

Estimativas Ágeis com Planning Poker

Autor: Dairton Bassi

Fonte: Revista Engenharia de software Ano 1 – Edição 9

Criar boas estimativas nunca foi uma tarefa simples para a indústria de software. Ainda hoje, essa é uma de suas principais dificuldades, pois uma estimativa é basicamente uma previsão e os seres humanos tendem a ser otimistas ao fazer previsões. Esse otimismo faz com que os prazos estimados sejam, na maioria das vezes, menores do que o necessário para produzir o software.

Ao estimar, é comum o desenvolvedor menosprezar ou desconsiderar dificuldades técnicas ou distrações do dia-a-dia. Além disso, a natural falta de conhecimento inicial sobre o que será desenvolvido colabora para que as estimativas não sejam precisas.

Para chegar a estimativas, há diversas técnicas usadas pela indústria de software. Neste artigo veremos uma que é compatível com metodologias ágeis de desenvolvimento de software: o Planning Poker.

Métodos Ágeis e Estimativas

O primeiro passo para resolver um problema é tomar consciência de que ele existe. Metodologias Ágeis aceitam a existência de diversos fatores que dificultam a criação de previsões precisas e duradouras.

Equipes ágeis baseiam-se na comunicação e na transparência. Ao invés de tratar suas estimativas como fatos, admitem que existe uma incerteza associada a ela e evidenciam isso para que o cliente e outros envolvidos também tomem ciência do grau de dificuldade de cada funcionalidade. Estimativas de longo prazo e funcionalidades complexas geralmente possuem graus maiores de incerteza associados. À medida que o tempo passa e o conhecimento sobre o assunto aumenta, as estimativas podem ser melhoradas considerando mais detalhes do projeto e com probabilidades mais altas de sucesso.

Estimativas de tamanho e estimativas de duração

Os inúmeros fatores que contribuem para agregar instabilidade ao projeto também colaboram para invalidar as estimativas de duração do projeto. Em projetos onde há muitos fatores de instabilidade, as estimativas tendem a durar pouco tempo pois mudanças no projeto contribuem para que elas percam rapidamente a sua validade. Para ter as estimativas atualizadas com frequência, mas evitar que elas tenham que ser refeitas constantemente, é importante separar as estimativas de tamanho das estimativas de duração. As estimativas de tamanho são medidas do volume de trabalho. As estimativas de duração estão associadas à quantidade de tempo necessária para executar esse trabalho.

Estimativas de tamanho requerem a compreensão das características da funcionalidade que será implementada. A grande vantagem dessas estimativas é que uma vez feitas, não se desatualizam, pois o tamanho da tarefa é independente do tempo disponível para realizá-la e da quantidade de pessoas na equipe.

Estimativas de duração são obtidas a partir das estimativas de tamanho e da velocidade da equipe de desenvolvimento.

Estimar a duração corresponde precisamente em dizer quanto tempo a equipe levará para concluir o trabalho. Para a mesma tarefa, o tempo para a sua realização pode variar, por exemplo, conforme o tamanho da equipe e a disponibilidade de seus membros, pois esses fatores influem na velocidade de desenvolvimento. Quando a velocidade da equipe muda durante o projeto, as estimativas de duração podem ser facilmente recalculadas com base nas estimativas de tamanho.

A velocidade da equipe é a quantidade de tarefas que ela consegue concluir em um período fixo, por exemplo, a cada semana, a cada quinzena ou a cada iteração do projeto.

A estimativa de duração diz o tempo necessário para a conclusão da tarefa considerando uma determinada velocidade. Para obtê-la basta dividir a estimativa de tamanho pela velocidade da equipe:

$$\textit{Estimativa_de_duração} = \textit{Estimativa_de_tamanho} / \textit{velocidade}$$

Para entender a diferença entre as estimativas de tamanho e de duração, vamos, por exemplo, considerar uma tarefa simples. Suponha que a tarefa é digitar um texto a partir de um áudio (transcrição). Para isso, faremos a estimativa de tamanho, depois verificaremos a velocidade e, finalmente, obteremos a duração. Para estimar o tamanho, precisamos considerar o volume de trabalho que a digitação requer. Meu palpite é de que esse áudio, após digitado, gera um texto de três páginas. Essa foi a minha estimativa de tamanho.

Agora poderemos chegar a uma estimativa de duração se considerarmos a minha velocidade de digitação.

Digamos que gasto aproximadamente 15 minutos para digitar cada página. Esta é a minha velocidade: 1 página a cada 15 minutos, portanto, a estimativa de duração é de 45 minutos para concluir as três páginas. Porém, se enquanto digito, estiver assistindo televisão, minha velocidade cai para 1 página a cada 20 minutos, logo, a estimativa de duração será de 60 minutos para terminar a tarefa. Se meu colega de equipe de digitação for fazer o trabalho ao invés de mim, ele digita uma página em apenas 10 minutos e, portanto, a estimativa de tempo cai para 30 minutos.

