

Estimando Projetos de Software Utilizando

User Stories e Planning Poker

Marcelo Bataglini Barbosa

Resumo

A estimativa de tempo em projetos de software sempre foi um desafio para engenharia de software e a gestão de projetos, pelo fato de que a estimativa tem grande chance de não atingir com precisão todo tempo que será gasto no projeto, isto ocorre devido a aspectos como tamanho funcional, complexidade, linguagem de implementação, arquitetura, relacionamento com usuários e modelo de gestão. As metodologias de desenvolvimento ágil de projetos de software propõem técnicas para maximização do resultado e a redução do tempo gasto com a estimativa, visando obter um prazo mais realista para entrega do projeto. Este estudo teve como principal objetivo pesquisar e entender tais técnicas e também preencher uma lacuna da pouca literatura existente sobre o assunto em língua portuguesa.

Palavras-chave: Estimativa; Metodologias Ágeis; *Planning Poker*, *Scrum*; *User Stories*; XP (*Extreme Programming*);

Introdução

É comum em projetos de software a apresentação de uma estimativa de prazo, as metodologias ágeis SCRUM e XP propõem o uso de *user stories* e *planning poker* para realizar a estimativa ágil do projeto de software. Esse tipo estimava possibilita não só a estimativa do projeto como um todo, mas também planejar as iterações que o projeto terá e também priorizar os requisitos mais importantes para os usuários. Em um projeto ágil uma *user storie* substitui uma descrição formal de requisitos ou então um caso de uso. Já o *planning poker* é uma reunião realizada entre os membros do time para estimar as *user stories* do projeto. A seguir serão apresentados os conceitos de *user stories* e *planning poker* bem como a sua utilização para realizar a estimativa ágil de projetos de software.

Marcelo Bataglini Barbosa, bacharel em ciências da computação e pós-graduando em engenharia de software com ênfase em SOA (*Service Oriented Architecture*).
mglini@terra.com.br

Revisão bibliográfica

Mike Conh em seus livros *User Stories Applied* e *Agile Estimating and Planning* demonstra como a estimativa de projetos de software pode ser feita de forma colaborativa, ágil e precisa. Este estudo tem o foco em analisar as técnicas apresentadas por Conh e também outros artigos da comunidade ágil sobre o assunto estimativa ágil de projetos de software.

1. User Stories

Segundo Cohn (2004) uma *user storie* descreve uma funcionalidade que têm valor para o usuário ou para quem está comprando o sistema ou software.

Uma *user storie* normalmente é escrita a mão em pequenos cartões de papel, também existem softwares para esta finalidade, porém o uso é recomendado apenas para equipes distribuídas, ela é composta por três aspectos: uma descrição, notas que servem para detalhar a *user storie* e testes de cobertura que podem ser usados para determinar quando uma *user storie* esta completa.

As *user stories* se diferem de descrições de requisitos, pois os requisitos informam o que o sistema deve fazer. Já uma *user storie* representa objetivos dos usuários dentro do sistema, elas também se diferem de casos de uso, apesar de ambos terem o objetivo de descreverem funcionalidades que entregarem valor ao negocio, as *user stories* tem o seu escopo mais reduzido e são colocadas restrições no seu tamanho. As técnicas para obtenção de *user stories* são praticamente as mesmas para obtenção de requisitos: entrevistas com usuários, questionários, observação e *workshops*.

A figura 1 demonstra um exemplo de uma *user storie*.

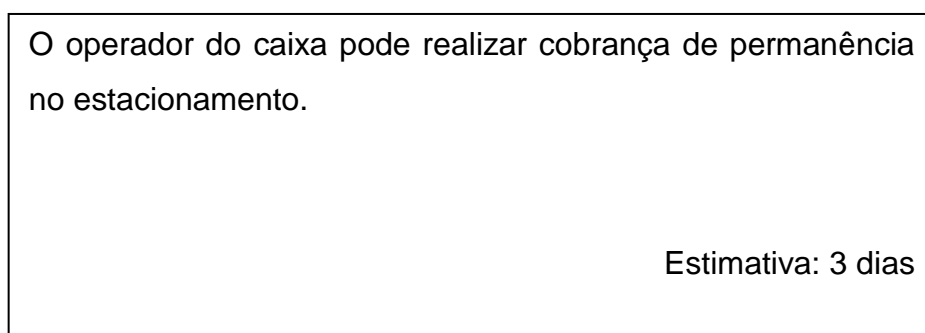


Figura 1 – Exemplo de uma *user storie*

Uma *user storie* bem escrita deve focar seis atributos principais conhecidos pelo acrônimo INVEST: independente, negociável, valiosa, estimável, pequena e testável.

