACM Computing Surveys* 35(2) 191-221.
- [26] Kitchenham, B.A. (2010). What's up with software metrics? - A preliminary mapping study. *Journal of Systems and Software* 83(1) 37-51.
- [27] Bellini, C., Pereira, R. & Becker, J. (2008). Measurement in software engineering: from the roadmap to the crossroads. *International Journal of Software Engineering and Knowledge Engineering* 18(1) 37-64.
- [28] Pfleeger, S.L. (2008). Software Metrics: Progress after 25 Years?. *IEEE Software* 25(6) 32-34.
- [29] Boehm, B.W., Abts, C. & Chulani, S. (2000). Software development cost estimation approaches - A survey. *Annals of Software Engineering* 10(1) 177-205.
- [30] Boehm, B.W. & Ross, R. (1989). Theory-W Software Project Management Principles and Examples. *IEEE Transactions on Software Engineering* 15(7) 902-916.
- [31] Dale, C.J. & van der Zee, H. (1992). Software productivity metrics: who needs them? *Information and Software Technology* 34(11) 731-738.
- [32] Gummesson, E. (1992). Quality dimensions: what to measure in service organizations. In: *Advances in services marketing and management*, T. A. Swartz, et al., (eds.), JAI Press, pp. 64-78.
- [33] Wagner, S. & Ruhe, M. (2008). A Systematic Review of Productivity Factors in Software Development. *Proc. 2nd International Workshop on Software Productivity Analysis and Cost Estimation (SPACE 2008)*, IEEE Computer Society.
- [34] Boehm, B.W. (1981). *Software Engineering Economics*, Prentice Hall PTR.
- [35] Walston, C.E. & Felix, C.P. (1977). A method of programming measurement and estimation. *IBM Systems Journal* 16(1) 54-73.
- [36] Paiva, E., Barbosa, D., Lima, R. & Albuquerque, A. (2010). Factors that Influence the Productivity of Software Developers in a Developer View. In: *Innovations in Computing Sciences and Software Engineering*, T. Sobh & K. Elleithy (eds.). Springer Netherlands, pp. 99-104.
- [37] Sommerville, I. (2010). *Software Engineering*, Addison-Wesley.
- [38] Gómez, O., Oktaba, H., Piattini, M. & García, F. (2008). A Systematic Review Measurement in Software Engineering: State-of-the-Art in Measures. *Software and Data Technologies, Communications in Computer and Information Science* 10, J. Filipe, et al. (eds.), Springer Berlin Heidelberg, pp. 165-176.
- [39] Jørgensen, M. & Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering* 33(1) 33-53.
- [40] Albrecht, A.J. & Gaffney, J.E. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transaction Software Engineering* 9(6) 639-648.
- [41] Al-Darrab, I.A. (2000). Relationships between productivity, efficiency, utilization, and quality. *Work Study* 49(3) 97-104.
- [42] Boehm, B.W. (1987). Improving Software Productivity. *Computer* 20(9) 43-57.
- [43] Grönroos, C. & Ojasalo, K. (2004). Service productivity: Towards a conceptualization of the transformation of inputs into economic results in services. *Journal of Business Research* 57(4) 414-423.
- [44] Gummesson, E. (1995). "Service productivity: a blasphemous approach. In: *Quality, productivity and profitability in service operations (conference papers from the QP&P Research Program 1992-1994)*, E. Gummesson, (ed.), University of Stockholm, Department of Business Administration, pp. 8-22.
- [45] Krishnan, M.S., Kriebel, C.H., Kekre, S. & Mukhopadhyay, T. (2000). An Empirical Analysis of Productivity and Quality in Software Products. *Management Science* 46(6) 745-759.
- [46] Hsu, J. S.-C., Chen, H.-G., Jiang, J. & Klein, G. (2010). The Role of User Review on Information System Project Outcomes: A Control Theory Perspective. *International Journal of Information Technology Project Management* 1(1) 1-14.