Neste exemplo, a estimativa de tamanho precisou ser feita apenas uma vez. Depois, ela serviu de base para chegar a diversas estimativas de duração, conforme a dedicação dispensada à tarefa. Com essa separação entre volume de trabalho e tempo de execução, a estimativa de tamanho pode ser usada para estimar a duração, mesmo se a tarefa for delegada para outros executores.

Técnicas para Estimar

Existem diversas maneiras de mensurar o desenvolvimento de software. Na prática, ao invés de tentar descobrir ou calcular, as melhores estimativas vêm de alguém que já tenha feito um trabalho igual ou equivalente ao que está sendo estimado. Se esta pessoa domina o assunto, podemos chamá-la de especialista e ela será a melhor pessoa para oferecer opiniões sobre o trabalho. Portanto, delegar a tarefa a um especialista é uma ótima maneira de obter estimativas precisas, pois ele tem experiência no assunto e já sabe como fazer o trabalho.

Outra forma de chegar a estimativas é através da divisão e conquista. Para dimensionar grandes funcionalidades ou sistemas inteiros, dividi-los em partes torna mais fácil o seu dimensionamento. Cada uma delas pode ser estimada por analogia comparando-a com implementações já realizadas. Depois, é importante considerar que as partes precisarão ser integradas e isso também irá requerer esforço de implementação.

Contudo, os sistemas atuais tornaram-se tão complexos que para reunir todas as competências necessárias para criá-los, é preciso fazer uso de vários tipos de especialistas. Programadores que conheçam determinados frameworks, DBAs e designers são algumas das especialidades necessárias em muitos projetos. Porém, nem sempre há especialistas de todas as áreas à disposição do projeto, ou então, quem faz as estimativas não é quem realizará a implementação. Essas situações dificultam a obtenção de planos factíveis e realistas.

A técnica que apresentamos a seguir é o Planning Poker, um modelo de obtenção de estimativas que reúne a opinião dos principais envolvidos com o projeto. Na maioria das vezes eles não são especialistas em todas as áreas, mas serão eles os que farão a implementação, então é importante que isso seja considerado.

Para chegar a estimativas realistas e razoáveis, as próprias pessoas que participarão da implementação usam a estratégia de divisão e conquista junto com o dimensionamento por analogia.

Planning Poker

Planning poker foi criado por James Grenning em 2002, mas foi de Mike Cohn a contribuição decisiva para difundir essa técnica entre os praticantes de métodos ágeis. O Planning Poker usa o conhecimento dos desenvolvedores para chegar às estimativas. As suas regras estimulam a interação entre os membros da equipe e permitem que todos expressem as suas opiniões, ao mesmo tempo em que participam do planejamento do projeto e aumentam o seu entendimento sobre o sistema que irão desenvolver. Além de obter estimativas para a implementação, essas características colaboram para que a equipe melhore o seu desempenho e atinja os objetivos do projeto.

Para começar a usar Planning poker, é preciso definir alguns conceitos e fazer alguns preparativos, conforme veremos a seguir.

Defina uma escala

Separando estimativas de tamanho e duração, isolamos a variável tempo e nos concentramos em identificar o volume de trabalho durante o processo de mensuração.

Fazer estimativas de tamanho para software não é trivial, pois ele não é tangível, portanto, para medi-lo também usaremos uma medida de pontos abstratos que representam a quantidade de esforço para produzir o software desejado. Para atribuir esses pontos, é importante definir os valores de uma escala. A partir dela, cada funcionalidade receberá uma quantidade de pontos que representará a estimativa do seu tamanho.

Boas escalas podem ser criadas usando números de Fibonacci (1, 2, 3, 5, 8, 13,...), potências de 2 (1, 2, 4, 8, 16, 32,...) ou (1, 5, 10, 20, 40, 80,...), pois elas tem a característica de absorver parte do erro associado às estimativas de funcionalidades grandes.

Opcionalmente, a equipe pode incluir o 0 (zero) na escala para estimar tarefas triviais. Depois, muitas tarefas triviais podem ser agrupadas e receber tamanho 1, afinal, mesmo sendo triviais, tomarão algum tempo para serem executadas.

A escala usada para estimar pode utilizar pontos ou dias ideais. Os pontos são medidas abstratas com tamanho definido pela equipe.

Os pontos fornecem estimativas de tamanho, ou seja, de volume de trabalho. A partir deles podemos obter estimativas de duração, como por exemplo, dias de trabalho, que prevêm a quantidade de tempo necessária para cumprir a estimativa de tamanho. Contudo, é importante que medidas de tamanho e duração fiquem separadas, pois a velocidade da equipe pode variar durante o projeto. Isso influencia as estimativas de duração, que são as que os clientes estão mais interessados, e obriga a equipe a refazê-las. Porém, as estimativas de tamanho não mudarão. Estas serão usadas para chegar às novas previsões de duração. Quando a velocidade de desenvolvimento mudar, basta identificar a nova velocidade e derivar as novas estimativas de duração.

Para chegar às estimativas de tamanho, a estratégia da comparação é o melhor caminho, pois a mente humana consegue lidar melhor com medidas relativas do que com medidas absolutas. Isso significa que é muito mais fácil dizer, por exemplo, quantas vezes uma sala é maior do que outra do que determinar a sua área em metros quadrados.

