

Estimaciones Ágiles

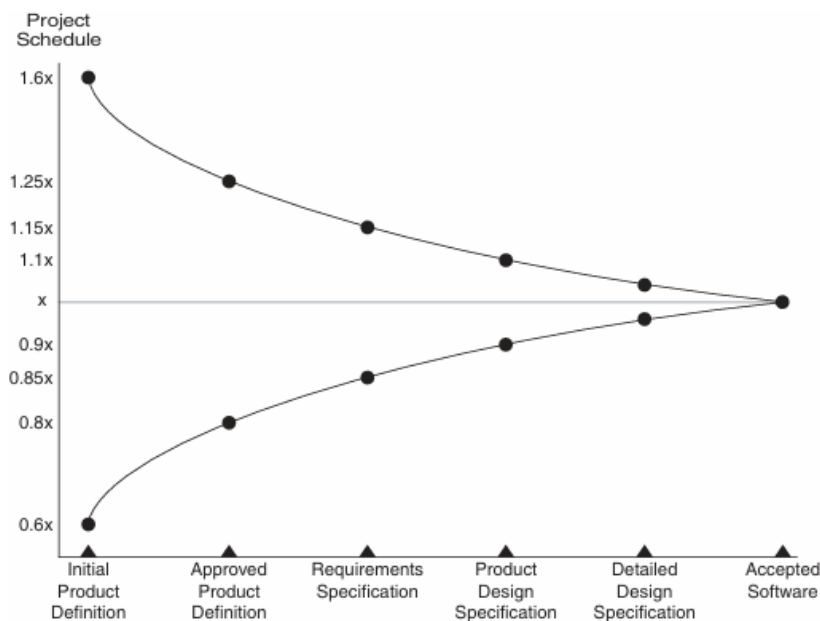
Libro: “*Agile Estimating and Planning*”

Capítulo 1: El propósito de la planeación

La estimación y la planeación son críticas para el éxito de cualquier proyecto de desarrollo de software de cualquier tamaño o consecuencia. Los planes guían nuestras decisiones de inversión.

Los equipos frecuentemente responden a esto yendo a uno de los dos extremos: no planean del todo, o ponen mucho esfuerzo en sus planes, que se convencen de que sus planes deben estar bien.

El cono de incertidumbre muestra que durante la fase de factibilidad de un proyecto, un calendario estima típicamente desde 60% a 160%. Esto es, que un proyecto que se estima que dure 20 semanas puede tomar desde 12 a 32 semanas. Luego de que los requerimientos son escritos, la estimación debería seguir entre +/- 15% en cualquier dirección. Entonces una estimación de 20 semanas significa que el trabajo puede tomar entre 17 a 23 semanas.



¿Por qué hacerlo?

Los planes y los calendarios pueden ser necesarios por una variedad de razones legítimas, como planear campañas de marketing, calendarizar las actividades de release de un producto, el entrenamiento interno de usuarios y así. Un buen proceso de planeamiento genera:

- Reducción de riesgo: la planeación incrementa la probabilidad del éxito de un proyecto proveyendo perspectivas dentro de los riesgos de proyecto. Algunos proyectos son tan riesgosos que son mejor no empezarlos una vez que aprendemos sobre esos riesgos. Otros proyectos pueden contener features cuyos riesgos pueden ser contenidos con una atención temprana.
- Reducción de incertidumbre: es crítico que este nuevo conocimiento sea aprendido y factorizado dentro de un proceso iterativo de planeamiento que es diseñado para ayudar al equipo a refinar su visión del producto.
- Soporte a una mejor decisión: entre decisiones sobre cómo o no empezar un proyecto, la estimación nos ayuda a estar seguros de que estamos trabajando en el proyecto más valorado posible. A veces, el perfil de personal de un proyecto puede ser más importante que su calendario. Estamos constantemente haciendo decisiones de compensación similares entre funcionalidad y esfuerzo, costo y tiempo.
- Establece confianza: La entrega frecuente y confiable de las funciones prometidas genera confianza entre los desarrolladores y proveedores de un producto y los clientes de ese producto. Estimaciones confiables permiten volver entrega responsable. Un cliente necesita estimaciones para realizar una priorización importante y decisiones de compensación
- Transmitir información: un plan transmite expectación y describe una posibilidad de qué puede venir en el curso del proyecto. Un plan no garantiza un exacto grupo de funciones en un día específico con un costo específico. Un plan comunica y establece un grupo de expectativas base.