- [47] Premraj, R., Shepperd, M., Kitchenham, B.A. & Forselius, P. (2005). An Empirical Analysis of Software Productivity over Time. In: *Book An Empirical Analysis of Software Productivity over Time*, Series An Empirical Analysis of Software Productivity over Time, IEEE Computer Society, pp. 37.

- [48] Colomo-Palacios, R., Tovar-Caro, E., García-Crespo, Á. & Gómez-Berbis, J. (2010). Identifying Technical Competences of IT Professionals: The Case of Software Engineers. *International Journal of Human Capital and Information Technology Professionals* 1(1) 31-43.
- [49] Beecham, S., Baddoo, N., Hall, T., Robinson, H. & Sharp, H. (2008). Motivation in Software Engineering: A systematic literature review," *Information and Software Technology* 50(9-10) 860-878.
- [50] Koskinen, K.U. (2008). Boundary brokering as a promoting factor in competence sharing in a project work context. *International Journal of Project Organisation and Management* 1(1) 119-132.
- [51] Gupta, A. (1995). Productivity measurement in service operations: a case study from the health-care environment. *Managing Service Quality* 5(5) 31.
- [52] Rees, A. (1980). Improving Productivity Measurement. *The American Economic Review* 70(2) 340-342.
- [53] DeMarco, T. & Lister, T. (1999). *Peopleware: productive projects and teams*, Dorset House Publishing Co., Inc.
- [54] Fernández-Ríos, M. (1995). *Análisis y descripción de puestos de trabajo*, Diaz de Santos, 1995, p. 423.
- [55] Schuler, R.S. & Jackson, S.E. (1987). Linking Competitive Strategies with Human Resource Management Practices. *The Academy of Management Executive* (1987-1989) 1(3) 207-219
- [56] Garcia-Crespo, A., Colomo-Palacios, R., Gomez-Berbis, J. M.,& Tovar-Caro, E. (2008). The IT Crowd: Are We Stereotypes?. *IT Professional* 10(6) 24-27.
- [57] *Métrica Versión 3* (2000). Consejo Superior de Informática - Ministerio de Administraciones Públicas.
- [58] Alexander, S. (1999). What's in a job title? Less and less, some say. *InfoWorld* 21(21) 1999.
- [59] Drucker, P. (1999). Knowledge-Worker Productivity: The Biggest Challenge. *California management review* 41(2) 79-85.
- [60] Colomo-Palacios, R., Cabezas-Isla, F., García-Crespo, Á. & Soto-Acosta, P. (2010). Generic Competences for the IT Knowledge Workers: A Study from the Field Knowledge Management, Information Systems, E-Learning, and Sustainability Research. *Communications in Computer and Information Science* 111, M. D. Lytras, et al. (eds.), Springer Berlin Heidelberg, pp. 1-7.
- [61] Rus, I. & Lindvall, M. (2002). Knowledge management in software engineering. *IEEE Software* 19(3) 26-38.
- [62] Jones, C. (1995). How Office Space Affects Programming Productivity. *Computer* 28(1) 76.
- [63] Carmel, E. (1999). *Global software teams: collaborating across borders and time zones*, Prentice Hall.
- [64] Gorla, N. & Lam, Y.W. (2004). Who should work with whom?: building effective software project teams. *Communications of the ACM* 47(6) 79-82.
- [65] Sawyer, S. (2004). Software development teams. *Communications of the ACM* 47(12) 95-99.
- [66] Cockburn, A. & J. Highsmith, J. (2001). Agile software development, the people factor. *Computer* 34(11) 131-133.
- [67] Dyba, T. & Dingsoyr, T. (2009). What Do We Know about Agile Software Development? *IEEE Software* 26(5) 6-9.
- [68] Herbsleb, J.D. & Moitra, D. (2001). Global Software Development. *IEEE Software* 18(2) 16-20.
