



**Autor: Dandara Pereira Aranha**

## **Estimaciones de Proyectos Ágiles de Software**

En los últimos años, los métodos de desarrollo de software ganaron mucha atención en el campo de la ingeniería de software. Son basados en una filosofía descrita en el manifiesto ágil publicado en el año 2001. Surgieron como alternativas a los llamados métodos tradicionales de desarrollo de software y las necesidades de atender las crecientes presiones de las organizaciones por innovación, productividad (plazos cada vez más cortos), flexibilidad y mejora en el desempeño/calidad de los proyectos de desarrollo de software.

Independientemente de la metodología de desarrollo de software que está siendo utilizada, las estimaciones deben ser una de las principales actividades de planeación de software, “*No se puede controlar aquello que no se puede medir*” afirmaba Peter Drucker. Ósea, un desconocimiento **cuantitativo** puede llevar al completo caos y a que su proyecto no alcance el éxito deseado. Un estudio de QSM sobre el desempeño ágil mostro consistentemente que es un buen plano que define el éxito en el desarrollo de software, y no la metodología de desarrollo.

Mucho Practicantes de las metodologías ágiles son adaptados al movimiento #NoEstimate, pues creen que no existe valor en estimar si usted está trabajando en pequeños incrementos de software, sprints cortas para mejorar su backlog. Entretanto, algunas investigaciones como la realizadas por la CA Technologies, referencia en servicios ágiles, muestran que las entregas realizadas por equipos que tienen el hábito de realizar estimaciones, son más rápidas que las entregas de los equipos que no realizan ninguna estimación. El movimiento #NoEstimates puede ser una solución en algunos casos, por ejemplo, si su equipo ya es trabajar con pequeñas tareas con un equipo box establecido (2,3 días), o en el contexto de las estimaciones operacionales con ventanas de 2 a 4 semanas. Pero no puede ser aplicado de forma absoluta, en cualquier caso.

Las metodologías ágiles proponen un conjunto grande de técnicas para estimar y planear proyectos, principalmente en términos de modelos no basado en algoritmos, La mayoría de las técnicas de estimaciones ágiles se concentran en el uso de "Historias de Usuarios".

Es importante recordar que no existe una mejor técnica, existe la que puede ser más adecuada a su propósito, o a su equipo o al proyecto/producto que usted está trabajando. Solo debemos estimar si eso fuera necesario para algo que traiga valor, y debemos siempre asociar las estimaciones a un propósito.

Dentro de las técnicas de estimaciones ágiles más conocidas tenemos:

### **Planning Poker**

Es una variación de la técnica Delphi, que estima el esfuerzo exigido para concluir tareas y no proyectos interiores. (COHN, 2005) y (HIGHSMITH, 2009) proponen esa técnica para realizar estimaciones en el nivel del proyecto. funciona de la siguiente manera:

Cada miembro del equipo tiene una baraja de cartas basada en una secuencia de Fibonacci modificada con algunas cartas adicionales, representando los puntos a ser atribuidos a cada historia de usuario. Uno a uno, el representante del cliente explica los requisitos descritos en la historia de usuario. Una vez que la historia es descrita, los miembros del equipo pueden hacer algunas preguntas para esclarecer su alcance. Después de eso, cada miembro, al mismo tiempo, muestra una carta de su conjunto con una estimación en puntos. Si estas son todas iguales, la historia recibe las estimaciones, si no, lo miembros que proponen la estimación más alta y las más baja explican sus puntos de vista y nuevas rodadas son jugadas hasta llegar a un consenso.

Entonces, el equipo prevé la velocidad, que es el número de puntos que estos pueden entregar en una interacción. El cual puede ser obtenido por medio de datos históricos, una interacción de prueba o apenas una suposición. Con ese dato se puede establecer la longitud de la interacción. El numero de la interacción es obtenido por la división del número total de puntos por la velocidad. De esa forma, la duración de los proyectos es calculada multiplicando el número de interacciones por el tamaño.

No es tan adecuada para estimaciones de plazo de duración de proyectos, o para la asignación de los recursos en el nivel de proyectos (o programa), pues es una medida de complejidad, subjetiva y particular a cada equipo. Entretanto, realizar estimaciones de forma colaborativa entre todos los miembros del equipo es muy válido para aumentar el comportamiento en la entrega, aclarar dudas, intercambiar ideas, esclarecer requisitos y percibir riesgos que usted no percibiría solo. Así, se evita la tendencia de las personas y organizaciones al subestimar el tiempo que va a necesitar para completar una tarea, mismo cuando tiene experiencia en las tareas similares.

### **Día Ideal**

(COHN, 2005) Afirma que un día ideal corresponde a cantidades de trabajo que un profesional de nivel senior con influencia en las tecnologías y herramientas involucradas consigue realizar en un día de trabajo sin interrupciones. Se trata de una estimación empírica, ejecutada por expertos para el desarrollo con base en una exploración adaptable (ALVES; FONSECA, 2008).

La velocidad es calculada a partir del número de horas que el quipo gasta para implementar un trabajo equivalente a un día ideal.

En caso el ítem pase un día de trabajo, es sugerido descomponer ese ítem en ítems menores que se consiga implementar en apenas un día.

Es más indicada para equipos que todavía no poseen experiencia en realizar estimaciones, pues es fácil de ser entendida.