¿Qué hace que un planeamiento sea ágil?

En un planeamiento ágil vamos a encontrar que:

- Esta centrado más en el planeamiento que en el plan
- Conlleva cambios
- Resulta en planes que son fácilmente cambiables
- Está repartido por todo el proyecto.

Capítulo 2: Por qué el planeamiento falla

El planeamiento esta más por actividad que por funcionalidad

Un problema crítico cercano al planeamiento es que se fijan más en la completitud de las actividades que en la entrega de funcionalidades. Un primer problema con las actividades basadas en el planeamiento es que los usuarios no valoran la completitud de las actividades. Las funcionalidades son la unidad de valor para el usuario.

Algunas de las razones por la que las actividades basadas en planeamiento derivan en el rebalsado de calendario incluye:

- Actividades que no terminan temprano: cuando un diagrama de Gantt muestra que una actividad se espera que tome 5 días, le da el permiso implícito al desarrollador a tomar todo ese tiempo para completarlo. Es natural en el humano que, cuando nos adelantamos a ese horario, llenemos el tiempo extra con otros trabajos que nosotros, pero quizá no los demás, valoramos.
- La impuntualidad se transmite en el calendario: un comienzo temprano requiere una combinación de cosas para que salga bien; un comienzo tardío puede ser causada por una cosa que salga mal. El problema se agrava porque ya hemos establecido que las actividades rara vez terminarán pronto. Esto significa que las actividades empezarán tarde y que el retraso se transmitirá al programa.
- Las actividades no son independientes: si las actividades de un esfuerzo de desarrollo no son independientes, las variaciones en el tiempo de finalización no se equilibrarán. El conocimiento real que podríamos ganar en una situación así es que cuando una actividad toma más tiempo que lo planeado, todas las actividades similares van tomar más tiempo de lo planeado igualmente.

El Multitasking produce retrasos lejanos

Multitasking cobra un precio horrible en la productividad. Se estudiaron los efectos del multitasking y se encontró que el tiempo que un individuo se toma para hacer trabajos valuados cae rápidamente cuando el individuo está trabajando en más de dos tareas.

Multitasking frecuentemente se vuelve un problema cuando el proyecto empieza a tener actividades que finalizan tarde. En ese punto, las dependencias entre actividades se vuelven crítica.

Multitasking se vuelve un problema en un proyecto tradicional planeado por dos razones principales. Primero, el trabajo se suele asignar con mucha antelación a su inicio, y es imposible asignarlo de forma eficiente con antelación. Segundo, se centra en lograr un alto nivel de utilización de todos los participantes en el proyecto, en lugar de mantener una holgura con la variabilidad inherente a las tareas típicas de un proyecto.

Las funcionalidades no son desarrolladas por prioridad

Una tercera razón por la que la planificación tradicional no conduce sistemáticamente a productos de alto valor es que el trabajo descrito en el plan no se prioriza en función de su valor para los usuarios y clientes. Esto significa que el trabajo es típicamente priorizado y calendarizado a conveniencia del equipo de desarrollo.

Ignoramos la incertidumbre

Ignoramos la incertidumbre sobre el producto y asumimos que el análisis inicial de requerimientos nos da una perfecta y completa especificación del producto. Asumimos que los usuarios no van a cambiar de mente, refinar sus opiniones, o que saldrán con nuevas necesidades durante el periodo cubierto por el plan.

La mejor forma de sobrellevar esta incertidumbre es iterar. Para reducir la incertidumbre acerca de cómo el producto debería ser, trabajar en iteraciones cortas, y mostrar software trabajando a los usuarios cada pocas semanas.

Las estimaciones se convierten en compromisos

Un problema con el que la planeación tradicional que puede aparecer es que el equipo de proyecto o los stakeholders igualen la estimación con compromiso. Una estimación es una probabilidad, y el compromiso no puede estar hecho con probabilidades. Los compromisos son citas.

Antes de hacer un compromiso, el equipo necesita verificar una variedad de factores y riesgos. Es importante que sea dada esta oportunidad, y que cada estimación no se convierta en un compromiso implícito.

