

Gestão Ágil de Projetos: Método para estimativa em projeto de software para fábrica de software acadêmica

Felix Soares Ferreira¹, Viviane Carla Batista Pocivi²

^{1,2}Bacharelado em Engenharia de Computação – Centro Universitário de Anápolis
(UniEVANGÉLICA) – Anápolis - GO

¹felix.soares.ferreira@gmail.com, ²viviane.batista@unievangelica.edu.br

Resumo. *A estimativa de tempo é uma das métricas de software mais utilizadas, pois a partir dessa dimensão é possível definir esforço, prazo e custo necessários para o desenvolvimento de software. Existem várias técnicas de estimativas ágeis de tempo e esforço, dentre elas a técnica PERT, utilizada nesta pesquisa. Neste artigo apresenta-se um estudo de caso realizado na Fábrica de Tecnologias Turing (FTT), uma fábrica de software acadêmica, com o fim de melhorar o processo de estimativa dos projetos de software. Para levantamento de dados e validação da pesquisa, o estudo de caso foi realizado em um projeto de grande porte, internacional e que estava em seu início no momento do levantamento dos dados. Neste artigo, será apresentado a metodologia, o ambiente do estudo de caso e os resultados obtidos.*

1. Introdução

A grande demanda por *software* de qualidade e desenvolvido em tempo hábil transformou o processo de se criar *software*, desde seu planejamento, gerenciamento até a entrega e manutenção. Atualmente as empresas que não se adequam, tanto estrategicamente quanto operacionalmente ao grande volume de informação, tendem a não cumprirem prazos, diminuem a qualidade do produto e geralmente não contam com bons relacionamentos com o cliente. Para Tavares, Carvalho e Castro (2004), estes podem ser alguns dos maiores desafios para as organizações que desenvolvem *software*.

Para se adequarem ao cenário atual as empresas tendem à adotar metodologias ágeis para gestão e desenvolvimento do projeto de *software*, buscando-se aproximar o cliente do processo de desenvolvimento e trabalhar com entregas mais curtas evitando-se retrabalho. A estimativa de tempo torna-se necessária desde o início do projeto, passando pela fase de desenvolvimento e testes. Este cenário tem exigido a adoção de técnicas para estimar tempo e esforço das atividades que envolvem a criação de sistemas, planejando-se produtividade, previsão e priorização [LEISHMAN 2008].

A Fábrica de Tecnologias Turing (FTT) foi o ambiente utilizado para validar o estudo de caso desta pesquisa. A FTT é uma unidade vinculada ao curso de Engenharia de Computação da UniEVANGÉLICA, cujo propósito é melhorar a formação profissional de seus acadêmicos. Para tanto, ela utiliza uma metodologia onde os envolvidos, sob orientação de docentes e funcionário técnico-administrativo, desenvolvem projetos de *software* reais. Assim, os alunos têm a oportunidade de vivenciar na prática tudo o que aprendem em sala de aula.

Neste contexto, por meio de uma pesquisa realizada entre membros dos times de projetos da FTT, observou-se que as equipes têm dificuldades em realizar estimativas, bem como de se manter a produtividade em consonância com as estimativas planejadas. O objetivo principal deste trabalho foi propor um método capaz de melhorar o processo de estimativas de tempo e esforço dos projetos da FTT. Com o método, a expectativa foi de melhorar as estimativas.

Nas seções seguintes apresenta-se o referencial teórico, com ênfase em métodos ágeis, estimativas e a técnica PERT. Na sequência, apresentam-se as análises e resultados, contendo a exposição do ambiente de estudo do caso e seu processo de desenvolvimento, a coleta e análise dos dados levantados pelo estudo de caso e os resultados obtidos. Por fim, são delineadas as conclusões e as referências bibliográficas.

2. Referencial Teórico

Nos itens a seguir serão apresentados os tópicos que correspondem à fundamentação teórica para este trabalho, a saber: Métodos Ágeis, Estimativas e a Técnica PERT.

