Técnicas de Abordagem de Estimativa Ágil: Planning Poker e Ideal Day

RogerRitter.com.br

rogersmartlife@gmail.com

Resumo. *Através da estimativa de software é possível garantir uma gerência de projetos mais assertiva quanto aos prazos e custos. Sua importância já não passa despercebida, pois o mercado cada vez mais deseja saber quando ficará pronto e quanto irá custar. Este artigo apresenta duas técnicas de estimativa, ambas utilizadas para estimar software em ambiente ágil de desenvolvimento.*

1. Introdução

Um projeto de software consiste de atividades realizadas por uma equipe de trabalho que atua em busca de um objetivo comum. Dentro desse processo, pode-se considerar que a atividade de estimar o esforço necessário para o desenvolvimento de um projeto é um elemento fundamental para o sucesso.

Dados mensuráveis relevantes desses fatores são oriundos da utilização de técnicas de estimativas sobre a métrica do tamanho de um projeto, e que podem ser as mais diversas variando desde analogias entre projetos e baseadas na experiência do desenvolvedor, até abordagens mais metódicas utilizando modelos matemáticos.

Este artigo apresenta, entre outros conceitos, duas técnicas que podem ser utilizadas para a abordagem de uma estimativa no desenvolvimento de software ágil. A técnica Planning Poker e a técnica Ideal Days.

2. Estimativa de Software

Estimativas em métodos ágeis, segundo BASSI (2012) são compostas por uma dupla de valores <*v*,*p*>, onde *v* é um palpite sobre determinado evento ou atividade e *p* é a probabilidade de *v* acontecer. Equipes ágeis se baseiam-se na comunicação e na transparência. Ao invés de tratar suas estimativas com fatos, admitem que existe uma incerteza ao valor estimado e evidenciam isso para que o cliente e outros envolvidos também tomem ciência do grau de dificuldade de cada tarefa.

Nas estimativas de longo prazo as incertezas são maiores e a medida que o tempo passa e o conhecimento sobre o assunto aumenta, as estimativas podem ser refeitas considerando um nível maior de detalhamento e assim associando em probabilidades maiores de acerto.

MCCONNELL (2006) ilustra através de uma figura um processo de estimativa.

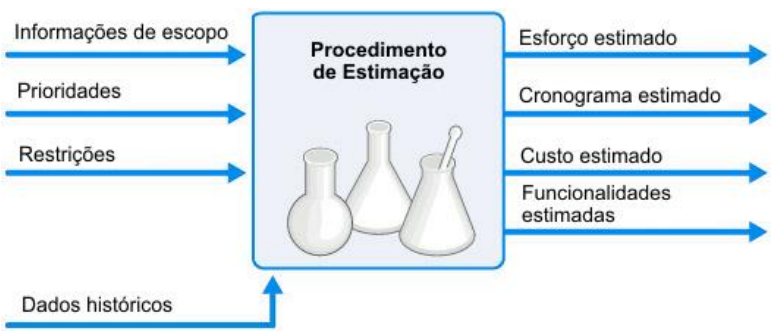

Figura 1. Procedimento de Estimativa de Software

A unidade de medida para se estimar segundo BASSI (2012) pode ser pontos, dias ideias ou horas de trabalho. Nos pontos as medidas são abstratas e definidas pela equipe a partir de uma tarefa pequena, ou cerimônia, que ela considera como sua unidade e os demais valores são atribuídos as funcionalidades de comparação por esta unidade. Um dia ideal é o quanto seria produzido em um dia de trabalho se 100% do tempo fosse dedicado a atividade, sem interrupções ou distrações. As horas de trabalho apontam quanto tempo será preciso para concluir-se-á cada tarefa. Esta última medida geralmente é utilizada para chegar ao custo do desenvolvimento.

