

Aplicação de Métricas de Software com Scrum

Mateus Luiz Gamba. Ana Cláudia Garcia Barbosa

Curso de Ciência da Computação – Unidade Acadêmica de Ciências,
Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) –
Criciúma, SC - Brasil

mateusgamba@gmail.com, agb@unesc.net

Resumo. *Este trabalho estudou a aplicação de métricas de software para obter estimativas de prazo de tempo de desenvolvimento de uma iteração do método Scrum. Além da apresentação da fundamentação teórica sobre o método Scrum e métricas, foi realizado um estudo de caso para validar as métricas definidas e posteriormente uma análise. Contribuindo para aumentar a precisão do tempo de desenvolvimento no Scrum, onde foram utilizadas as métricas Ideal Day, Planning Poker e Pontos de Função.*

Palavras-chave: *Engenharia de Software; Metodologia Ágil; Métricas de Software;*

Abstract. *This study is on the application of software metrics to obtain estimates of stated period of time in the development of an iteration of the Scrum method. Besides the presentation of theoretical background on Scrum method and metrics, it was carried out a case study to validate the defined metrics and after a review. Contributing to increase the time precision of the development in the Scrum, which were used the metrics Ideal Day, Planning Poker and Function Points.*

Keywords: *Software Engineering; Agile Methodology; Software Metrics;*

1. Introdução

Em projetos em que há muitas mudanças, refazer partes do código é uma atividade inevitável, geralmente com equipes pequenas e prazos de entrega curtos, o desenvolvimento de um planejamento para suprir essas necessidades se torna algo difícil.

Devido esses motivos e entre outros se originou a metodologia ágil com a intenção de ser adaptável a processos de desenvolvimento e mudanças de software de uma maneira flexível e rápida, alterando em partes o paradigma da engenharia de software tradicional.

A metodologia ágil enfatiza o desenvolvimento iterativo, incremental e o compartilhamento de idéias para o desenvolvimento de software, onde os projetos sofrem constantes alterações, agindo de forma a adequar-se as mudanças (ISOTTON NETO, 2008).

Entre os diversos métodos existentes, o Scrum se destaca por ser utilizado em trabalhos complexos, onde é quase impossível prever o que ira ocorrer durante o desenvolvimento do projeto. Assim, Scrum oferece um conjunto de práticas que mantém tudo visível. Isso permite que os membros do Scrum saibam exatamente o que

está acontecendo e quais são os ajustes necessários para manter o projeto em direção as metas desejadas (SCHWABER, 2004).

Segundo Pressman (2006) para um projeto ter maior probabilidade de ser bem sucedido é necessário aplicar métricas para controlar o desenvolvimento, ter estatísticas e comparações, pois não há como ter uma precisão ou controle naquilo que não se pode mensurar.

Devido o método Scrum ser atual, a pouca documentação a respeito de medições e métricas, e avaliações sobre sua utilização. Com o intuito de aprimorar o Scrum, este trabalho envolve um estudo de caso e sua respectiva análise onde descreve a aplicação de métricas de estimativas no Scrum.

Para um melhor entendimento este trabalho está organizado em três capítulos. Tendo continuidade no Capítulo dois, que explica o método proposto Scrum. No Capítulo três são apresentadas as métricas de software aplicadas, onde são descritas resumidamente e por fim é abordada no Capítulo quatro a aplicação das métricas de software no método Scrum, no qual foi aplicada em um estudo de caso e realizada uma análise para verificar qual se obteve maior compatibilidade.

2. Metodo Scrum

O método Scrum não define ferramentas e nem técnicas de desenvolvimento, mas sim, como as equipes devem trabalhar em ambientes com constantes alterações e com surgimentos de novos requisitos (ZANATTA, 2004).

O Scrum é composto papéis e responsáveis, conhecidos como *Product Owner*, *Scrum Master* e *Scrum Team*. Suas práticas são realizadas pelas etapas conhecidas como: *Sprint Planning Meeting*, *Daily Scrum*, *Sprint Review Meeting* e *Sprint Retrospective* e também composto pelos artefatos *Product Backlog* e *Sprint Backlog* que está descrito resumidamente no item a seguir.

2.1 Desenvolvimento com Scrum

O gerenciamento do projeto com Scrum, se início definindo a lista dos requisitos necessários para o desenvolvimento, essa lista é conhecida como *Product Backlog*.