1.1.Independente

Ao se escrever uma *user storie* deve-se ter o cuidado para não criar dependências entres elas, isto pode causar problemas no planejamento e priorização.

Supondo o desenvolvimento de um site de comercio eletrônico, é necessário escrever as *user stories* de como os clientes do site podem realizar pagamentos com cartão de crédito. As *user stories* poderiam ser escritas desta forma:

- O cliente pode efetuar o pagamento com um cartão Visa.
- O cliente pode efetuar o pagamento com um cartão Master Card.
- O cliente pode efetuar o pagamento com um cartão Aura.

Supondo ainda que os desenvolvedores deste projeto estimassem que para desenvolver o pagamento com o primeiro cartão (independente de qual fosse) seriam necessários quatro dias de trabalho e para o segundo e terceiro dois dias. Desta forma não seria possível saber para qual cartão de crédito é necessário quatro dias de trabalho. O ideal nesta situação é criar uma única *user storie* que represente o pagamento através de cartão de crédito como um todo e os tipos aceitos podem ser lembrados como notas no cartão, como na figura 2.

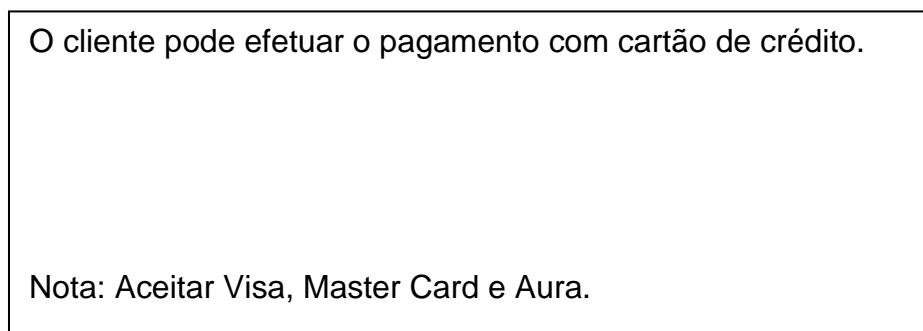


Figura 2 – User storie representando o pagamento com cartão de crédito

Com a *user storie* escrita desta forma fica mais fácil e independente estimar que o pagamento com cartão de crédito irá levar seis dias para ser desenvolvido.

1.2. Negociável

As *user stories* não são contratos ou requisitos que o software deve implementar, elas são pequenas descrições de funcionalidades, como lembretes, os detalhes devem ser negociados e detalhados entre o cliente e o time de desenvolvimento em conversas sobre a funcionalidade. Uma *user storie* deve conter informação suficiente para que se entenda o que ela representa, porém muitos detalhes podem causar a falsa impressão que toda informação incluída é precisa e completa, perdendo assim as vantagens de se combinar lembretes escritos com a comunicação face a face entre os envolvidos no projeto.

1.3. Valiosa

As *user stories* devem ser focadas em recursos não em tarefas, devem ser escritas na linguagem dos negócios, para que as pessoas de negocio possam entendê-las e priorizá-las, devem representar algo que tenha valor para usuário ou comprador do sistema. Segundo Cohn (2004), a melhor forma de se obter *user stories* valiosas e que o cliente escreva suas próprias *user stories*, porém o cliente pode se sentir desconfortável para realizar este trabalho, neste caso alguma pessoa do time deve se reunir com o cliente ou usuário e captá-las.

1.4. Estimável

As *user stories* devem ser estimadas pelo time de desenvolvimento, existem três aspectos principais que podem influenciar negativamente na estimativa ou não permitir que ela seja estimada: os desenvolvedores não possuem o conhecimento suficiente do domínio do problema a ser resolvido, os desenvolvedores não possuem o conhecimento técnico necessário ou a *user storie* é muito grande.

Caso os desenvolvedores não consigam entender as *user stories* eles devem conversar com o cliente ou usuário para esclarecer as dúvidas, não é necessário entender todos os detalhes, mas sim ter uma ideia geral sobre o assunto.

Se os desenvolvedores não possuem o conhecimento técnico necessário, uma *user storie* pode não ser estimável, pois os desenvolvedores não têm conhecimento da tecnologia envolvida, caso isso ocorra uma solução pode ser a realização de um treinamento ou pequenas experimentações para se obter o conhecimento necessário.

A *user storie* é muito grande. Por exemplo, para um site de comércio eletrônico, “Um cliente pode comprar produtos”, é uma *user storie* muito grande. Para realizar a estimativa será necessário dividir a *user storie* em outras menores, facilitando a sua compreensão e estimativa.