Os pontos e os dias ideias fornecem estimativas de tamanho, que são medidas sobre o volume de trabalho. As horas de trabalho preveem a quantidade de tempo necessário para realizar tal tarefa, denominado estimativa de duração, defende BASSI (2012). Medidas de duração podem ser obtidas a partir da velocidade da equipe, que é a quantidade de pontos que ela consegue concluir a cada iteração. De qualquer forma, é importante que medidas de tamanho e duração fiquem separadas, pois a velocidade da equipe pode variar durante o projeto e isto influencia as estimativas de duração, que são as que os clientes estão mais interessados.

Baseando as estimativas no tamanho, quando a velocidade de desenvolvimento mudar, basta identificar a nova velocidade e derivar as novas estimativas de duração.

2.1 Cone da Incerteza

MCCONNELL (2006) fez a definição de um cone para ilustrar que à medida que o projeto avança, a incerteza sobre ele diminui, proporcionando assim melhores estimativas. Denominou como ‘Cone da Incerteza’ conforme mostrado na Figura 2, onde o grau de incerteza na fase inicial é de 4x e a precisão da estimativa é de 0,25x. Porém ao decorrer do projeto, essas medidas diminuem. E para que haja a diminuição da incerteza durante o avanço do projeto, é necessário monitoramento.

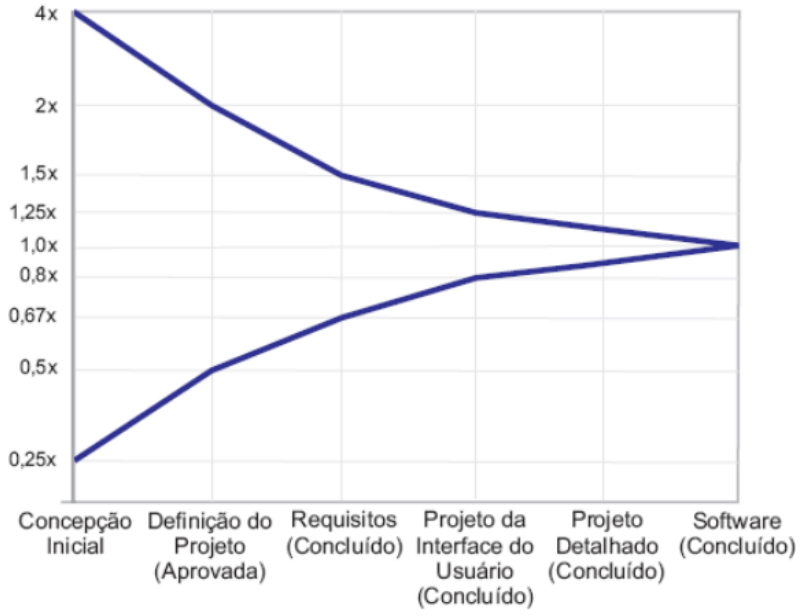

Figura 2. Cone da Incerteza, proposta por MCCONNELL (2006). Com grau de incerteza na vertical e etapas de desenvolvimento de Software na horizontal.

Caso não seja realizado o gerenciamento de cada etapa do projeto, podem produzir nuvens de incerteza, conforme ilustrado na Figura 3.