O desenvolvimento ocorre num período de no máximo trinta dias, conhecido como *Sprint*, onde é realizado o *Sprint Planning Meeting* para analisar quais requisitos do *Product Backlog* tem maior prioridade, assim criando uma nova lista conhecida como *Sprint Backlog*. Durante o *Sprint* não ocorrem mudanças nos requisitos selecionados. Ao decorrer do *Sprint* a equipe desenvolve os requisitos selecionados do *Product Backlog*.

A cada dia do *Sprint* é realizada uma reunião onde cada membro relata como está o desenvolvimento de sua tarefa e se necessita de ajuda ou recursos para continuar.

Sempre no último dia de cada *Sprint* é realizada o *Sprint Review Meeting* que é uma revisão onde a equipe mostra o trabalho desenvolvido para o *Product Owner* (pessoa responsável pelo projeto) e para o *stakeholders* (pessoas interessadas pelo projeto). Também é realizado o *Sprint Retrospective* onde é feita a retrospectiva do *Sprint*, que nada mais é que uma revisão do trabalho feito.

Para que esses passos sejam seguidos de forma correta, há uma pessoa, chamada de *Scrum Master*, responsável para gerenciar e liderar a equipe. Tendo também o dever de fazer o levantamento dos requisitos até a entrega do projeto final.

3. Métricas de Software

Segundo Pressman (2006) a mensuração é aplicada no processo de desenvolvimento de software ou atributos de um produto com o objetivo de melhorá-lo de forma contínua que utilizando ao longo do projeto auxilia na estimativa, no controle de qualidade, na avaliação da produtividade e no controle do projeto.

Com as métricas os gerentes conseguem informações quantitativas que auxiliam na tomada de decisões e para obter uma visão melhor do trabalho que está sendo realizado, podendo assim desenvolver um software mais confiável (INTHURN, 2001).

Basicamente a medição abrange a coleta dos dados para a computação das métricas e por fim, a avaliação dos resultados obtidos.

O item a seguir descreve a importância da aplicação de estimativas para aplicação de métricas.

3.1 Estimativas

Com a aplicação de estimativas é possível coletar métricas que permitam prever a quantidade de pessoas necessárias, o tempo necessário e os custos para o desenvolvimento do projeto. "Assim, torna-se importante o investimento na implantação de um processo de estimativas" (HAZAN, 2008, p. 25).

A estimativa geralmente é realizada por um engenheiro de software ou especialista que analisa os requisitos garantindo a qualidade, e então estima o tamanho do projeto de software. Após a definir o tamanho, é realizada a derivação para determinar o esforço, prazo (cronograma) e custo (orçamento). Caso ocorram mudanças nos requisitos é necessário re-estimar (HAZAN, 2008).

Para estimar existem várias técnicas e métodos, como: *Ideal Day*, *Planning Poker* e Ponto de função que são descritas nos itens a seguir.

3.2 Ideal Day

É utilizado para realizar estimativas de forma ágil, sendo aplicada para planejar o projeto e iterações.

Segundo F. Alves, M. Alves e Fonseca (2008) *Ideal Day* corresponde à quantidade de trabalho que um profissional da área consegue concluir em um dia de trabalho.

De acordo com Martins (2007) velocidade é calculada a partir do número de horas que a equipe gasta para implementar um trabalho equivalente a um *Ideal Day*. Caso o item passe um dia de trabalho, é sugerido decompor esse item em itens menores que se consiga implementar em apenas um dia.

Segundo Martins (2007) para efetuar o cálculo dos dias estimados utiliza a seguinte fórmula:

$$DE = \frac{IED}{1 - IED_REAL\%}$$

Onde:

DE: quantidade de dias estimado para concluir a tarefa;

IED: prazo necessário para implementar o item, esse prazo é definido pela equipe;

IED_REAL%: percentual que indica a estimativa de quanto tempo do dia o desenvolvedor ficara dedicado a implantação do item.

3.3 Planning Poker

No *Planning Poker* para estimar é necessário do *Product Backlog* e de um baralho de cartas, onde as cartas devem ter os seguintes números 0, 1, 2, 3, 5, 8, 13, 20, 40 e 100, ou seja, semelhante à seqüência de *Fibonacci* (PlanningPoker, 2009).