2.1. Métodos Ágeis

Ao longo dos anos, vários métodos de desenvolvimento de *software* foram propostos. Entre eles, existem os chamados métodos ágeis [AMBLER 2002], também conhecidos por métodos leves [FOWLER 2000]. Em comparação aos métodos tradicionais, os métodos ágeis são caracterizados como mais adaptativos à realidade do time e flexíveis à mudanças. Além disso, eles são indicados para cenários em que os requisitos são instáveis, ou seja, constantes mudanças ocorrem na especificação dos requisitos e os resultados devem ser entregues em pequenos espaços de tempo.

Geralmente, esses métodos dividem o desenvolvimento em diversas iterações de ciclos mais curtos e realizam entregas ao final de cada uma delas, de forma que "[...] o cliente (interno ou externo) receba uma versão que agregue valor ao seu negócio." [DANTAS 2003]. No caso do *Scrum*, estes ciclos são chamados de *Sprints*. Assim, as mudanças de requisitos podem ser acompanhadas pela equipe no início de cada ciclo e ao final são verificados e validados os itens que foram desenvolvidos junto ao cliente, minimizando os erros de interpretação e implementação.

Esses métodos foram fortemente influenciados pelas melhores práticas da indústria japonesa, particularmente pelos princípios da manufatura enxuta implementados pelas companhias Honda e Toyota e pelas estratégias de gestão do conhecimento de Takeuchi e Nonaka (1986).

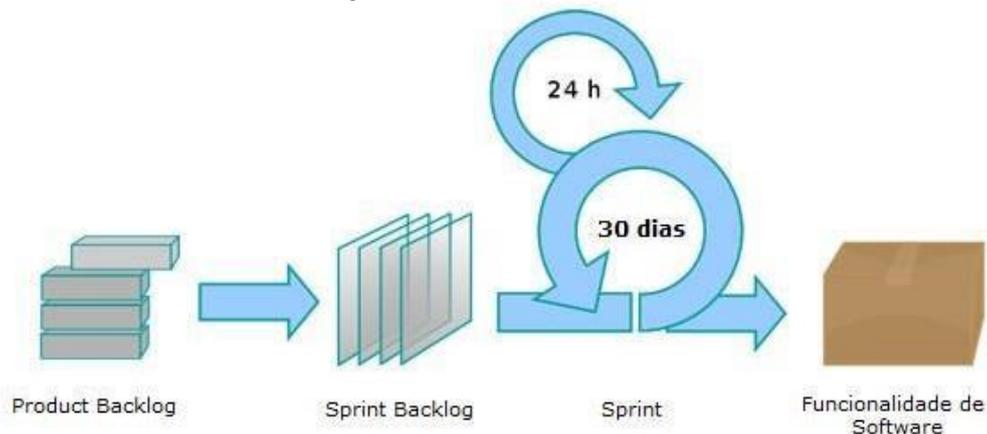
Os métodos tradicionais de desenvolvimento têm o foco na geração de documentação sobre o projeto e no cumprimento rígido de processos. Já os métodos ágeis concentram as atenções na constante entrega do produto e nas interações entre os indivíduos [AGILEMANIFESTO 2001]. Nestes métodos, é minimizada a fase de planejamento inicial total, de modo que os desenvolvedores se concentram em entregar o produto ao fim de cada iteração, ao invés de traçar diretrizes e planejamentos para o projeto como um todo.

Existem diversos métodos ágeis, além do *Scrum: Extreme Programming*, o *Feature Driven Development*, *Dynamic Systems Development Method*, *Adaptive Software Development*, *Crystal*, *Pragmatic Programming*. Para o escopo desta pesquisa, o *Scrum* se apresenta como principal fundamento. Isso se dá pelo fato do ambiente de estudo de caso utilizar um processo baseado no *Scrum*.