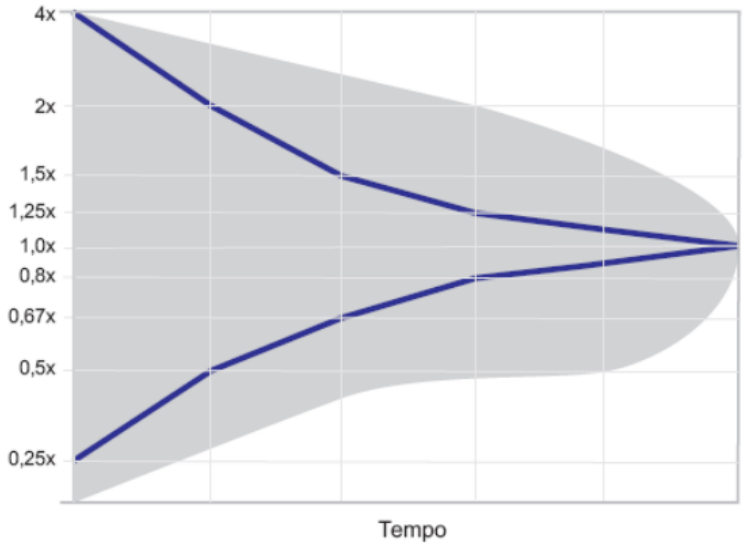

Figura 3. Nuvens de Incerteza

Na figura 3, podemos ver uma grande nuvem no gráfico, que representa a falta de gerenciamento nas diversas etapas do processo, ocasionando maior incerteza sobre o projeto. Como a incerteza do projeto ainda é elevada, reflete-se em um menor nível de precisão nas estimativas.

SIMÕES (1999) afirma que a estimativa de esforço é fundamental para o projeto de desenvolvimento de software, pois viabiliza maior certeza do escopo do projeto em sua fase inicial e assim possibilitando a entrega do produto com qualidade e prazo, garantindo uma maior competitividade.

2.2 Abordagem de Estimativas

As principais abordagens de estimativa de software trabalham com o princípio de que o tamanho de um software é uma métrica para a determinação do esforço para a sua construção. Essa, segundo HAZAN (2008), é uma variante importante para determinar a execução, o custo e o tempo de desenvolvimento. PUTMAN (1992) destaca que a estimativa de tamanho de software é o coração do processo de estimativas de um projeto de software. Com estas informações bem definidas é possível garantir um melhor gerenciamento do projeto, obtendo-se assim maior exatidão no processo.

Além disto, para estimar é preciso que o responsável, ou no caso da metodologia ágil de desenvolvimento de software, os responsáveis, sejam capazes de garantir a qualidade do produto através da análise dos requisitos. Para tanto, a escolha correta da abordagem de estimativa a aplicar no projeto é fundamental para extrair o tipo de estimativa desejado.

2.2.1 Estimativa de Tamanho

As métricas de tamanho de software, segundo PUTMAN (1992), surgiram com o objetivo de estimar o esforço (número de pessoas-hora) e o prazo associados ao desenvolvimento de sistemas.

Para saber o custo de um projeto de software precisamos saber o esforço necessário para desenvolvê-lo e para determinar o esforço precisamos saber o tamanho do projeto de software. Desta forma, determinar o tamanho de um projeto de software é uma das primeiras e principais atividades relacionadas às estimativas a serem efetuadas durante o clico de vida do projeto.

Realizar uma estimativa de tamanho de um projeto não é uma tarefa trivial, pois exige um conhecimento sobre técnicas de estimativas, base histórica e conhecimento sobre o projeto a ser estimado.

KNIBERG (2009) descreve que no Scrum as equipes devem estimar o tamanho que é igual a quantidade de trabalho de cada item aos quais o time se comprometeu. Somando o tamanho de cada item concluído ao final de cada Sprint, teremos a estimativa de velocidade.

2.2.2 Estimativa de Velocidade

KNIBERG (2009) descreve que a estimativa de velocidade é uma medida da capacidade, ou seja, a quantidade que o time consegue entregar por Sprint.

Saber a velocidade média de um time é importante pois permite fazer previsões mais realistas sobre quais itens pode-se entregar no futuro, e portanto, fazer planos de entregas cada vez mais próximos a realidade.

2.2.3 Estimativa de Esforço

PRESSMAN (1995) declara que mesmo após desenvolver uma estimativa de quantidade de trabalho a ser feito através da medição do tamanho do software, estimar o esforço necessário para o desenvolvimento do projeto não é uma tarefa simples. Um exemplo disto é que a relação de tempo e pessoas sofrem influência por uma série de fatores.

Um projeto estimado em um mês com oito pessoas e com uma disponibilidade de quatro desenvolvedores, não obrigatoriamente levará um mês para ser concluído. É preciso checar riscos, restrições tecnológicas que serão utilizadas no desenvolvimento, e ainda fatores mais subjetivos, como a experiência dos integrantes da equipe.