Cada membro da equipe deve possuir esse baralho de cartas, pois cada item do *Product Backlog* corresponde a um valor definido através de rodadas entre os membros.

A cada funcionalidade do *Product Backlog*, os membros relacionam uma carta com o valor que acham ser o ideal. Após todos os membros jogarem, é discutido qual o valor ideal para aquele item, caso a equipe não chegue a um consenso, e feito mais rodadas até a equipe chegar a um consenso, e assim, continua com os outros itens do *Product Backlog*.

3.4 Pontos de Função

Conforme Dekkers (1998) Pontos de Função (PF) medem o tamanho funcional do software, onde o tamanho funcional pode ser definido como uma medida de tamanho de software, essa medida é realizada a partir da funcionalidade implementada em um sistema sob o ponto de vista do usuário.

Para a contagem do ponto de função são realizadas sete etapas, como é representada na Figura 1:

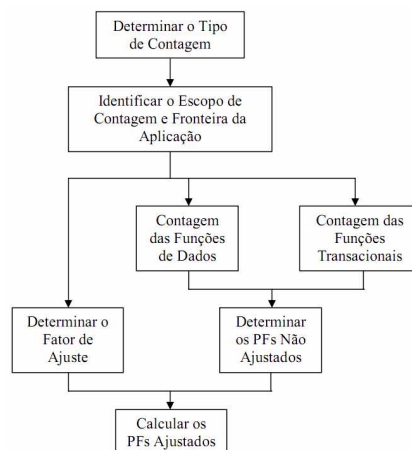


Figura 1 - Procedimento de Contagem de PF

Fonte: HAZAN, 2001.

PF além de medir o tamanho funcional do software, oferecem os seguintes benefícios:

- a) melhor acompanhamento na produtividade da equipe;
- b) melhoria no controle da qualidade e os custos de desenvolvimento;
- c) melhor previsão do projeto por possibilitar realizações de estimativas nas fases iniciais do desenvolvimento de software.

Com a descrição dessas métricas pode-se obter um resultado mais significativo em projetos, que a seguir é realizado um estudo de caso e uma análise para verificar qual se adéqua melhor ao projeto definido.

4. Métricas de Software aplicadas no Scrum

Este trabalho tem como objetivo descrever uma análise de estimativas de prazo no método Scrum sobre o estudo das métricas de software.

Com o estudo das métricas, foi realizado um estudo de caso com a finalidade de analisar qual métrica se adaptava melhor ao método Scrum.

Foram realizadas análises em uma empresa que estava no início da implantação do método Scrum e com a finalidade de adotar uma métrica para estimar o prazo. Os detalhes estão na seção seguinte.

4.1 Estudo de Caso

A empresa estudada é desenvolvedora de sistemas integrados de Gestão Empresarial, tendo aplicações voltadas à gestão empresarial.

Os sistemas dessa empresa é em plataforma web, sendo desenvolvido sobre o banco de dados pós-relacional Caché, com sua linguagem nativa, também envolvendo as linguagens *HyperText Markup Language* (Html), *Cascading Style Sheets* (Css) e Javascript.

Ao adotar o Scrum, a empresa definiu os seguintes requisitos de forma que agilizasse os seus processos:

- a) equipes de quatro integrantes, sendo geralmente composto por: dois desenvolvedores, um analista de teste e um engenheiro de software;
- b) *Sprint* de dez dias úteis;

Na aplicação das métricas para estimativa de prazo de desenvolvimento, a empresa optou estudar e testar às métricas: *Ideal Day*, *Planning Poker* por serem consideradas ágeis e adaptáveis ao Scrum, e Pontos de Função por se uma métrica conhecida por diversas empresas para definir um preço fixo por pontos baseada na norma da IFPUG.

Para uma primeira análise, essas métricas foram aplicadas em um *Sprint* para verificar qual se adapta melhor nos processos de desenvolvimento da empresa.

O projeto em desenvolvimento foi sob demanda para um de seus clientes, que consistia em três novos módulos que seriam desenvolvidos tendo base o produto já existente.

A empresa teve a colaboração de um especialista da empresa do cliente solicitante, o *Product Owner*, para descrever e priorizar os requisitos a serem desenvolvidos.