Scrum é uma das metodologias ágeis mais conhecidas atualmente, este modelo é resultado dos esforços conjuntos de especialistas da área de sistemas, com destaque especial para Ken Schwaber e Jeff Sutherland. Schwaber e Sutherland (2013) basearam-se no estudo de Takeuchi e Nonaka (1986), no qual notaram-se que projetos com equipes pequenas e multifuncionais obtinham melhores resultados. O termo denomina uma jogada do *rugby* em que os membros dos respectivos times se unem para colocar a bola em jogo após uma falta ou penalidade, e foi escolhido devido à semelhança entre os atletas e os desenvolvedores: cada grupo age como uma unidade integrada, onde cada elemento desempenha um papel específico e todos se ajudam em busca de um objetivo comum [CARVALHO, MELLO 2009].

Segundo Schwaber e Sutherland (2013), *Scrum* não é um processo ou técnica de produção; seu papel é fazer transparecer a eficácia de suas práticas de desenvolvimento para que seja possível melhorá-las, enquanto provê um framework dentro do qual produtos complexos podem ser desenvolvidos. Por ser fundamentado na teoria de controle de processos empíricos, emprega uma abordagem iterativa e incremental, conforme a Figura 1, para otimizar a previsibilidade e controlar riscos. Portanto, o *Scrum* concentra-se em “como” as equipes devem realizar suas tarefas de forma ordenada para produzir um sistema flexível num ambiente com constantes mudanças, descreve Zanatta (2004).

Figura 1. Processos do Scrum.



Fonte: http://videos.web-03.net/artigos/Renato_Groffe/Desenvolvimento_Agil_Scrum/Desenvolvimento_Agil_Scrum1.jpg

O *Scrum* resumidamente pode ser dividido em quatro etapas: *Product Backlog*, *Sprint Backlog*, *Sprint* e Entrega da funcionalidade. No processo denominado *Product Backlog* realiza-se o contato com o cliente para entender o sistema a ser realizado, coletar requisitos, especificações etc. Ao final desse processo tem-se uma lista de funcionalidades. A partir dessa lista de funcionalidades pode-se dar início o próximo processo, denominado *Sprint Backlog*, onde irá ocorrer o refinamento dessas funcionalidades para serem desenvolvidas durante a *Sprint*. Neste processo ocorrem as estimativas, a separação das funcionalidades para cada integrante do time e ao final, tem-se uma lista de funcionalidades da *Sprint* estimada.

Em seguida, inicia-se a *Sprint*, este é o ciclo de iterações para o desenvolvimento das funcionalidades, dividida em 2 (duas) a 4 (quatro) semanas. Ao iniciar o dia durante este processo ocorre o *Daily Meeting*, uma reunião diária para que ocorra transparência entre os integrantes do time. A reunião é realizada em pé e recomenda-se que não passe de 15 (quinze) minutos, o *Scrum Master* questiona cada integrante com as seguintes questões: O que foi feito ontem? O que irá fazer hoje? Teve algum impedimento? Ao final da *Sprint* espera-se que tenha uma ou várias funcionalidades prontas e o processo de Entrega consiste justamente nessas funcionalidades validadas e entregues para o cliente.

2.2. Estimativa

Estimativa é um valor aproximado, dado através de cálculos ou não, que se realiza sobre alguma coisa e que pode ser usado como informação. Mesmo uma estimativa sendo incompleta ou incerta poderá ser utilizada, pois quando se utiliza estimativas deve-se considerá-las com uma margem de erro, seja ela pequena ou grande [PRIKLADNICKI, WILLI, MILANI 2014].

Segundo Pressman (2006), estimativa pode ser considerada como a tentativa de determinar quanto dinheiro, esforço, recursos e tempo serão necessários para construir um sistema ou produto.

Estimativas, quando bem elaboradas, tornam-se base para um bom planejamento. Segundo Leishman (2008) existem três motivos para equipes ágeis estimarem: previsão, performance e priorização. Ou seja, prever o quanto a equipe irá gastar de tempo e esforço, identificar perdas ou ganhos de desempenho durante o desenvolvimento baseado no histórico do time e informar o custo de implementação de modo que consigam priorizar os itens a agregar valor para o cliente.