Capítulo 3: Un acercamiento ágil

Los autores del Manifiesto Ágil escribieron que ellos valoran:

- Individuos e interacciones sobre procesos y herramientas: un equipo funcional de grandes individuos con herramientas mediocres siempre va a tener mejores resultados que un equipo disfuncional con individuos mediocres y grandes herramientas y procesos. Hay que tomar los puntos fuertes (y débiles) únicos de cada persona y aprovecharlas, en lugar de intentar homogeneizar a todo el mundo.
- Software funcionando sobre documentación comprensiva: los lleva a tener una versión estable e incrementalmente mejorada del producto al final de cada iteración. De este modo es posible recoger frecuentes reseñas sobre el producto y el proceso. Hay que estar seguros de que el equipo está siempre trabajando en las funcionalidades más valuada y aquellas que satisfacen las expectativas del usuario.
- Colaboración con los compradores sobre la negociación de contrato: porque los equipos ágiles les gustaría que todas las partes del proyecto trabajasen por los mismos objetivos. El juego es sorprendentemente divertido, y nos gustaría que los equipos de software y los clientes abordaran los proyectos con esta misma actitud de colaboración y objetivos compartidos. Sí, los contratos suelen ser necesarios, pero los términos y detalles de un contrato pueden influir mucho en la decisión de las partes de colaborar o competir.
- Responder al cambio que seguir un plan: porque le ponen mucho foco en dar el mayor valor posible al proyecto de los clientes y usuarios. Es inevitable que los usuarios vengan con nuevas ideas, y casi es inevitable que ellos van a decidir que algunas funcionalidades deseadas hoy, mañana serán prioridades bajas. Para un equipo ágil, el plan es una visión del futuro, pero muchas visiones son posibles.

Un acercamiento ágil a los proyectos

Las siguientes secciones abarcan algunas de las principales formas de trabajar de los equipos ágiles:

- Trabajan como uno solo: crítico para el éxito de un proyecto es que todos los participantes se vean como un equipo con un objetivo en común. Hay un número específico de roles en un equipo. Es justificable identificar y clarificar esos roles que juegan en la parte de estimación y planeación ágil:
 - Producto Owner: las principales funciones son asegurarse de que todos los miembros del equipo persiguen una visión común del proyecto, establecer prioridades para que siempre se trabaje en las funciones de mayor valor y tomar decisiones que permitan rentabilizar la inversión en el proyecto. Suele ser alguien del departamento de marketing o gestión de productos de la empresa.
 - Cliente: la persona que toma la decisión de iniciar el proyecto o comprar software. Es usualmente una parte representativa de un grupo o división. En cualquier caso, el cliente puede o no ser un usuario del software, lo que es, obviamente, otro rol importante.
 - Desarrolladores: cualquier persona que desarrolle el software.
 - Project Manager: se centra más en el liderazgo que en la gestión. En algunos proyectos ágiles, la persona que desempeña el papel de gestor del proyecto también desempeñará otra función, a menudo como desarrollador, pero ocasionalmente como propietario del producto.
- Trabajan en iteraciones cortas: las iteraciones están sujetas a plazos, lo que significa que terminan a tiempo aunque se eliminen funciones. Los plazos suelen ser muy cortos. La mayoría de equipos se decantan por una duración de iteración relativamente constante.
- Entregan algo en cada iteración: muchos equipos no entregarán los resultados de cada iteración a sus usuarios; el objetivo es simplemente que puedan hacerlo. Esto significa que los equipos progresan añadiendo una o más pequeñas características en cada iteración, pero que cada característica añadida está codificada, probada y es de calidad publicable. Dado que una sola iteración no suele proporcionar tiempo suficiente para completar una funcionalidad nueva que satisfaga los deseos del usuario o del cliente, se introduce el concepto más amplio de

versión. Una versión comprende una o más iteraciones que se basan unas en otras para completar un conjunto de funcionalidades relacionadas.