1.5. Pequena

O tamanho de uma *user storie* é importante por que ela não deve ser muito grande e não muito pequena que não seja possível utilizá-la no planejamento. *User stories* “grandes” são chamadas de épicos e podem ser utilizadas como lembretes de que determinado recurso necessita de maiores detalhes. Um épico deve ser dividido em *user stories* menores, baseadas nas capacidades do time do projeto e na tecnologia utilizada, permitindo assim que o cliente possa priorizar a *user storie* e o time possa estimá-la.

1.6. Testável

Uma *user storie* deve ser escrita de forma que possa ser testada. Os testes de aceitação são escritos na parte de trás dos cartões e trazem informações que são resultado das conversas entre clientes, usuários e desenvolvedores, essas notas também expressam a expectativa de funcionamento da *user stories*.

Essas notas também são utilizadas como critério para concluir se a user storie foi corretamente desenvolvida. Seguindo o exemplo da figura 2 os testes da user storie poderiam ser os mostrados na figura 3.

Testar pagamento com Visa, MasterCard e Aura.

Testar cartões vencidos.

Testar cartões com números incorretos.

Testar com valores acima do limite.

Figura 3 – Testes da *user storie* “O cliente pode efetuar o pagamento com cartão de crédito”.

2. Estimando *User Stories*

Em projetos de desenvolvimento de software dificilmente o planejamento se mantém o mesmo desde o início, funcionalidades são retiradas, acrescentadas ou quebradas em funcionalidades menores, as prioridades mudam por causa dos riscos e de novas necessidades. Devido essa grande incerteza, principalmente no início do projeto, gastar muito tempo no levantamento profundo de requisitos e investir em métricas detalhadas, pode ser um desperdício de tempo e dinheiro. Baseado em experiências próprias e de outros profissionais, Cohn (2005) propôs um gráfico onde é demonstrada a porcentagem de acerto da estimativa baseada no tempo investido para realizá-la. A figura 4 demonstra este gráfico.

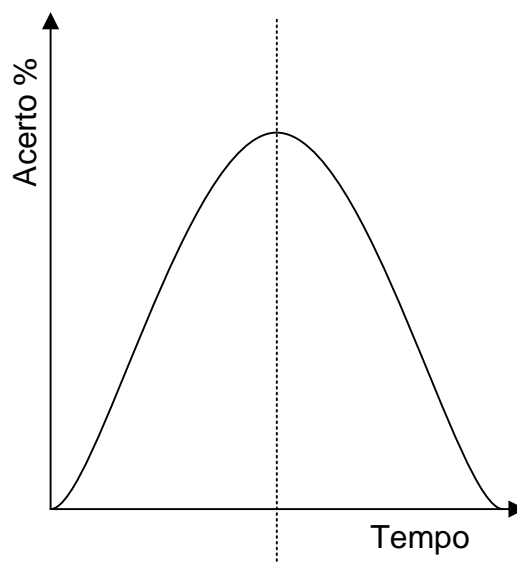


Figura 4 – Métricas: Porcentagem de Acerto X Tempo Investido

Fazendo a análise do gráfico percebe-se que se é investido pouco tempo na estimativa do projeto a margem de acerto é baixa e se muito tempo é investido também a porcentagem de acerto também tende a ser baixa. Existe um ponto ideal de investimento na estimativa, representado pela linha tracejada, onde o acerto é maximizado, é neste ponto que se baseia a estimativa de *user stories* com *story points*. A estimativa com *story points* é uma métrica diferente das técnicas tradicionais UCP (*Use Case Point*) ou APF (*Análise de Ponto por Função*), que levam em

consideração principalmente o tamanho funcional do software e em muitas vezes apenas a opinião de analistas e coordenadores, na estimativa com *story points* a opinião de todos os envolvidos no projeto é levada em consideração, essa estimativa muitas vezes se concretiza em realidade, por que não leva em conta apenas aspectos funcionais, mas a opinião de toda a equipe sobre o projeto.

2.1. Story Points

Uma *story point* é uma unidade de tempo, que deve ser definida pelo time do projeto. Essa unidade pode ser um dia de trabalho ou uma semana. O mais indicado para o controle e facilidade é que uma *story point* seja definida em um dia de trabalho. As *story points* são atribuída as *user stories* em reuniões de planejamento com a participação de todo o time, cada membro do time apresenta sua estimativa sobre quantas *story points* cada *user storie* irá consumir, esse valor atribuído deve ser escrito no cartão da *user storie*. Com essa atribuição podemos definir quantas *story points* o projeto terá e por conseqüência uma estimativa de quanto tempo levará para ser concluído.