Estimar *software* nunca foi uma tarefa muito fácil [PRESSMAN 2006], pois na maioria dos casos as equipes realizam previsões quantitativas tendo somente informações qualitativas. Contudo, estimar não precisa ser de forma aleatória, pois existem técnicas disponíveis na literatura que podem ser utilizadas. Segundo Pressman (2006): “Apesar de estimar ser tanto arte como ciência, essa importante atividade não precisa ser conduzida de modo aleatório. Já existem técnicas úteis para estimativa de tempo e esforço”.

Existem várias técnicas de estimativas ágeis que variam em seus resultados, formas e aplicações. A forma adotada pelas organizações pode variar de acordo com a metodologia utilizada e o tipo de projeto. É desejável que tenha conhecimento prévio da técnica e uma visão macro do escopo do projeto, e quando possível uma base histórica onde seja possível consultar as estimativas em atividade semelhantes já finalizadas [SOMMERVILLE 2003].

2.3. Técnica PERT

Segundo Sanches (2013), a técnica PERT (*Program Evaluation and Review Technique*) consiste em calcular a duração de desenvolvimento de uma atividade baseando-se em três estimativas: Otimista (O), Pessimista (P) e Mais Provável (MP).

Esta técnica considera que Otimista é o cenário perfeito, onde tudo dará certo, Pessimista é o pior cenário, onde tudo dará errado e o Mais Provável é o cenário razoável, onde tudo ficará na normalidade sem grandes surpresas.

A combinação dessas três possibilidades é o grande diferencial da técnica segundo Sanches (2013), pois ela pondera as incertezas e riscos envolvidos na atividade, podendo ser aplicada em times com poucas e grandes experiências.

A fórmula para se calcular as estimativas utilizando a técnica PERT é dada por:

$$PERT = \frac{(P + 4MP + O)}{6}$$

onde: *PERT* = valor da Estimativa; *P* = estimativa Pessimista; *MP* = estimativa Mais Provável; *O* = estimativa Otimista.

Esta fórmula aplica um peso maior para as estimativas Mais Prováveis, pois é a estimativa mais próxima do que seria realizado, porém não deixa de considerar as estimativas Pessimistas e Otimistas. Pode-se inclusive alterar os pesos entre as estimativas de acordo com o projeto, mas os pesos não podem ultrapassar a soma de 6 [SANCHES 2013].

A técnica PERT também considera o desvio padrão, que serve para medir o quanto de variação ou dispersão existem em relação à média, a fórmula do desvio padrão é dada por:

$$Desvio\ Padr\tilde{a}o = \frac{(P - O)}{6}$$

onde: *Desvio Padrão* = variação em relação aos pontos de vista Pessimista e Otimista; *P* = estimativa Pessimista; *O* = estimativa Otimista.

Em termos estatísticos os três pontos de vista (otimista, mais provável e pessimista) correspondem a uma distribuição triangular. Conforme citado anteriormente, a fórmula da estimativa PERT dá-se por uma média ponderada com peso 4 para estimativa mais provável. Esta

equação se explica no conceito estatístico da distribuição beta, o qual não será detalhado neste artigo, mas em síntese permite transpor a Distribuição Triangular (contínua) em uma Distribuição Normal.

3. Análises e Resultados

Nos itens a seguir serão apresentados os tópicos que correspondem ao ambiente de Estudo de Caso, as coletas e análise de dados e os resultados.

3.1. Ambiente de Estudo de Caso

A Fábrica de Tecnologia Turing é um laboratório-empresa que dá a oportunidade aos acadêmicos do Curso de Engenharia de Computação da UniEvangélica de vivenciarem e colocar em prática o conhecimento adquirido em sala de aula. Orientado por professores e sob a direção do curso, a FTT simula um ambiente de uma fábrica de *software*, com padrões bem definidos, processos e postura profissional. Assim, os alunos têm a oportunidade de adquirir experiências profissionais em desenvolvimento, testes, gerenciamento e outros perfis profissionais da produção de tecnologias.

