Estimativas de Projetos Ágeis de Software

Autora: Dandara Pereira Aranha

Nos últimos anos, os métodos ágeis de desenvolvimento de software ganharam muita atenção no campo da engenharia de software. São baseados em uma "filosofia" descrita no Manifesto Ágil publicado no ano de 2001. Surgiram como alternativa aos chamados métodos tradicionais de desenvolvimento de software e da necessidade de atender às crescentes pressões das organizações por inovação, produtividade (prazos cada vez mais curtos), flexibilidade e melhoria no desempenho/qualidade dos projetos de desenvolvimento de Software.

Independente da metodologia de desenvolvimento de software que está sendo utilizada, as estimativas devem ser uma das principais atividades do planejamento de software. "*Não se pode controlar aquilo que não se pode medir"* afirmava Peter Drucker. Ou seja, o desconhecimento de quantitativos como o prazo de duração do projeto, alocação de recursos e o esforço a ser empregado em um projeto é preocupante. O desconhecimento desses quantitativos pode levar ao completo caos e que seu projeto não alcance o sucesso desejado. Um estudo da QSM sobre desempenho ágil mostrou consistentemente que é um bom plano que define o sucesso no desenvolvimento de software, e não a metodologia de desenvolvimento.

Muitos praticantes das metodologias ágeis são adeptos ao movimento #NoEstimate pois acreditam que não existe valor em estimar se você estiver trabalhando com pequenos incrementos de software, sprints curtas para aprimorar seu backlog. Entretanto, algumas pesquisas como a realizada pela CA Technologies, referência em serviços ágeis, mostram que as entregas realizadas por times que tem o hábito de realizar estimativas são mais rápidas do que as entregas dos times que não realizam nenhuma estimativa. O movimento #NoEstimate pode ser uma solução em alguns casos, por exemplo se seu time já trabalhar com pequenas tarefas com um timebox estabelecido (2, 3 dias), ou no contexto das estimativas operacionais com janelas de 2 a 4 semanas. Mas não pode ser aplicado de forma absoluta em qualquer caso.

As metodologias ágeis propõem um conjunto grande de técnicas para estimar e planejar projetos, principalmente em termos de modelos não baseados em algoritmos. A maioria das técnicas de estimativa ágil concentram-se no uso de "Histórias de Usuários".

É importante lembrar que não existe a melhor técnica, existe a que pode ser mais adequada ao seu propósito, ao seu time e ao projeto/produto que você está trabalhando. Só devemos estimar se isso for necessário para algo que traga valor, e devemos sempre associar a estimativa a um propósito.

Dentre as técnicas de estimativas ágeis mais conhecidas temos:

Planning Poker

É uma variação da técnica Delphi, que estima o esforço exigido para concluir tarefas e não projetos inteiros. (COHN, 2005) e (HIGHSMITH, 2009) propõem essa técnica para realizar estimativas no nível do projeto. Funciona da seguinte maneira:

Cada membro da equipe tem um baralho baseado em uma sequência Fibonacci modificada com algumas cartas adicionais, representando os pontos a serem atribuídos a cada história de usuário. Um a um, o representante do cliente explica os requisitos descritos na história de usuário.

Uma vez que a história é descrita, os membros da equipe podem fazer algumas perguntas para esclarecer seu escopo. Depois disso, cada membro, ao mesmo tempo, mostra um cartão do seu conjunto com a estimativa em pontos. Se elas são todas iguais, a história recebe a estimativa, se não, os membros que propõem a estimativa mais alta e a mais baixa explicam seus pontos de vista e novas rodadas são jogadas até chegar a um consenso.

Então, a equipe prevê a velocidade, que é o número de pontos que eles podem entregar em uma iteração. Ele pode ser obtido por meio de dados históricos, uma iteração de teste ou apenas uma suposição. Com esse dado pode-se estabelecer o comprimento da iteração. O número de iterações é obtido pela divisão do número total de pontos pela velocidade. Da mesma forma, a duração do projeto é calculada multiplicando o número de iterações pelo tamanho.

Não é a mais adequada para estimativas de prazo de duração de projeto, ou para alocação de recursos no nível de projeto (ou programa) pois é uma medida de complexidade, subjetiva e particular a cada time. Entretanto, realizar estimativas de forma colaborativa entre todos os membros do time é muito válido para aumentar o comprometimento na entrega, tirar dúvidas, trocar ideias, esclarecer requisitos e perceber riscos que você não perceberia sozinho. Assim, evita-se a tendência das pessoas e organizações a subestimar o tempo que vai precisar para completar uma tarefa, mesmo quando têm experiência em tarefas similares.

Dia Ideal

(COHN, 2005) afirma que um Dia Ideal corresponde à quantidade de trabalho que um profissional de nível sênior, com fluência nas tecnologias e ferramentas envolvidas consegue realizar em um dia de trabalho sem interrupções. Trata-se de uma estimativa empírica, executada por especialistas para desenvolvimento com base em exploração adaptativa (ALVES; FONSECA, 2008).