As *story points* também são utilizadas no projeto para determinar a velocidade do time, por exemplo: em uma interação de quatro semanas o time conseguiu concluir trinta *story points*, no melhor dos palpites podemos supor que para a próxima interação o time conseguira concluir as mesmas trinta *story points*.

Para se utilizar as *story points* também é necessário que uma seqüência nos pontos seja definida, a seqüência mais utilizada são os números da seqüência de Fibonacci, do número um ao treze, esse seqüência é largamente utilizada devido aos intervalos entre os números terem intervalos entre eles apropriados conforme o número da sequêcia aumenta.

3. Técnicas de estimativa

As três técnicas mais comuns para estimativas são: a opinião de um especialista, analogia e desagregação.

Na estimativa baseada na opinião de um especialista, um especialista é perguntado sobre quanto tempo alguma coisa leva para ficar pronta. O benefício desta técnica é que a estimativa leva pouco tempo para ser concluída, já que o especialista lendo a *user storie* fará algumas poucas perguntas e dará a estimativa. Em projetos tradicionais isto não é um problema já que cada pessoa esta atrelada a uma tarefa, já em projetos ágeis onde o conhecimento deve ser compartilhado entre todo o time está técnica pode não funcionar muito bem.

Na estimativa baseada em analogia, uma *user storie* é comparada a outras *user stories* pelo estimador, nesta técnica não existe uma *user storie* base ou uma referência universal, ao invés disto cada nova *user storie* será comparada ao conjunto de *user stories* já estimadas.

A estimativa baseada em desagregação consiste em quebrar as *user stories* em *user stories* menores que sejam mais fáceis de serem estimadas.

O *planning poker* combina cada uma destas técnicas para realizar a estimativa do projeto.

4. *Planning Poker*

O *planning poker* é uma técnica que resulta em estimativas rápidas e confiáveis para o projeto, pois utiliza cada uma das técnicas de estimativa e também conta com a participação de todo time do projeto. O recomendável para se realizar a estimativa com *planning poker* é que não se ultrapasse dez pessoas na reunião, caso contrário a equipe deve ser separada em duas e as estimativas podem ser feitas independentemente.

No início do *planning poker* cada estimador recebe um conjunto de cartas com os números 1, 2, 3, 5, 8 e 13. Os cartões devem ser preparados antes da reunião e os números devem ser grandes o suficiente para que possam ser visto por todos os participantes, esses cartões também podem ser reutilizados para as próximas reuniões.

Para cada *user storie* a ser estimada, o moderador lê a descrição para que o time possa realizar a estimativa. O moderador é usualmente o *product owner* ou um analista, mas também pode ser qualquer outra pessoa não sendo necessário nenhum requisito especial para exercer este papel.

O *product owner* responde a qualquer questão que os estimadores tiverem. Depois de respondidas todas as questões, cada estimador seleciona privadamente um cartão que represente sua estimativa para a *user storie*. Os cartões não são exibidos até que cada participante tenha feito sua escolha. Após todos terem escolhido um cartão de estimativa, todos simultaneamente exibem seus cartões para que todos possam ver a estimativa. Neste ponto a estimativa de cada participante pode diferenciar de forma significativa, se isto ocorrer os estimadores que deram os pontos mais altos e mais baixos devem explicar o motivo da estimativa, isto não visa atacar os estimadores, ao contrário visa aprender o que eles estão pensando sobre a *user storie*. O grupo então pode discutir por alguns minutos a *user storie*, o moderador pode tomar notas que possam ser úteis para quando a *user storie* for programada e testada. Após a discussão cada estimador re-estima a *user storie* selecionando um cartão de forma privada e novamente são apresentados. Estas ações devem se repetir até que o time chegue a uma estimativa que esteja de acordo com a opinião do time como um todo.

As reuniões devem ser curtas de até uma hora, caso não seja possível concluir a estimativa, novas reuniões devem ser agendadas.

O *planning poker* deve ser utilizado em dois diferentes períodos do projeto, primeiro antes do projeto iniciar ou nas primeiras iterações, onde o conjunto inicial de *user stories* será definido e onde se levará mais tempo para estimar e segundo durante as iterações quando novas *user stories* forem surgindo e tiverem a necessidade de serem incluídas no projeto.

A tabela 1 demonstra um artefato da metodologia ágil SCRUM chamado *Product Backlog*, com as *user stories* estimadas após a finalização do *planning poker*.