Os projetos que são desenvolvidos na FTT são reais, com clientes e com a proposta de resolver problemas para os quais a tecnologia é a solução. Os projetos são tanto para os setores e cursos da UniEvangélica, como para clientes externos. Em seu portfólio, a FTT já conta com 13 projetos na área de desenvolvimento.

O processo de gestão e desenvolvimento utilizado durante as execuções dos projetos é um processo híbrido formado por conceitos do *OpenUP* e do *Scrum*. Esse processo é apresentado na Figura 2.

Figura 2. Processo híbrido utilizado na Fábrica de Tecnologia Turing.



Fonte: FTT (2016)

O processo da FTT é subdividido em etapas. São elas: Definição da Visão, Desenvolvimento do *Backlog* do Produto, Reunião de Planejamento da *Sprint*, entre outras.

Neste processo há papéis e responsabilidades distintas, a saber: docente, líder técnico, *Scrum Master*, *Product Owner* (PO) e o time. Um docente é o orientador do projeto e tem a função de conduzir o time em todas as etapas do processo de desenvolvimento do projeto. O líder técnico é o responsável por ajudar o time no que diz respeito à infraestrutura e desenvolvimento. O *Scrum Master* é o *coach* tanto do time quanto do PO; ele ajuda todos a compreenderem os valores, princípios e práticas do *Scrum*, resolver impedimentos, evitar interferências e é um agente de mudanças [SCHWABER, SUTHERLAND 2013]. O PO é o que mais tem contato com o cliente, tem a responsabilidade de definir os *products backlog* e priorizá-los [SCHWABER,

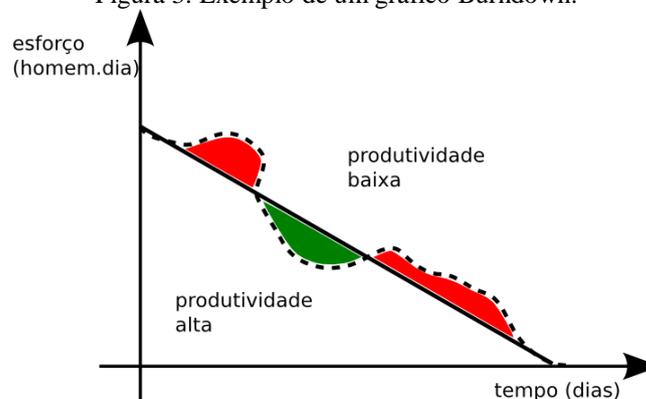
SUTHERLAND 2013]. Já o time é composto por desenvolvedores, testadores e o pessoal de requisitos. Destacam-se no processo de estimar as atividades o *scrum master* que irá comandar o processo de estimar e o integrantes do time que planejarão as estimativas de suas atividades.

A primeira etapa do processo da FTT é a Definição da Visão. Nela, todos os papéis participam e é onde ocorre o primeiro contato com o cliente para dar início ao planejamento do projeto. Alguns atrefatos são produzidos, tais como: diagrama de casos de uso, projeto de banco de dados e o próprio documento de visão. Seguindo, há a criação dos *products backlogs* pelo PO junto aos docentes, criação das especificações de caso de uso, aprovação das interfaces de usuário, documento de arquitetura, criação do gráfico burndown entre outros. A reunião de planejamento ocorre em duas partes (Seção 3.2), na primeira há a discussão do que será desenvolvido mediante aos conhecimentos da equipe e a priorização do PO [SCHWABER, SUTHERLAND 2013]. Todos participam dessa reunião. A segunda reunião só não participa o PO e é discutido como será desenvolvido e aqui entra o processo de estimativa para a *sprint* a fim de gerar o cronograma e um documento sobre os itens que serão desenvolvidos (*sprint backlog*) [SCHWABER, SUTHERLAND 2013].