Existem diversas maneiras de obter a estimativa de esforço, porém nenhuma pode ser tomada como verdade absoluta, portanto, cabe ao responsável pelos cálculos uma comparação dos resultados entre mais de uma técnica. Se os resultados tiverem valores próximos, a estimativa poderá ter sucesso.

2.3 Técnicas de Estimativas

Em alguns casos as estimativas são realizadas apenas com a experiência como guia, esse método pode ser útil para projetos semelhantes entre si, uma vez que exigirão a mesma quantidade de esforço, haverá um custo parecido e levarão um período de tempo equivalente para ser concluído. Porém quando existe um projeto novo, apenas a experiência não é o suficiente, destaca VALENÇA (2007). Existe uma série de abordagem de estimativas existentes, cada qual com sua particularidade, mas todas com os seguintes atributos em comum:

- Escopo, prioridades e restrições devem ser estabelecidos antecipadamente.
- Métricas de software são utilizadas.
- Informações históricas usadas como base de estimativas.

Estes atributos têm o objetivo de determinar o tamanho do software, e a partir do tamanho obter o esforço necessário e consequentemente derivar custos, cronograma e funcionalidades do software, conforme mostra a Figura 4, ilustrada por MCCONNELL (2006).

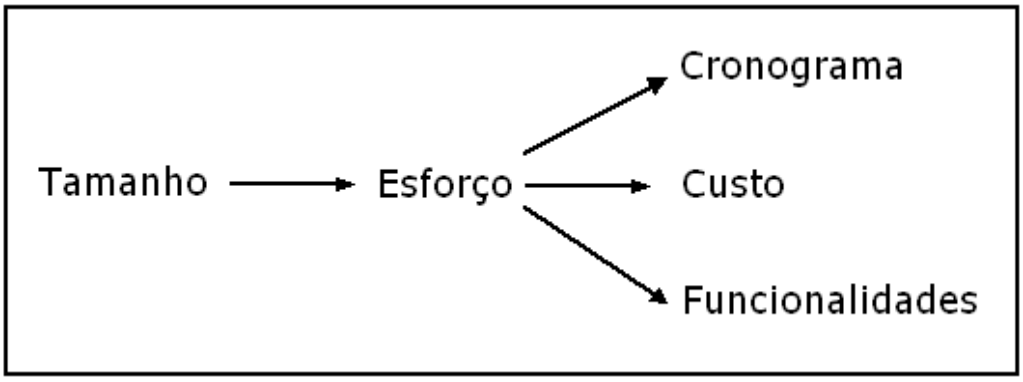

Figura 4. Derivando esforço, cronograma, custo e funcionalidades.

É importante observar que as saídas são influenciadas pela mudança dos dados de entrada e não pela mudança da técnica de estimativa, descreve VALENÇA (2007). Por isto o processo de estimativa é bastante sensível a qualidade dos dados de entrada, como mostra a Figura 5.

Figura 5. Sensibilidade do processo de estimativa quanto à qualidade dos dados.

São comuns erros no processo de estimativa, pois os requisitos do software geralmente são levantados em tempo limitado, e em momento em que não se sabe ao certo todos os requisitos do sistema. Isso traz prejuízos em ambas as partes envolvidas, atrasando a entrega, aumentando os custos e dificultando renegociações.

2.3.1 Planning Poker

O Planning Poker foi definido pela primeira vez e nomeado por James Grenning, em 2002, e mais tarde popularizada por Mike Cohn no livro Agile Estimating and Planning no qual registrou o termo.

Trata-se de uma técnica de estimativa voltada para as metodologias ágeis, inclusive o Scrum no qual Grenning GRENNING (2002) afirma que é a melhor ferramenta encontrada para se estimar tamanho de projetos em metodologias ágeis.

O Planning Poker pode ser considerado a combinação de três técnicas menos comuns de métodos de estimativas ágeis: opinião de especialista, analogia e desagregação, destaca COHN (2005), o que a torna uma prática bastante interativa e prática.

