

Midiendo la productividad de un equipo de software

Carlos Eduardo Vazquez
Fatto Consultoria y Sistemas (www.fattocs.com)
carlos.vazquez@fattocs.com

Alcanzar la máxima productividad de los equipos de software es el deseo de cualquier empresa que tiene como objetivo el éxito de su negocio. Pero, ¿Cómo asegurar que un equipo de software es realmente productivo? Y ¿Cómo mejorar el desempeño para lograr un alto rendimiento?. Estas son algunas de las preguntas aclaradas por Carlos Eduardo Vázquez, experto en Puntos de Función.

Fecha de la Entrevista: 12/03/2010 | Entrevistado: Carlos Eduardo Vázquez

LG: ¿Qué es un equipo de desarrollo de software productivo?

Carlos Vazquez: La respuesta es simple: Es un equipo que produce. En este caso, vale la pena preguntarse: ¿Qué es lo que produce?

El término software se refiere a una variedad de productos tales como programas de computador, especificaciones de interfaces con el usuario y otros componentes de software, documentos de requisitos, proyectos de software, etc. Por lo tanto, cuando se discute de productividad en el contexto de software es necesario, ante todo, establecer cuál es la perspectiva en que se quiere evaluar y qué tan productivo es un equipo en comparación con la escala de productividad. Esta escala es necesaria debido a que el interés general no es saber si un equipo es productivo, pero si:

1. Qué tan productivo es
2. Qué tan productivo necesita ser
3. Cómo el equipo está posicionado en comparación con otros equipos del mercado.

Cuando se elige una perspectiva económica, independiente de los detalles técnicos - y hay muchos detalles técnicos involucrados en el desarrollo de software - una buena opción es considerar como producto la funcionalidad entregada por el proyecto a los usuarios. La respuesta de demandas de mejora de un software ya existente puede tener como producto las nuevas funcionalidades incluidas y las funcionalidad existente, ya sean excluidas o alteradas.

- **LG: Sabemos que hay diferentes actividades relacionadas con el software, tales como: correcciones, mejoras funcionales, mejoras tecnológicas, etc. ¿Es posible medir la productividad en todos ellos? ¿De qué forma?**

Carlos Vazquez: Siempre que se hable de productividad, en primer lugar, debemos evaluar lo que se está comprando. Cuando hablamos de correcciones (no cubiertas por la garantía), soporte de tercer nivel y otras actividades de ayuda, se debe evaluar lo que se está comprando y la disponibilidad del servicio (en lugar de comprar un producto). Para que se pueda hablar de productividad, es necesario que exista un producto o servicio que esté siendo producido.

En una ventana de tiempo determinada, ningún trabajo de apoyo puede ser exigido. Pero, cuando lo es, la atención debe cumplir con ciertos niveles de servicio como la fecha de inicio o el plazo para la resolución de un problema, por ejemplo.

En ese tipo de demanda, lo importante no es la productividad, sino el nivel de servicio. La gestión de este tipo de demanda debe ser realizada según la planeación y el monitoreamiento de los Acuerdos de Nivel de Servicio, y no por la productividad.

Supongamos que una mejora tecnológica se refiere a aspectos que no se asignan a las prácticas ni a los procedimientos de negocio. Esta mejora no afecta ninguna funcionalidad. Cuando se trata de una intervención de gran impacto, que afecta la aplicación en su conjunto, podemos decir que el producto entregado son las funcionalidades existentes en un nuevo medio o con una nueva infraestructura de soporte. Por lo tanto, puede ser tratado como un nuevo desarrollo en términos de cuantificación de la producción y evaluación de la productividad.

Vale destacar que cuando hablamos de productividad, considerar que medimos a un equipo es una simplificación de las cosas. De hecho, lo que se evalúa es un proceso productivo. Lo que no se puede hacer es poner un tipo de demanda de este tipo, por ejemplo de tipo refactoring, en un mismo grupo de desarrollo de un nuevo sistema desde cero: cada proceso de producción tendrá su propia productividad media y un conjunto de entradas y salidas diferentes.

La "mejora tecnológica" en una escala menos amplia – que no se trata de un refactoring-, también puede tener su producción cuantificada. Con todo eso, la mejor forma de hacerlo es establecer un peso relativo comparado que llamó "el mantenimiento funcional" para que se pueda trabajar con una unidad de medida de software única (UMS).