A velocidade é calculada a partir do número de horas que a equipe gasta para implementar um trabalho equivalente a um Dia Ideal.

Caso o item passe um dia de trabalho, é sugerido decompor esse item em itens menores que se consiga implementar em apenas um dia.

É mais indicada para equipes que ainda não possuem experiência em realizar estimativas, pois é fácil de ser entendida.

Entretanto, ao medir o tamanho de uma atividade em **dias ideais**, ainda estamos o associando à noção de tempo, o que nem sempre é uma boa estratégia. Diversos fatores podem influenciar esse tipo de estimativa como: evolução da experiência do time, conhecimentos adquiridos, mudanças de *frameworks*, entre outros. O mesmo time agora pode entender que uma funcionalidade similar a uma anterior pode ser concluída em menos dias. A base de estimativa muda, e assim o histórico fica prejudicado.

The planning game

O objetivo do Planning Game é maximizar o valor agregado e minimizar o custo do sistema a ser desenvolvido. Essa técnica, proposta pela eXtreme Programming, pressupõe que o cliente é conhecedor do negócio, sabendo relacionar e priorizar os requisitos do sistema de acordo com seu valor agregado. Por outro lado, o Programador é conhecedor das questões técnicas, sabendo estimar o custo de implementação desses requisitos (SHORE; WADEN, 2008).

Suas duas regras básicas são: (i) o cliente prioriza e (ii) o programador estima.

O jogo começa com o cliente relacionando os requisitos em cartões. Em seguida, esses cartões são colocados em ordem, de acordo com o valor agregado observado pelo cliente. Por fim, o programador deve estimar o custo de implementação de cada requisito priorizado. Esses passos são repetidos até que todos os requisitos tenham sido priorizados e estimados.

Durante o processo, desenvolvedores e clientes interagem para resolver dúvidas sobre prioridades e estimativas. Inicialmente, esta técnica foi utilizada para avaliar o trabalho em uma iteração, mas também pode ser usada para realizar um planejamento de uma release completa.

Novamente a maior vantagem está na colaboração e interação, nesse caso entre cliente e time de desenvolvimento. Contudo, o cliente pode representar várias partes interessadas, que podem possuir diferentes interesses, e assim pode não ser tão fácil chegar a um acordo com relação ao valor agregado de um determinado requisito. Da mesma maneira, o conjunto de programadores pode divergir nas estimativas de custo dos requisitos e, nesses casos, eles devem chegar a um consenso.

Métricas de Tamanho

Métricas de tamanho nasceram com o objetivo de estimar esforço e prazos associados ao desenvolvimento de software segundo (PUTMAN, 1992). É uma das primeiras e principais atividades relacionadas às estimativas de software. E de acordo com a pesquisa realizada por (USMAN; MENDES e BORSTLER, 2015), o Story Points e o Ponto de Função são as métricas de tamanho mais utilizadas em projetos ágeis.

O Story Point é uma medida relativa onde o time decide o quanto vale um ponto e, baseado nesse tamanho, determina quantos pontos cada atividade possui.

Pontos de Função oferecem uma medida padronizada do tamanho funcional dos requisitos do usuário, independentemente de qualquer aspecto de implementação dos mesmos.

Embora a técnica de pontos de função tenha surgido bem antes dos Story Points, ambos pode ser utilizados de maneira similar em projetos ágeis. Pode-se estimar as histórias de usuário, as sprints e o product backlog em pontos de função. A ideia de velocidade (ou produtividade) também pode ser derivada: PF/sprint. Com essas informações pode-se então definir o número de sprints de um release. A velocidade inicial pode ser mais facilmente obtida com os pontos de função, porque é uma medida padrão entre os projetos.

A técnica de pontos de função fornece uma medida objetiva e comparável, e assim permite uma visão Tática e Estratégica sobre o desenvolvimento de software, análise de viabilidade, Benchmarking e pode ajuda a entender as variações de produtividade e crescimento de escopo entre projetos. É mais adequada quando se precisa um fator de normalização para comparação de software e para medir a funcionalidade solicitada pelo usuário, antes do projeto de software, de forma a estimar seu tamanho e seu custo.

Conclusão

Os projetos de desenvolvimento estão se tornando maiores e mais complexos, desafiando as equipes a conhecer e a determinar prazos de entrega realistas. Assim, é imprescindível que os times realizem estimativas que os permitam prever esforço, prazos e custos de seus projetos.

É importante dizer que as estimativas não devem ser difíceis, onerosas ou ineficientes. Feitas da maneira correta poderá ser muito efetiva e ajudar a entregar valor a suas organizações. Métodos e métricas para estimativas em sua maioria são aplicadas à um determinado contexto de desenvolvimento ágil com adaptações a fim de se adequarem ao projeto em questão. Ou seja, devem sofrer adaptações para se adequarem ao contexto do seu projeto, pois cada um deles tem características próprias que podem influenciar no resultado das estimativas.

Referências 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 Time, 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