Consiste-se na obtenção de estimativa através de um jogo de cartas. A ideia principal é permitir que todos os membros da equipe de desenvolvimento (programadores, testadores, design, analistas, etc.) participem colocando a sua visão de complexidade, levando em consideração o fator tempo e esforço para pontuar a estória e após juntos chegar a um denominador comum na equipe. Nas metodologias ágeis, a equipe não deverá exceder a dez pessoas.

As estimativas de esforço e tempo são aplicadas a estórias e essa atividade é de responsabilidade da equipe técnica, frisa CTIC-UFPA (2011). Onde, segundo COHN (2005), o Product Owner participa no planejamento, mas não estima.

No Planning Poker, cada integrante tem a sua disposição um baralho de 12 cartas, numeradas numa sequência similar a encontrada nos números de Fibonacci, conforme mostra a Figura 6.

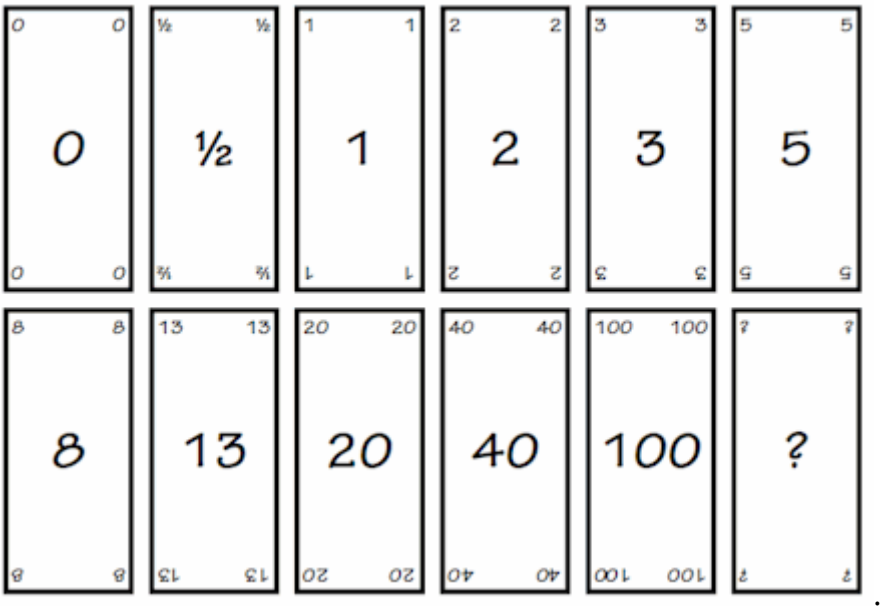

Figura 6. Cartas de Planning Poker.

Valores acima de 20 são considerados altos e, portanto trata-se de uma estória grande, que podem não ser completamente finalizadas numa única iteração. Uma alternativa é de detalhar mais a estória, quebrando-a em pedaços menores, com redução

em suas complexidades. O recomendável é que ao final do Planning Poker, se consiga fazer com que as estórias da iteração possuam algum valor do intervalo de ½ a 13. No entanto, é necessário evitar outro problema que é deixar que as estórias tornarem-se muito pequenas, pois isso provavelmente tornará a equipe vítima do micro gerenciamento, destaca KNIBERG (2009).

O baralho do Planning Poker ainda pode conter algumas cartas com propósitos especiais. A carta com valor '0' (zero) indica que uma estória está finalizada ou é tão pequena que pode ser resolvida em questão de minutos. A carta com '?' (interrogação) indica que o participante não tem o conhecimento apropriado para estimar o esforço necessário para a estória. Não é frequentemente utilizada, mas se for, a equipe precisa trabalhar a respeito da estória para tentar chegar a um melhor consenso entre os membros da equipe.

Durante o Planning Poker, devem ser realizadas rodadas para obter a estimativa da estória, com base nestes valores. As diferenças que surgirem durante estas rodadas deverão ser mediadas por um coordenador. No Scrum, este papel é de responsabilidade do Scrum Master. O Product Owner, por sua vez, será responsável por explicar as estórias, sendo importante para retirar as possíveis dúvidas que surgirem a respeito destas estórias.