En este escenario, una unidad de medida funcional (UMF) es equivalente a una UMS y hay una "lista de precios" expresada en UMS para tipos de mejora tecnológica no medibles por la UMF. Fue confuso, ¿no es así? Vamos a intentar con un ejemplo:

En un proceso en particular que demora dos días de procesamiento hoy, los programas tendrían que ser reestructurados para que ese procesamiento demorara apenas un día. En ese caso, sólo nos resta considerar los artefactos técnicos que deben sufrir intervención. Después de todo, no se alteró la función.

En tiempo de definición de políticas y supuestos, lo ideal es comparar en ensayos controlados el nivel de trabajo promedio involucrado con cada programa y crear una escala relativa a un UMF. Digamos

que cada programa sea equivalente a cinco horas de trabajo y que la tasa de entrega prevista sea en promedio de 10 horas por UMF. Podemos decir que el producto referente a cada programa será "equivalente a 0,5 UMF". Como no tiene relación con algo funcional, serían 0,5 UMS. En tiempo de medición de demandas, y habiendo la necesidad de intervención de ese tipo en 10 programas, serán considerados para efectos de medición, 5 UMS.

Lo que llamamos de "mejora funcional" no existe en un sentido estricto. Hay mantenimientos que modifican las reglas y procedimientos de negocio asignados como funciones de aplicación (requisitos funcionales). Sin embargo, no hay función que no requiera de un soporte tecnológico ni de una infraestructura técnica. Las intervenciones relativas a la atención de esos nuevos requisitos funcionales no deben pasar por la ponderación descrita en el párrafo anterior, y utilizar sólo las unidades de medición funcional para la ponderación.

- **LG: ¿Cuáles son las ventajas del uso del APF en la medición de la productividad de un equipo de software?**

Carlos Vazquez: El APF como unidad de medición funcional es el método con mayor nivel de madurez que existe y mayor apoyo en términos profesionales y organizacionales con experiencia. Además, existe el IFPUG - Grupo Internacional de Usuarios de Puntos de Función - que es el responsable del mantenimiento y evolución de la técnica desde 1986.

Me tomo la libertad de ampliar su pregunta: ¿Por qué utilizar la medición funcional? Debido a que considera los requisitos funcionales, asignados a las prácticas y procedimientos del negocio, externos a la función de desarrollo y, por tanto, relacionados directamente con la esfera del control del cliente, que es quien demanda los servicios de desarrollo. Si utilizamos alguna métrica interna (como desafortunadamente es inevitable en aquellas demandas del ítem anterior) no sería posible que el cliente cuestionara el resultado presentado por el desarrollo. Eso por sí sólo, en mi opinión ya es motivo suficiente.

Esto permite la creación de un proceso de gestión en el que las tendencias opuestas, pero complementarias, interactúen en la búsqueda de un equilibrio. ¿Cuáles tendencias? El desarrollador tiene tendencia a inflar su producción, mientras que el cliente tiene la tendencia opuesta a disminuir la medición de su demanda. Lo interesante es que el APF no es una discusión apasionada. Existe todo un conjunto de normas y procedimientos que cumplen el papel de moderar estas interacciones.

- **LG: ¿Una persona que no está en la área de TI puede utilizar esta técnica?**

Carlos Vazquez: Claro que sí. He entrenado a más de 2.000 personas en este tema y puedo decir con seguridad que más de la mitad del tiempo es dedicado a apoyar a los profesionales de TI a des-aprender una serie de detalles técnicos que dificultan la medición. Incluso, existe una minoría que por una cuestión de cultura, era del área de negocios y tuve la oportunidad de observar el proceso de aplicación

de la técnica de manera adecuada.

- **LG: ¿Por qué cree que la productividad debe ser medida?**

Carlos Vazquez: Porque es imposible gerenciar aquello que no se puede medir. Cualquier reunión de negocios se convierte en conversación de bar. Si no existiera una métrica de producto y de productividad, cualquier reunión para discutir el proceso de trabajo, el rendimiento del equipo y las consecuencias de la adopción de una nueva herramienta, se convertirían en una conversación sólo de puntos vista con un nivel de incertidumbre muy alto.

- **LG: Después de medir, ¿Cómo mejorar la productividad?**

Carlos Vazquez: Una organización opera en diferentes niveles. Lo que hemos discutido hasta ahora se refiere a una perspectiva táctica y estratégica. Existe la necesidad de una supervisión cotidiana en una escala operacional, es decir, hay necesidad de liderazgo efectivo. Existe la necesidad de profesionales técnicos con habilidades no sólo para desempeñar su trabajo, sino también para actuar como multiplicadores de habilidades en un escenario en el que la demanda por esos servicios sigue aumentando y el volumen de los recién llegados es grande. Existe la necesidad de procesos que no limiten la creatividad y que no sean sólo pedazos de papel para mantener una certificación, sino una cosa viva que cumpla con el propósito de guiar la secuencia de trabajo y las normas de los productos entregados entre las partes interesadas. Existe la necesidad de talento, de personas que sean apasionados por lo que hacen y no se sientan saturadas por un proceso que, en lugar de mejorar sus capacidades, las deterioren.