- Se enfocan en prioridades de negocio: en primer lugar, entregan las funciones en el orden especificado por el PO, que debe priorizar y combinar las funciones en una versión que optimice el rendimiento de la inversión de la organización en el proyecto. Para ello, se crea un plan de lanzamiento basado en las capacidades del equipo y en una lista priorizada de las nuevas funciones deseadas. Para que el PO tenga la máxima flexibilidad a la hora de establecer prioridades, las funciones deben redactarse de forma que se reduzcan al mínimo las dependencias técnicas entre ellas. En segundo lugar, los equipos ágiles se centran en completar y entregar características valoradas por el usuario en lugar de completar tareas aisladas. Una de las mejores formas de hacerlo es trabajar con US, que son una técnica ligera para expresar requisitos del software.
- Inspecciona y se adapta: el plan creado al inicio de cualquier proyecto no está garantizado que ocurra. Los equipos ágiles ven cada cambio como una oportunidad y una necesidad de actualizar el plan para reflejar mejor la realidad de la situación actual. Si el equipo aprendió algo que puede afectar el valor del plan, ellos ajustarán el plan. La posibilidad de modificar las prioridades entre iteraciones contribuye en gran medida a maximizar el rendimiento de la inversión en el proyecto

Múltiples niveles de planeamiento

Un proyecto corre peligro si su planificación se extiende mucho más allá del horizonte del planificador y no incluye tiempo para que éste levante la cabeza, mire el nuevo horizonte y haga ajustes. Es necesaria una elaboración progresiva del plan. Los equipos ágiles lo consiguen planificando en tres horizontes distintos: versión, iteración y día actual.



El objetivo de la planeación de “release” es determinar una respuesta adecuada a las preguntas sobre el alcance, el calendario y los recursos de un proyecto. Se realiza al inicio del proyecto, pero no es un esfuerzo aislado.

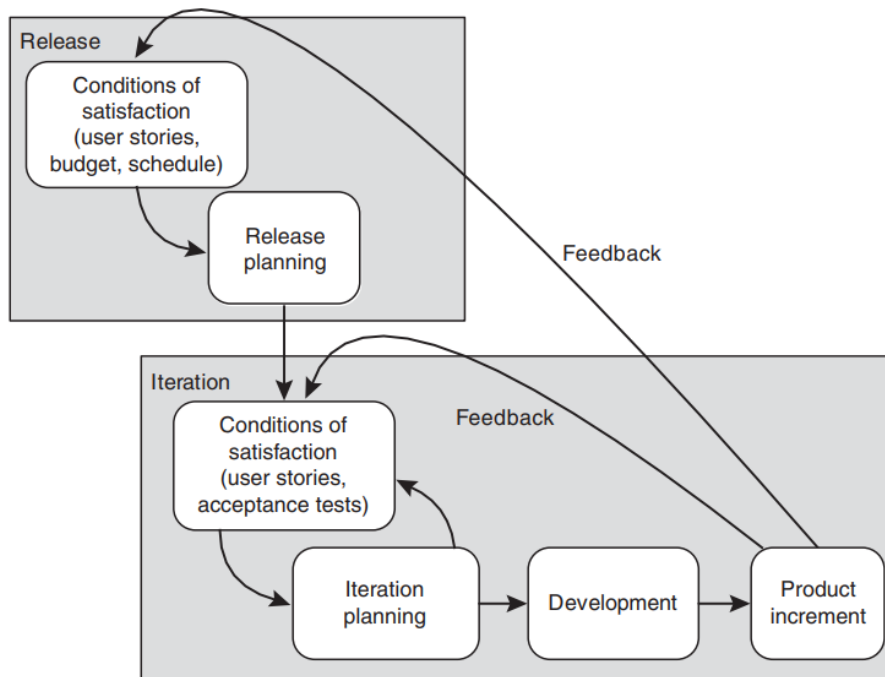
La planeación de la iteración, que es creada al inicio de cada iteración, el PO identifica el trabajo con más prioridad que el equipo debería direccionar en la nueva iteración.

La planeación del día es para coordinar trabajo y sincronizar el esfuerzo diario. Se concentran en planear tareas y en coordinar las actividades individuales que llevan a la completitud de una tarea.

La planificación del producto implica que el PO mire más allá del lanzamiento inmediato y planifique la evolución del producto o sistema lanzado. La planificación del portfolio implica la selección de los productos que mejor aplicarán una visión establecida a través de la planificación estratégica de una organización.