Segundo CTIC-UFPA (2011) e COHN (2005), os seguintes procedimentos devem ser tomados ao se estimar uma estória, quando realizado por uma equipe Scrum:

1. Inicialmente pontua-se a estória que mais se aproxima do valor '3' (três), pois servirá como referência para as demais.
2. As estórias são apresentadas por vez. O Product Owner descreve para a equipe Scrum a estória e fornece informações a respeito do seu valor de negócio. A seguir, o coordenador vai perguntar o esforço necessário para concluir a história aos membros da equipe.
3. Cada membro da equipe deve pensar a respeito do tempo e esforço necessário para se implementar a estória lida. Os membros da equipe devem estimar considerando todas as tarefas envolvidas pelos responsáveis em desenvolver, testar, criar design, etc. Então, deve escolher uma carta no baralho correspondente ao valor desta estimativa e coloca-a virada para baixo.
4. Quando todos os membros fizerem o procedimento acima, então devem revelar as cartas escolhidas simultaneamente. Isso faz com que cada membro da equipe pense por si próprio na hora de uma decisão de estimativa.
5. Todos avaliam os resultados e verificam se houve convergência entre as cartas mostradas, ou seja, todas as estimativas possuam valores aproximados para a mesma estória.
6. Caso contrário, o Scrum Master solicita aos membros, que mostraram o menor e o maior valor estimado, que expliquem o motivo que os levaram a tal estimativa. Isto faz com que os integrantes reflitam sobre alguns pontos da estória, o que pode fazer com que mudem os valores propostos para as estimativas. Então, uma nova rodada é realizada até que as estimativas de esforço cheguem a uma convergência.
7. A estimativa final da estória será o valor que tiver maior ocorrência ou a média entre as estimativas informadas. Então começa uma nova rodada com a leitura

da próxima estória pelo Product Owner. O processo se repete até que tenha sido concluída a estimativa das estórias dos itens do Product Backlog.

Estudos de FONSECA (2008) mostram uma melhora na eficiência das estimativas com pontos por estórias quando utilizada esta técnica para atribuição dos pontos. De acordo com o autor, o maior problema com as técnicas não-estruturadas é que pessoas com personalidades mais fortes e indivíduos com, supostamente, melhor compreensão da estória, podem influenciar os demais participantes. O sujeito mais disposto a falar, assim que tiver a oportunidade, vai tomar a palavra e dar sua estimativa, deixando os outros membros estimulados a seguir aquela linha de pensamento apresentada, ou com receio de expor seu ponto e vista, se manterão caladas.

2.3.2 Ideal Days

O Ideal Day é utilizado para realizar estimativas de forma ágil, sendo aplicada para planejar o projeto e iterações. MARTINS (2007) faz ênfase a velocidade calculada a partir do número de horas que a equipe gasta para implementar um trabalho equivalente a um Ideal Day (Dia Ideal).

COHN (2005) defende que um Ideal Day corresponde à quantidade de trabalho que um profissional de nível sênior, com fluência nas tecnologias e ferramentas envolvidas consegue realizar em 08 (oito) horas de trabalho dedicadas (sem interrupções). Se tratando de uma estimativa empírica, executada por especialistas para desenvolvimento com base em exploração adaptativa. Por isto o nome de Dia Ideal.

É importante que se compreenda que o "Dia Ideal", com 08 (oito) horas de trabalho sem interrupções, de um "desenvolvedor ideal", raramente irá ocorrer na prática, e, portanto deve ser utilizada unicamente como “moeda” estável para quantificação de tamanho de referência e balizador ideal de produtividade.

COHN (2005) ainda afirma que o item seja equivalente a um dia de trabalho, caso não seja possível, é sugerido decompor esse item em itens menores para 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 estimados 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.

Para se manter uma unidade, afirma FONSECA (2008), não é interessante expor se um integrante executa suas atividades em mais ou menos tempo. Trata-se de uma

dinâmica do grupo. Ele deve aprender como interagir melhor para a busca de entrega de maior retorno de valor para o cliente. Se for necessário utilizar de técnicas como *pair-programming* para agilizar o desenvolvimento e validação de um requisito, o time como um todo deve escolher o melhor caminho.