O projeto teve início com a elaboração do *Product Backlog* de 34 requisitos, definidos pelo *Scrum Master*, *Product Owner* e os *stakeholder*.

Após a elaboração do *Product Backlog* tem-se o início do *Sprint*, começando pela primeira etapa, o *Sprint Planning Meeting*, que foram definidos os itens do *Product Backlog* de maior prioridade e as estimativas. Nessa etapa o *Product Owner* selecionava os itens do *Product Backlog* detalhando-o a equipe até ela obter informações suficientes para desenvolver e para definir até qual item conseguiria para completar no *Sprint*.

A cada item do *Product Backlog* a equipe aplicou a métrica *Ideal Day* e *Planning Poker*.

A previsão da métrica *Ideal Day* foi realizada de acordo com a seguinte escala de porcentagem do dia: 0,25, 0,5, 0,75, 1, 1,25, 1,5 e assim por diante. O resultado apresentou um total de 13,5 dias estimado pela equipe.

Para encontrar o *Ideal Day* a equipe discutiu cada item até chegar a um consenso, onde foi considerado que a equipe trabalharia 90% do seu dia que é representado por oito horas de trabalho diário. Com os valores definidos, foi aplicada a fórmula, onde se obteve 14,64 dias.

No processo de estimativas de *Ideal Day* foram definidos também um período em horas para cada escala, resultando em um total de 117,3 horas.

Com a métrica *Planning Poker*, cada membro da equipe possuía um jogo de cartas com a seqüência de *Fibonacci*. A equipe iniciou o jogo escolhendo o requisito mais fácil a ser desenvolvido.

Após varias rodadas e discussões, resultou em um total de 117 pontos.

No *Planning Poker* foi utilizada a mesma lógica do *Ideal Day* para medir a estimativa de tempo em horas. Para cada ponto foi estimado um valor de hora, assim resultando um total de 117,5 horas.

Na segunda etapa do *Sprint Planning Meeting* se planejou como seria realizado o *Sprint* a partir do *Product Backlog*. Nesta etapa a equipe criou o *Sprint Backlog* com os itens do *Product Backlog* descritos de forma mais resumida e técnica, onde alguns itens foram divididos em partes menores, por uma ordem que se adaptasse de forma mais ágil ao desenvolvimento, resultando em um total de 23 requisitos.

Com a definição do *Sprint Backlog* se iniciou o desenvolvimento, onde também foi realizada a análise das informações discutidas na reunião para iniciar o processo de contagem dos PF¹.

Com as informações coletadas durante o *Sprint Planning Meeting* e o *Sprint Backlog* foi realizada a contagem de PF.

Para a contagem foi aplicado os seguintes processos:

- a) tipo de contagem;
- b) escopo de contagem e fronteira da aplicação;
- c) contagem das funções de dados;
- d) contagem das funções transacionais;
- e) determinação dos PFs não ajustados.

Por serem três novos módulos foi classificado como projeto de desenvolvimento.

A estimativa realizada foi separada por tipo de dados (arquivos lógicos internos) e por tipo de transação (entradas externas, saídas externas e consulta externas), pois para cada tipo poderia ser representado um prazo diferente.

O fator de ajuste e o cálculo de PFs ajustados não foram aplicados devido os módulos serem desenvolvidos sobre a plataforma do software já existente, por ser orientado objeto, plataforma Web e pelo fato do Scrum ser uma metodologia ágil e PF não ser considerada uma métrica de estimativa de prazo ágil.

Para cada ponto foi representado um prazo de horas, semelhante às métricas descritas acima, assim, obtendo um total de 211 pontos que resultando em 40,30 horas.

Apesar de o resultado ter divergência consideravelmente grande das outras métricas, a equipe preferiu não recalcular para verificar se algum dos valores resultava a um valor semelhante ao do projeto concluído.

Durante os períodos da manhã dos dias do *Sprint*, a equipe se reunia de pé em círculo para realizar o *Daily Scrum*, e sempre após encerrar o *Daily Scrum* cada membro preenchia a planilha *Sprint Backlog* para ter um acompanhamento para análise das métricas.

Após a finalização do desenvolvimento do *Sprint* a equipe se reunia para o *Sprint Review Meeting* para discutir o *Sprint*.