Condiciones de satisfacción

Al igual que la planificación de release, la planificación de iteraciones es iterativa. El PO y el equipo debaten diversas formas de cumplir mejor las condiciones de satisfacción de la iteración.



En esta figura se muestran los bucles de retroalimentación desde el incremento del nuevo producto resultante hasta los cuadros de condiciones de satisfacción al inicio de la planificación de la versión y la iteración. En función de su experiencia en el desarrollo del incremento de producto durante la iteración, el equipo puede haber adquirido conocimientos o experiencia que afecten a la planificación en uno o más de estos niveles. Del mismo modo, mostrar el incremento del producto a los usuarios existentes o probables puede generar nuevos conocimientos que provoquen cambios en los planes. Un equipo ágil incorporará estos cambios a sus planes en la medida en que conduzcan a un producto de mayor valor.

Capítulo 4: Estimando tamaño con SP

Las SP son relativas

Los SP son una unidad de peso para expresar el tamaño general de una US, funcionalidad u otra pieza de trabajo. Lo que importa es el valor relativo. Una US que tiene un 2 es dos veces más grandes que la US que tiene 1.

La estimación en SP es una amalgama de la cantidad de esfuerzo que supone desarrollar la característica, la complejidad de su desarrollo, el riesgo inherente a la misma, etc.

El primer enfoque consiste en seleccionar una historia que usted espera que sea una de las historias más pequeñas con las que va a trabajar y decir que esa historia se estima en un punto de historia. El segundo método consiste en seleccionar una historia que parezca mediana y asignarle un número intermedio dentro del intervalo que espera utilizar.

En un proyecto ágil no es raro empezar una iteración con requisitos incompletamente especificados, cuyos detalles se descubrirán durante la iteración. Sin embargo, hay que asociar una estimación a cada historia, incluso a las que están definidas de forma incompleta.

Velocidad

La velocidad es una medida del ritmo de progreso de un equipo. Se calcula sumando el número de puntos de historia asignados a cada historia de usuario que el equipo ha completado durante la iteración.

Si sumamos las estimaciones de puntos de historia de todas las características deseadas, obtendremos una estimación del tamaño total del proyecto. Si conocemos la velocidad del equipo, podemos dividir el tamaño por la velocidad para obtener una estimación del número de iteraciones.

La velocidad corrige errores de estimación

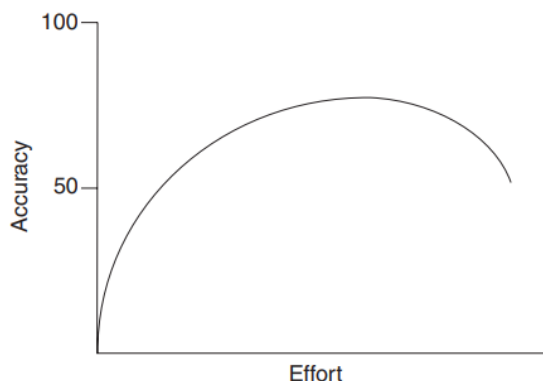
Lo bueno de un enfoque de la estimación basado en puntos es que los errores de planificación se autocorriguen gracias a la aplicación de la velocidad. Supongamos que un equipo estima que un proyecto incluirá 200 puntos de trabajo. Inicialmente creen que podrán completar veinticinco puntos por iteración, lo que significa que terminarán en ocho

iteraciones. Sin embargo, una vez iniciado el proyecto, la velocidad observada es de sólo veinte. Sin reestimar ningún trabajo, habrán identificado correctamente que el proyecto llevará diez iteraciones en lugar de ocho.

Lo bueno de esto es que la estimación en puntos de historia separa completamente la estimación del esfuerzo de la estimación de la duración. Por supuesto, el esfuerzo y el calendario están relacionados, pero separarlos permite estimar cada uno por separado. De hecho, ya ni siquiera se estima la duración de un proyecto, sino que se calcula o se obtiene.

Capítulo 6: Técnicas de estimación

por mucho esfuerzo que se invierta, la estimación nunca se sitúa en la parte superior del eje de precisión. Por mucho esfuerzo que se invierta en una estimación, una estimación sigue siendo una estimación. Ningún esfuerzo adicional hará que una estimación sea perfecta. A continuación, fíjese en el poco esfuerzo que se necesita para aumentar drásticamente la precisión con respecto a la línea de base.