Mientras tanto, al medir el tamaño de una actividad en días ideales, todavía estamos asociando la noción del tiempo, lo que no siempre es una buena estrategia. Diversos factores pueden influenciar ese tipo de estimación como: evolución de la experiencia del tiempo, conocimiento adquiridos, cambios de *frameworks*, entre otros. El mismo equipo ahora puede entender que una funcionalidad similar a una anterior puede ser concluida en menos días. La base de la estimación muda es así como el histórico es perjudicado.

### **The planning game**

El objetivo del planning game es maximizar el valor agregado y minimizar el costo del sistema a ser desarrollado. Esa técnica, propuesta por la eXtreme Programming, presupone que el cliente es conocedor del negocio, sabiendo relacionar y priorizar los requisitos del sistema de acuerdo con su valor

agregado. Por otro lado, el programador es conocedor de las cuestiones técnicas sabiendo estimar el costo de la implementación de esos requisitos (SHORE; WADEN, 2008). Sus 2 reglas básicas son: (i) o cliente prioriza y (ii) o programador estima.

El juego comienza con el cliente relacionando los requisitos en cartas. En seguida, esas cartas son colocados en orden, de acuerdo con el valor agregado observado por el cliente. Por fin el programador debe estimar el costo de la implementación de cada requisito priorizado. Esos pasos son repetidos hasta que todos los requisitos hayan sido priorizados y estimados.

Durante el proceso, desarrolladores y clientes interactúan para resolver dudas sobre prioridades y estimaciones. Inicialmente, esta técnica fue utilizada para evaluar el trabajo en una interacción, pero también puede ser usada para realizar una planeación de una rease completa.

Nuevamente la mayor ventaja está en la colaboración e interacción, en ese caso entre el cliente y el desarrollo. Con todo el cliente puede presentar varias partes interesadas, que pueden poseer diferentes intereses y así puede no ser tan fácil llegar a un acuerdo con relación al valor agregado de un determinado requisito, de esa misma manera, el conjunto de programadores puede diferir en la estimación de costo de los requisitos y en esos casos ellos deben llegar a un consenso.

### **Métricas de Tamaño**

Métricas de tamaño nacerán con un objetivo de estimar esfuerzo y plazos asociados al desarrollo de software segundo (PUTMAN, 1992). Es una de las primeras y principales actividades relacionadas a las estimaciones de software. Y de acuerdo con las investigaciones realizadas por (USMAN; MENDES e BORSTLER, 2015), El story points o el punto de función son métricas de tamaño más utilizadas en proyectos ágiles.

El story point es una medida relativa donde el equipo decide cuánto vale un punto y basado en ese tamaño determina cuantos puntos cada actividad posee.

Puntos de función ofrecen una medida estandarizada del tamaño funcional de los requisitos del usuario, independientemente de cualquier aspecto de implementación de estos en el software.

Aunque la técnica de puntos de función haya surgido mucho antes de los story points, ambos pueden ser utilizados de manera similar en proyectos ágiles. Se puede estimar las historias de usuarios, los sprint y el product backlog en puntos de función. La idea de velocidad (o productividad) también puede ser derivada: PF/sprint. Con esa información se puede entonces definir el número de sprints de un release.

La velocidad inicial puede ser más fácilmente obtenida con los puntos de función porque es una medida estando entre los proyectos.

La técnica de puntos de función provee una medida objetiva e incomparable, y así permite una visión táctica y estratégica sobre el desarrollo de software, análisis de viabilidad, benchmarking y puede ayudar a entender las variaciones de productividad y crecimientos del alcance entre proyectos. Es más adecuada cuando se necesita un factor de normalización para comparación de software y para medir la funcionalidad solicitada por el usuario, antes de los proyectos de software, de forma a estimar su tamaño y su costo.

### **Conclusión**

Los proyectos de desarrollo están siendo más grandes y más complejos, desafiando los equipos a conocer y a determinar plazos de entrega realistas. Así, es imprescindible que los equipos realicen estimaciones que los permitan prever esfuerzo, plazos y costos de sus proyectos.

Es importante decir que las estimaciones no deben ser difíciles, costosas o ineficientes. Hechas de manera correcta pueden ser muy efectivas y ayudar a entregar valor a sus organizaciones. Métodos y métricas para estimaciones en su mayoría son aplicadas a un determinado contexto de desarrollo ágil con adaptaciones a fin de que se adecuaran a un proyecto en cuestión. Osea, deben sufrir adaptaciones para que se adecuen al contexto de su proyecto, pues cada uno de ellos tiene características propias que pueden influenciar en el resultado de las estimaciones.

## Referencias Bibliográficas

ALVES, F.; FONSECA, M. Ideal day e priorização: Métodos Ágeis no planejamento. Engenharia de Software, Rio de Janeiro, 2008.

COHN, M. Agile Estimating and Planning, Prentice Hall, 1ª edição, 2005

HIGHSMITH, J. Agile Project Management: Creating Innovative Products. [S.l.:s.n.], 2009.

Muhammad Usman, Emilia Mendes, and Jürgen Börstler. 2015. Effort estimation in agile software development: a survey on the state of the practice. In Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering (EASE '15). ACM, New York, NY, USA

PUTMAN, L. H.; WARE M. Measures for Excellence: Reliable Software On Equipo, Within Budget, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1992.

SHORE, J.; WADEN, S. The Art of Agile Development. [S.l.: s.n.], 2008.

Doug Putman; Good Planning – Not Development Methodology – Is the Key to Successful Project Delivery , 2018