¹ A aplicação de pontos de função foi baseada principalmente na publicação dos autores Vazquez, Simões e Albert (2007), que consta nas referências.

A Figura 2 representa o funcionamento de um *Sprint* da aplicação do Scrum no projeto.

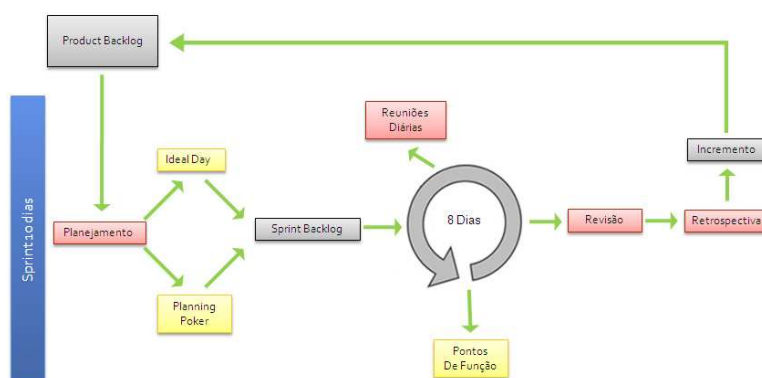


Figura 2. Demonstrativo de Processo Aplicado

Com o *Sprint* realizado foram analisadas às métricas, onde se encontram melhor descrita no item seguinte.

5.2 Análise das Métricas Aplicadas

O objetivo deste estudo foi verificar a viabilidade de aplicação das métricas *Ideal Day*, *Planning Poker* e PF para uma primeira estimativa de prazo necessário para o desenvolvimento do primeiro *Sprint*.

O início da coleta destas estimativas se fez necessário para que a empresa começasse a formar um histórico de dados através da comparação das horas estimadas com o total de horas no final do *Sprint*.

A Tabela 1 apresenta o total das horas estimadas das métricas *Planning Poker*, *Ideal Day* e PF em comparação com o total de horas do projeto.

Tabela 1. Comparação das estimativas

Tempo Total do Projeto	<i>Planning Poker</i>	<i>Ideal Day</i>	Pontos de Função
107,2	117,5	117,2	40,27

As métricas *Ideal Day* e *Planning Poker* analisadas pela equipe deram um resultado bem similar, pois as informações foram sugeridas com valores de pouca diferença.

A quantidade de horas calculadas pela equipe durante o *Sprint* ficou na média de 8,66% abaixo do tempo estimado das métricas *Ideal Day* e *Planning Poker*.

Para o próximo *Sprint* a equipe avaliou e sugeriu novos valores das cartas do *Planning Poker* que resultaria em um valor de 113 horas se fosse aplicado nesse primeiro *Sprint*. Já para o *Ideal Day* foi sugerido para os próximos *Sprint* um percentual menor de tempo do dia que o desenvolvedor fica dedicado a implantação do item, nesse caso de 90% definido no primeiro *Sprint* para 75% nos próximos *Sprints*.

A coleta da estimativa de PF não foi possível definir a complexidade do projeto e sim, o tamanho. Para as próximas estimas realizadas se sugeriu novos valores,

resultando em um total de 211 pontos e 106 horas/PF caso fosse aplicado nesse primeiro *Sprint*.

Pela primeira análise de PF a empresa optou por analisar mais algumas vezes, para posteriormente fazer outro estudo para analisar custos, onde PF tem a vantagem de se basear na norma da IFPUG, assim tendo um preço fixo por pontos.

As estimativas *Ideal Day* e *Planning Poker* ficaram bem próximas da realidade atual da equipe de desenvolvimento.

Nessa primeira aplicação do *Planning Poker* a equipe encontrou dificuldades em estimar o prazo de tempo, devido não conseguir encontrar uma estimativa de prazo ideal para a seqüência de *Fibonacci*.

Para aplicação do PF houve dificuldade na elaboração e coleta de dados, pois o pelo Scrum ser uma metodologia ágil e PF não ser considerada uma métrica de estimativa de prazo ágil.

Para obter estimativas mais precisa é necessário que todas as iterações da fase de elaboração estejam completas, assim servindo como um histórico para outros projetos.

Com a adoção do Scrum a questionamentos sobre como comparar horas estimadas com total de horas realizadas. A expectativa a está comparação é para melhorar cada vez mais as próximas estimativas, permitindo melhorar a previsibilidade do progresso do projeto.