Para estimar desenvolvimento de software por comparação, a equipe identifica uma funcionalidade pequena, cujo esforço para implementá-la seja conhecido e a define como a sua base para comparações. Usando, por exemplo, a escala {1, 2, 4, 8, 16, 32}, atribui a essa funcionalidade o valor 2. Depois, o valor das próximas funcionalidades é obtido comparando cada uma delas com a funcionalidade base de tamanho 2. Funcionalidades com o dobro do tamanho receberão o valor 4, funcionalidades quatro vezes maior, o valor 8, e assim por diante.

Perceba que nas escalas sugeridas, os valores não são sequenciais, o que obriga os desenvolvedores a usar apenas os valores da escala. Isso ajuda a fazer com que as estimativas transmitam seu significado real. Quanto maior é a funcionalidade, menor a precisão da estimativa. A fixação de valores evita que a estimativa seja confundida com a realidade. Se você usar números sequenciais e estimar duas funcionalidades grandes, respectivamente, com os valores 78 e 81, você transmitirá uma falsa sensação de precisão. Agora, se você usa uma escala com os valores 10, 20, 40 e 80 e diz que as duas funcionalidades têm valor 80, fica implícito que o tamanho real pode ser um pouco mais ou um pouco menos do que 80.

Com os valores fixados, os desenvolvedores usarão naturalmente aquele que estiver mais próximo da sua percepção, forçando-os a fazer um arredondamento em suas previsões. Esse comportamento é natural e não representa um problema porque algumas vezes o arredondamento aumentará o valor da estimativa e, em outras, o diminuirá, de forma que quando um conjunto de funcionalidades for considerado, essas diferenças tendem a se equilibrar.

Materiais e preparativos

Para estimar com Planning Poker, é preciso reunir todos os envolvidos com a implementação e preparar o material. Cada um dos participantes deve ter um conjunto de cartas nas quais cada uma contém um dos valores da escala escolhida.

As funcionalidades já devem ter sido identificadas e escritas no formato de cartões de histórias de XP (em XP, cada funcionalidade é descrita com texto em um cartão de papel. Quando necessário, pode-se fazer desenhos ou incluir exemplos para facilitar a compreensão. O tamanho do cartão é variável, em torno de 10cm altura e 15cm de largura, com espaço para algumas linhas de texto) e os participantes já devem tê-las lido e tirado suas dúvidas com o cliente ou o analista que as escreveu.

Quando a história é muito complexa, ela deve ser dividida em histórias menores para que seja possível classificá-la na escala preestabelecida. Ao criar histórias pequenas, o trabalho de dimensionamento fica mais simples, pois é mais fácil estimar várias tarefas pequenas, uma a uma, do que medir apenas uma grande tarefa.

Chegando às Estimativas

Um dos desenvolvedores lê em voz alta cartão a cartão. Após a leitura de cada um, todos têm a chance de tirar novas dúvidas ou fazer observações que ajudem a equipe a compreender os requisitos. Quando todos estiverem certos de que entenderam a funcionalidade, cada participante escolhe em seu conjunto de cartas aquela com o valor de sua estimativa.

Quando todos estiverem prontos, as cartas escolhidas são reveladas ao grupo. Como cada um deu a sua opinião, é comum que nem todos apresentem o mesmo valor. Quando isso acontece, **os participantes que**

apresentaram a maior e a menor estimativas explicam porque as fizeram. Baseados nas explicações, todos repensam suas opiniões e apresentam as estimativas revisadas. Se elas ainda não forem iguais, os novos estimadores do maior e do menor valor apresentam seus argumentos para a equipe. Este ciclo se repete até que a equipe chegue a um consenso sobre o valor da estimativa.

As explicações fornecidas pelos que fizeram a maior e a menor estimativas ajudam a equipe a compreender o problema e a encontrar melhores soluções. O desenvolvedor que sugeriu o valor mais alto pode ter percebido alguma dificuldade que os demais não observaram. Ao justificar a sua estimativa, ele ajuda o resto da equipe a refletir sobre o problema sob um novo ponto de vista. Da mesma forma, o desenvolvedor que faz a estimativa mais baixa expõe seus argumentos. Ele pode ter encontrado uma solução simples e eficaz que resolve o problema. Ao explicá-la para a equipe, todos aprenderão uma forma eficiente de implementar aquela funcionalidade. A experiência com esse tipo de técnica tem mostrado que o mais comum é que sejam realizados poucos turnos de explicações e que raramente é preciso mais do que duas revisões nas estimativas para que todos os valores convirjam.

Quando a equipe concorda com um tamanho para a funcionalidade, o valor é escrito no cartão para ser usado mais tarde na montagem das entregas e iterações do projeto e para calcular a velocidade da equipe e as estimativas de duração.

Se a equipe tiver à sua disposição uma ampulheta de aproximadamente dois minutos, ela poderá ser usada para controlar o tempo de argumentação. Idealmente a obtenção da estimativa de cada cartão não deve durar mais do que o um giro da ampulheta. Isso ajuda a equipe a concentrar-se em estimativas e evita que ela perca o foco com discussões sobre detalhes específicos da implementação.