La ventaja de un modelo de acompañamiento de la productividad bien estructurado, es la libertad que trae al equipo de desarrollo (y, por tanto, una gran responsabilidad). No se pretende definir la productividad de una orden de servicio en particular, ni crear un modelo automático de gerencia, sino determinar una meta de productividad para cada proceso o versión del producto entregado. El equipo de desarrollo tiene la libertad de realizar toda la gestión de su propio trabajo, siempre y cuando al final del proyecto, cuando se calcula el total de las entregas y el total invertido en términos de horas de equipo de desarrollo, se respeten los objetivos de productividad y calidad.

- **LG: En su opinión, ¿Qué es lo que más perjudica la productividad de un equipo?**

Carlos Vazquez: Veo dos mundos: el mundo del desarrollo de software y el otro de mantenimiento. En un primer momento, veo que el principal enemigo de la productividad está en el plan de los requisitos. Imagine una obra en la que se comienza con la carpintería externa mientras se trabaja en cimentación. Imagine como serían hechas las modificaciones en una planta en AutoCAD con las misma intensidad intensidad y frecuencia con las que son realizadas en el cantero de obras. Hoy existen dos estrategias principales para el desarrollo sistemas: el desarrollo en cascada, en el que sólo después de terminado el trabajo de requisitos, se inician los demás trabajos.

¿Cuál es el problema? El usuario no consigue materializar sus necesidades y deseos sin materializar mejor la solución y, como consecuencia, sólo cuando recibe el producto en un estado más avanzado, obtiene éxito en la elicitación y análisis de los requisitos. Abordando también ese problema, se presentan enfoques iterativos e incrementales en los cuales no hay más coincidencia entre las fases de

acompañamiento gerencial del proyecto que se despeguen de las disciplinas de la ingeniería de software. En términos más simples, en una fase de concepción de la solución, no sólo existe el ejercicio de las disciplinas de los requisitos, sino también se hace uso de las disciplinas de análisis y desarrollo o implementación. La intención es abordar lo más pronto posible los aspectos de mayor riesgo y materializarlos para los usuarios a través de prototipos, que yo prefiero llamar de juguetes; aquello que se recibe cuando se entrega el producto. El problema es que, aunque ya no es una novedad, en mi opinión pocos ponen en práctica esa estrategia de desarrollo. Este tema es muy amplio y no pretendo salirme del enfoque de la entrevista.

En el mundo del mantenimiento, tuve contacto con los sistemas de misión crítica con treinta años de edad en la producción y el mantenimiento continuo. ¿Es eso un problema? No necesariamente. Pero una cosa es un entorno en el que se invierte en el mantenimiento de manera que no se degrade la integridad y la estructura del software, y otra es el contexto de "¿Cuál es la forma más rápida y económica para hacer que el sistema funcione de esa manera?". Creo que cualquier persona que trabaje con el mantenimiento, entiende perfectamente lo que estoy diciendo. Esa pérdida de la estructura presenta tanta complejidad en el software, que a menudo parece que tiene vida propia y eso aumenta exponencialmente el trabajo necesario para cumplir con la inclusión, modificación o exclusión de una función.

- **LG: En su opinión, ¿Cómo podemos motivar a un equipo de desarrollo de software para que sea cada vez más productivo?**

Carlos Vazquez: El manifiesto ágil no debe ser ignorado en la evaluación de su cuestionamiento. Yo creo que no son sólo los procesos y herramientas, también son las personas. Me gustaría resaltar un punto crítico: veo a los practicantes de Scrum pregonando los aumentos de la productividad, pero nunca vi que eso se midiera de forma consistente.

Carlos Eduardo Vazquez



Socio fundador de Fatto Consultoría y Sistemas. Especialista en la técnica de Análisis de Puntos de Función. Está certificado por la Grupo Internacional de Usuarios de Puntos de Función - IFPUG. Graduado de la PUC-Rio, con más de 20 años de experiencia en desarrollo, mantenimiento y gestión de sistemas corporativos. Actúa como instructor y consultor de empresas. También es autor del libro "Análisis de Puntos de Función: Medición, Estimaciones y Gestión de Proyectos de Software."