ID	User Stories	Prioridade	Estimativa (Story Point)
1	Carregar Informações dos clientes do ERP	1	5
2	Manter Divisão	1	1
3	Manter Conta	1	2
4	Manter Saldo	1	3
5	Manter Período	1	3
6	Manter Percentuais	1	3
7	Manter Composição	1	8
8	Consultar Históricos Processados	1	2
9	Realizar Emissão de Nota Fiscal	1	3
10	Manter Demonstrativo e Item	2	5
11	Consultar Composição Apurada	2	2
12	Consultar Demonstrativo Apurado	2	2
13	Abrir e Fechar Período	2	2
14	Apurar Valores - Parte I	2	5
15	Apurar Valores - Parte II	2	8
16	Abrir pedido	2	3
17	Relatórios Gerenciais	3	13
18	Migração	3	8

Tabela 1 - Exemplo de *product backlog* estimado

Verificando a tabela acima podemos chegar a uma estimativa para o projeto. Supondo que o time de desenvolvimento tenha definido que uma story point vale 8 horas de trabalho, realizando o somatório da coluna estimativa teremos:

Tempo total=Total de *story point* x Valor de uma story point

$$624 = 78 \times 8$$

Pela tabela exemplo o projeto terá o total de 624 horas.

Metodologia

Para realização deste estudo foram utilizadas pesquisas bibliográficas em livros e internet. As pesquisas mostraram a falta de literatura em língua portuguesa sobre o assunto, fato que motivou o uso no estudo de livros e sites em língua inglesa.

Resultados e discussões

Neste estudo foi verificado que a utilização de user stories e planning poker pode ser uma alternativa interessante as metodologias de estimativa de projetos de software tradicionais, o fato destas técnicas serem simples e de fácil aprendizagem tornam sua adoção simples e de obtenção resultados positivos em pouco tempo. A estimativa de projetos de forma colaborativa e com a participação de todo o time parece ser o ponto forte para atingir o tempo ideal gasto na estimativa do projeto.

Conclusão

Neste estudo foram apresentadas as técnicas para realização da estimativa de projetos de software com a utilização de user stories e planning poker. Esta duas técnicas demonstram que a obtenção da estimativa do projeto passa por todos os envolvidos no projeto e que se as necessidades não forem claras o suficiente a estimativa mesmo utilizando este método pode falhar. Diferente das técnicas de UCP e APF, a estimativa por métodos ágeis não foca apenas no tamanho funcional do projeto e não é necessário o aprofundamento muito grande nos requisitos, o método foca também a opinião de todos os envolvidos, isso faz com que a estimativa utilizando estas técnicas tenha um diferencial dentre as outras técnicas, em termos de precisão e velocidade.

As técnicas apresentadas são uma boa alternativa aos métodos tradicionais, com a evolução na adoção das metodologias ágeis o seu uso e estudo serão ainda maiores.

Recomendações

Este estudo sobre estimativa de software pode ser complementado através de outras pesquisas com foco em:

- Comparação entre metodologias ágeis e metodologias tradicionais;
- Realização de estudos de caso para avaliação da eficácia do método proposto.

Estimativa ágil de projetos de software

Abstract

The time estimation of software projects have been a challenge to software engineer and project management, for the simple fact that estimation have a great chance of not reach with precision all time that will be necessary in the project, this happening because some aspects as functional size, complexity, implementation language, architecture, relationship with users and management model. The agile software methodologies offers techniques to maximize the results and reduce the time expended in software estimative, aiming a more realistic deadline to the project delivery. This work had as the main objectives researches and understands these techniques and fills a blank of the literature about this subject in Portuguese.

Key words: *Estimative; Agile Methodology; Planning Poker; Scrum; User Stories; XP (Extreme Programming);*

Referências Bibliográficas

COHN, Mike. *User Stories Applied*. [S.L.]: Addison-Wesley, 2004.

COHN, Mike. *Agile Estimating and Planning*. [S.L.]: Addison-Wesley, 2005.

KNIBERG, Henrik. *Scrum e XP Direto das Trincheiras*. [S.L.]: C4Media Inc, 2007.

SCHWABER, Ken. *Agile Project Management with Scrum*. [S.L.]:Microsoft Press, 2004.

Waters, Kelly. User Stories Should Be *Valuable*. Disponível em: <http://www.agile-software-development.com/2008/03/user-stories-should-be-valuable.html>. Acessado em 26 de setembro de 2009.

Wells, Don. User Stories. Disponível em: <http://www.extremeprogramming.org/rules/userstories.html>. Acessado em 26 de agosto de 2009.

Ambler, Scott W.. User Stories. Disponível em: <http://www.agilemodeling.com/artifacts/userStory.htm>. Acessado em 28 de agosto de 2009.