- [69] Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á. & Soto-Acosta, P. (2010). Team Software Process in GSD Teams: A study of new work practices and models. *International Journal of Human Capital and Information Technology Professionals* 1(3) 32-53.
- [70] Linna, P., Karttunen, E. & Jaakkola, H. (2011). Software engineering companies' multicultural education. In *Proc. 34th International Convention (MIPRO, 2011)*, pp. 1140-1145.
- [71] Abufardeh, S. & Magel, K. (2010). The impact of global software cultural and linguistic aspects on Global Software Development process (GSD): Issues and challenges. In *Proc. 4th International Conference on New Trends in Information Science and Service Science (NISS, 2010)*, pp. 133-138.
- [72] Moe, N.B. & Šmite, D. (2008). Understanding a lack of trust in Global Software Teams: a multiple-case study. *Software Process: Improvement and Practice* 13(3) 217-231.
- [73] Jones, C. (1981). *Programmer Productivity: Issues for the Eighties*, IEEE Computer Soc. Press.
- [74] DeMarco, T. & Lister, T. (1985). Programmer Performance and the Effects of the Workplace. In: *Proc. 8th International Conference on Software Engineering. New York: Institute of Electrical and Electronics Engineers*, pp. 268-272.
- [75] DeMarco, T. & Lister, T. (1987). *Peopleware: productive projects and teams*, Dorset House Publishing Co., Inc.
- [76] Anda, B.C.D., Sjoberg, D.I.K. & Mockus, A. (2009). Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System. *IEEE Transaction Software Engineering* 35(3) 407-429.
- [77] Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á. & Cabezas-Isla, F. (2011). Software Engineering Productivity: Concepts, Issues and Challenges. *International Journal of Information Technology Project Management* 2(1) 37-47.

- [78] Briand, L. C., Morasca, S. & Basili, V. R. (2002). An Operational Process for Goal-Driven Definition of Measures. *IEEE Transactions on Software Engineering* 28(12) 1106-1125.
- [79] Santillo, L. & Moretto, G. (2011). *Software Productivity Factors Taxonomy*, GUFPI-ISMA.
- [80] Boehm, B.W. (1981). An Experiment in Small-Scale Application Software Engineering. *IEEE Transactions on Software Engineering* 7(5) 482-493.
- [81] Hall, T., Sharp, H., Beecham, S., Baddoo, N. & Robinson, H. (2008). What Do We Know about Developer Motivation? *IEEE Software* 25(4) 92-94.
- [82] Procaccino, J.D., Verner, J.M., Shelfer, K.M. & Gefen, D. (2005). What do software practitioners really think about project success: an exploratory study. *Journal of Systems and Software* 78(2) 194-203.
- [83] Koch, M.J. & McGrath, R.G. (1996). Improving labor productivity: Human resource management policies do matter. *Strategic Management Journal* 17(5) 335-354.
- [84] Mohagheghi, P. & R. Conradi, R. (2007) Quality, productivity and economic benefits of software reuse: a review of industrial studies. *Empirical Software Engineering* 12(5) 471-516.
- [85] Banker, R.D., Datar, S.M. & Kemerer, C. (1987). Factors Affecting Software Maintenance Productivity: An Exploratory Study. In: *Proc. Eighth International Conference on Information Systems*, pp. 160-175.
- [86] ISO (2006). *ISO/IEC 14764*.
- [87] Boehm, B.W. (2006). A view of 20th and 21st century software engineering. In: *Proceedings of the 28th international conference on Software engineering (ICSE'06)*, pp. 12-29.
- [88] Erne, R. (2011). What is Productivity in Knowledge Work? - A Cross-industrial View -. *Journal of Universal Computer Science* 17(10) 1367-1389.
- [89] Kitchenham, B.A. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*, Keele University and University of Durham.
- [90] Strauss, A. & Corbin, J. (1990). *Basics of qualitative research*, Sage.
- [91] Bardin, L. (1986). *El análisis de contenido*, Akal.