Los equipos ágiles, sin embargo, optan por situarse más cerca de la izquierda en esta figura. Reconocen que no podemos eliminar la incertidumbre de las estimaciones, pero adoptan la idea de que los pequeños esfuerzos se recompensan con grandes ganancias. Aunque estén menos arriba en la escala precisión/esfuerzo, los equipos ágiles pueden producir planes más fiables porque suelen entregar pequeños incrementos de código totalmente funcional, probado e integrado.

US, Épicas y Temas

Aunque en general queremos estimar historias de usuario cuyo tamaño esté dentro de un orden de magnitud, no siempre puede ser así. Si queremos estimar todo dentro de un orden de magnitud, significaría escribir todas las historias a un nivel bastante fino. Para las características que no estamos seguros de querer (una estimación preliminar del coste es deseable antes de invertir demasiado en ellas) o para las características que pueden no suceder en un futuro próximo, a menudo es deseable escribir una historia de usuario mucho más grande. Una gran historia de usuario a veces se denomina épica.

Además, un conjunto de historias de usuario relacionadas puede combinarse (normalmente con un clip si se trabaja con tarjetas de notas) y tratarse como una única entidad para la estimación o la planificación de la versión. Este conjunto de historias de usuario se denomina tema. Una epopeya, por su propio tamaño, es a menudo un tema en sí mismo.

Es importante que se den cuenta de que las estimaciones de los temas y las epopeyas serán más inciertas que las estimaciones de las historias de usuario más específicas y pequeñas.

Derivando una estimación

Las tres técnicas más comunes de estimación son:

- **Opinión experta:** en una estimación basada en la opinión de expertos, se pregunta a un experto cuánto tiempo llevará algo o cuál será su magnitud. El experto se basa en su intuición y proporciona una estimación. Este enfoque es menos útil en proyectos ágiles que en proyectos tradicionales. El beneficio de esta técnica es que no lleva mucho tiempo.
- **Analogía:** cuando estimamos por analogía, el estimador compara la US que está siendo estimada con una u otras más US. Al estimar de este modo, no se comparan todas las historias con una única línea de base o referencia universal. En lugar de eso, lo que se pretende es estimar cada nueva historia comparándola con una serie de historias ya estimadas. Es lo que se denomina triangulación.

- Desagregación: se refiere a dividir una US o funcionalidad en más pequeñas y manejables partes. De cualquier forma, necesitamos ser cuidadosos en no ir muy lejos. No sólo aumenta la probabilidad de olvidar una tarea si la desagregamos demasiado, sino que sumar las estimaciones de muchas tareas pequeñas también genera problemas

Planning Poker

Combina todas las técnicas de antes en una forma divertida de estimar que resulta en rápidas y creíbles estimaciones. Los participantes de esto incluyen a todos los desarrolladores del equipo. En un proyecto ágil, no suelen ser más de diez personas. Si es así, lo mejor suele ser dividirlo en dos equipos. Así, cada equipo podrá estimar de forma independiente, lo que reducirá el tamaño. El PO participa en esto pero no estima. De cualquier forma, tomarse mucho tiempo en las discusiones lleva al equipo a acercarse al gráfico del principio de esfuerzo/asertividad.

Por qué funciona PP

En primer lugar, el póquer de planificación reúne múltiples opiniones de expertos para realizar la estimación. Dado que estos expertos forman un equipo interfuncional de todas las disciplinas de un proyecto de software, están mejor preparados que nadie para la tarea de estimación.

En segundo lugar, durante el póquer de planificación se entabla un animado diálogo y los estimadores son interpelados por sus compañeros para que justifiquen sus estimaciones. Se ha comprobado que esto mejora la precisión de la estimación, sobre todo en elementos con gran incertidumbre. También se ha demostrado que cuando se pide a los participantes que justifiquen sus estimaciones, éstas compensan mejor la falta de información.

En tercer lugar, los estudios han demostrado que promediar las estimaciones individuales conduce a mejores resultados, al igual que las discusiones en grupo sobre las estimaciones. La discusión en grupo es la base del póquer de planificación, y esas discusiones llevan a promediar las estimaciones individuales.