2.3.2.1 Caso Prático

FONSECA (2008) exemplifica um caso prático de aplicação com a técnica Ideal Day:

Considera-se uma jornada de trabalho de uma equipe com 4 horas diárias, deve-se estipular um valor de referência, ou seja, 1 Ideal Day = 4 horas e determinar a velocidade média inicial desta equipe, como por exemplo, ID = 10 horas. Este último dado, sendo resgatado de um histórico ou inicialmente definido por um especialista de maneira empírica.

1) Segue abaixo a lista de itens do Backlog com suas estimativas de Ideal Day já atribuídas via Planning Poker:
    a. RF01 – Implementar carrinho de compras – **0,5 ID;**
    b. RF02 – Cadastrar Livros – **0,3 ID;**
    c. RF03 – Consultar livros por autor – **0,1 ID;**
    d. RF04 – Implementar recomendação automática de livros – **0,4 ID;**
2) Pelo conceito de pilha, cada membro deve retirar uma tarefa para executar e apropriar as horas gastas ao final do dia. É necessário para que, ao final do Sprint, seja possível determinar quantos Ideal Days foram concluídos e o seu tempo de execução.

O requisito, uma vez com seu valor definido, não poderá mais ser alterado, somente o esforço que depende do recurso alocado. Neste exemplo, o recurso 1 irá implementar o RF01, quem tem 0,5 ID. Como é a primeira vez, ainda não existem dados históricos para determinar o esforço do recurso, neste exemplo, utiliza-se empiricamente o valor de 10 horas. Por isto o requisito terá duração prevista de 5 horas (meio Ideal Day).

Esforço = tamanho x velocidade

Esforço = 0,5 x 10

Deve-se destacar que a tarefa deve ser concluída em um dia, caso não ocorra, será necessário ‘quebrar’ este requisito e mover o post-it concluído de ‘em andamento’ para ‘finalizado’ evidenciando assim o trabalho realizado.

A Tabela 1 exemplifica os dados apurados após implementação.

Requisito	Tamanho Previsto	Recurso	Horas Real
RF01	0,5 ID	Recurso 1	4,5 horas
RF02	0,3 ID	Recurso 2	2,5 horas
RF03	0,1 ID	Recurso 1	1,5 horas

RF04	0,4 ID	Recurso 2	3 horas
TOTAL	**1,3 ID**		**11,5 horas**

Tabela 1. Dados apurados.

De acordo com a Tabela 1, o recurso 1 implementou os requisitos RF01 e RF03, totalizando entrega de 0,6 ID. Já o recurso 2, os RF02 e RF04, totalizando 0,7 ID. Para calcular a nova velocidade, após o Sprint, devemos considerar os dados da Tabela 2.

	Velocidade Inicial	ID previsto	ID realizado	Horas Real
Recurso 1	10	0,6	0,6	6
Recurso 2	10	0,7	0,7	5,5

Tabela 2. Dados consolidados.

Para concluir o cálculo da velocidade é preciso considerar o tempo de retrabalho, ou seja, o tempo gasto em correções que terão um peso maior no cálculo da velocidade da equipe. Para isto, temos que a fórmula para cálculo da velocidade é:

*Horas_realizadas + (horas_retrabalho * 1,3) / ID realizados*

Com isto:

Recurso 1 = 6 + 0 / 0,6 = **10 horas.**

Recurso 2 = 5,5 + 0/0,7 = **7,8 horas.**

Média da equipe = **8,9 horas.**

No próximo Sprint, a média a ser considerada será a calculada no Sprint anterior (8,9 horas) e não mais o valor empírico de 10 horas.

A produtividade é extraída da avaliação do número de Ideal Days por Sprint. No primeiro Sprint, a produtividade é igual ao número de Ideal Days entregues.