6. Considerações Finais

No método Scrum como em qualquer outro método de gerenciamento, torna-se importante a aplicação de métricas para medir a produtividade, qualidade, prazo e custo. Assim a implantação de uma dessas métricas aumenta essas estimativas qualificando o projeto.

Por meio de um estudo de caso, este trabalho analisou o papel das métricas para estimar o prazo de desenvolvimento de um projeto. Nesse projeto foi aplicado o método Scrum com as métricas *Ideal Day*, *Planning Poker* e Pontos de Função.

No projeto avaliado, apesar das estimativas do *Ideal Day* e do *Planning Poker* terem resultados semelhantes, por serem empíricas, o *Ideal Day* se apresentou de melhor forma a equipe nesse início de projeto, devido a sua facilidade em estimar o prazo em dias que se pretendia terminar o requisito.

A métrica *Planning Poker* pela análise se torna interessante para medir o tamanho dos requisitos futuros, pois ao decorrer do projeto o prazo de dias do *Ideal Day* pode alterar devido à troca de membros da equipe ou outros problemas não previstos, nesse caso, com o resultado do *Planning Poker* apenas seriam alterados os valores das horas dos pontos, assim não tendo a necessidade de medir novamente.

A métrica Pontos de Função obteve o resultado divergente das outras métricas, devido não analisar a complexidade da iteração do usuário com o sistema em desenvolvimento, um dos propósitos podendo ser pelo fato do sistema ser web. Para aplicação do Pontos de Função definiu se que seria necessário mais estudos.

Contudo o estudo foi satisfatório apesar de ser estimado somente um *Sprint*, com a aplicação de duas métricas consideradas ágil e uma considerada tradicional, onde houve um entendimento mais específico da aplicação no projeto.

Referências

- Agile Manifesto. Manifesto for Agile Software Development. Disponível em: <<http://www.agilemanifesto.org>>. Acesso em: out. 2009.
- Alves, Fernanda; Alves, Marcia; Fonseca, Isabella. Ideal Day e Priorização: Métodos Ágeis no Planejamento. Engenharia de Software, Rio de Janeiro, n., p.8-13, 01 nov. 2008.
- Dekkers, Carol A.. Pontos de Função e Medidas O que é um Ponto de Função? Disponível em: <<http://www.bfpug.com.br/Artigos/Dekkers-pontosDeFuncaoEMedidas.htm>>. Acesso em: 30 jun. 2009
- Hazan, Cláudia. Análise de Pontos de Função: Uma aplicação nas estimativas de tamanho de Projetos de Software. Engenharia de Software, Rio de Janeiro, n. , p.25-30, 01 jun. 2008.
- Hazan, Cláudia. Medição da Qualidade e Produtividade em Software. In: WEBER, Kival Chaver; ROCHA, Ana Regina Cavalcanti; NASCIMENTO, Célia Joseli. Qualidade e Produtividade em Software. 4. ed. São Paulo: Makron Books, 2001. p. 25-41.
- Inthurn, Cândida. Qualidade & Teste de Software. 2. ed. Florianópolis: Visual Books Editora, 2001. 108 p.
- Isotton Neto, Erasmo. Scrumming: Ferramenta Educacional para Ensino de Práticas do SCRUM. 2004. 80 f. - Pontifícia Universidade Católica Do Rio Grande Do Sul, Porto Alegre, 2004.
- Planning Poker. Play. Estimate. Plan. Disponível em: <<http://www.planningpoker.com/>>. Acesso em: 01 nov. 2008.
- Pressman, Roger S. Engenharia de Software. 6. ed São Paulo: McGraw-Hill, 2006. 720 p.
- Martins, José Carlos Cordeiro. Técnicas para Gerenciamento de Projetos de Software. 1. ed. Florianópolis: Brasport Editora, 2001. 465 p.
- Schwaber, Ken. Agile Project Management with Scrum. Microsoft Press, 2004.
- Zanatta, Alexandre Lazaretti. XScrum: uma proposta de extensão de um Método Ágil para Gerência e Desenvolvimento de Requisitos visando adequação ao CMMI. 2004. 180 f. - Universidade Federal De Santa Catarina, Florianópolis, 2004.