Produtividade da equipe = **1,3 ID**

Considera-se a produtividade da equipe no próximo Sprint no qual será possível alocar itens do Backlog que totalizem o número de Ideal Days entregues no Sprint anterior. (Em nosso exemplo 1,3 ID)

Ao final deve-se apurar novamente este número e fazer a média entre Sprints. Informação esta que deverá ser constantemente atualizada.

Para conhecer a média de produtividade e velocidade da equipe, deve-se medir novamente e obter mais dados históricos.

Assim a produtividade da Release pode ser calculada como:

Produtividade da Release = ID Realizados / Número de Sprints.

De acordo com o exemplo, durante o Sprint 1 a equipe manteve a produtividade planejada de 1,3 ID. Caso, a realese tenha 3 Sprints e a equipe entregar 0,9 ID no segundo e 1,1 ID no terceiro, a produtividade média da release será:

1,3+0,9+1,1/3 = 1,1 ID.

3. Conclusão

Técnicas de abordagem de estimativas de software, tanto ágil como de desenvolvimento tradicional é de suma importância para se identificar o andamento e a saúde de qualquer projeto, trata-se de uma etapa fundamental antes do início do desenvolvimento e que deverá ser monitorada durante todo o ciclo.

No Scrum, o Scrum Master é o responsável por intermediar ambas técnicas apresentadas neste artigo, cada técnica tem suas características de obtenção do tamanho de um software, e a equipe como um todo deverá escolher a técnica que trabalhará. Técnica esta, que tende a ser desenvolvida pelo grupo através de seu conhecimento empírico.

Por fim, o artigo apresentou duas técnicas que podem ser aplicadas de forma a obter o tamanho de um software, ou proejto, em seu desenvolvimento ágil, o Planning Poker e o Ideal Days.

4. Referências Bibliográficas

BASSI, F. D. L., (2012) "Experiências com desenvolvimento Ágil", São Paulo, http://www.teses.usp.br/Dissertacao_Metodos_Ageis_Dairton_Bassi.pdf, Acesso em: 05 jun. 2014.

COHN, M. Agile Estimating and Planning, Prentice Hall, 1ª edição, 2005.

CTIC-UFPA. "*Procedimentos do Planning Poker*", versão 1.3. 2011. Centro de Tecnologia da Informação e Comunicação da Universidade Federal do Pará.

FONSECA, da S. W. D. Ferramenta para apoio à estimativa baseada em Planning Poker utilizando a metodologia Scrum, Julho 2008. UFPE – Recife. Disponível em: http://www.cin.ufpe.br/~tg/2012-1/dwas.pdf. Acesso em: 29 de abr. 2014.

GRENNING, J. Planning Poker or How to avoid analysis paralysis while release planning. 2002. Disponível em: http://renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf. Acesso em 02 set. 2013.

HAZAN, C., (2008) "Análise de Pontos de Função: Uma aplicação nas estimativas de tamanho de Projetos de Software", Engenharia de Software, Rio de Janeiro, p.25-30.

KNIBERG, H.; SKARIN, M., (2009) "Kanban and Scrum – Making the most of both" Estados Unidos da América. ISBN: 978-0-557-13832-6. http://www.infoq.com/br/minibooks/kanban-scrum-minibook. Acesso em 05 jun. 2014.

MARTINS, José Carlos Cordeiro. Técnicas para Gerenciamento de Projetos de Software. 1. ed. Florianópolis: Brasport Editora, 2007. 465 p.

MCCONNELL, S., Software estimation: Demystifying the black art, Microsoft Press, 2006.

PRESSMAN, R. Engenharia de Software. 3ª edição. Makron Books, 1995.

PUTMAN, L. H.;WARE M. Measures for Excellence: Reliable Software On Time, Within Budget, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1992.

SIMÕES, C. A., Sistemática de métricas, qualidade e produtividade, Technical report, BFPUG. 1999. Disponível em http://www.bfpug.com.br/Artigos/sistematica_metricas_simoes.htm. Acesso em 02 mai. 2014.

VALENÇA, A. Implantação de Processo de Estimativa de Esforço de Desenvolvimento de Software – Caso Prático. 2007. Tese de Mestrado, Universidade Federal de Pernambuco.