Após ter os itens do *product backlogs* priorizados, estimados e entendidos por todos do time, e o *sprint backlog* documentado, dá-se início à *Sprint* [SCHWABER, SUTHERLAND 2013]. Um ciclo de iteração dividido em semanas (*time-box*) que permite ao time desenvolver as funcionalidades propostas no planejamento da *Sprint* [SCHWABER, SUTHERLAND 2013]. Ao início de cada dia durante a *Sprint* o *Scrum Master* se reúne com o time para remover impedimentos, tirar dúvidas e compartilhar os acertos com o time [SCHWABER, SUTHERLAND 2013]. Utiliza-se três perguntas: O que fiz ontem? O que irei fazer hoje? Há algum impedimento? O gráfico de *burndown* é atualizado.

O gráfico de *burndown* (Figura 3) é um artefato que compara o trabalho planejado, feito, concluído e o trabalho restante em um gráfico. No eixo vertical, o somatório de pontos de esforço necessários para realizar as atividades da iteração atual e, no eixo horizontal, os dias de trabalho da iteração, traça-se uma reta diagonal onde liga o topo do eixo vertical (total de pontos) ao limite do eixo horizontal (último dia de trabalho) [SCHWABER, SUTHERLAND 2013]. Se um desvio acima da reta ideal ocorre, significa que a equipe está produzindo menos do que deveria e que, por isso, pode não acabar as tarefas designadas para a iteração até o seu final [SCHWABER, SUTHERLAND 2013]. Por outro lado, se o desvio ocorre abaixo da reta ideal, a equipe está produzindo a mais do que o previsto, o que pode significar uma subestimação da produtividade da equipe ou uma superestimação de complexidade das tarefas designadas para a iteração [SCHWABER, SUTHERLAND 2013].

Figura 3. Exemplo de um gráfico Burndown.



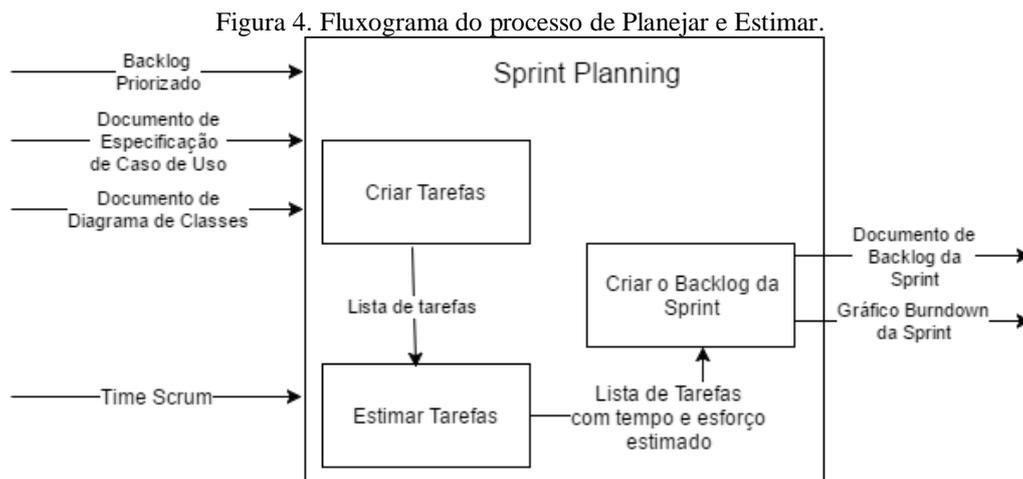
Fonte: Autores.

Ao final de cada *Sprint* tem início a revisão do que foi produzido (*software* e documentos) durante todas as iterações e todos participam. Em seguida todos irão trocar informações sobre o que funcionou e o que não funcionou a respeito do processo, para que na próxima iteração, reduza as chances de ocorrer problemas e, assim, de melhorar continuamente o processo [SCHWABER, SUTHERLAND 2013].

3.2. Planejar e Estimar

O processo de Planejar e Estimar é relacionado com os processos de planejamento e estimativas de tarefas. Dentre os processos de planejamento e estimativas considerados pelo Guia para o Conhecimento em Scrum [SBOK 2016], o time da FTT utiliza os seguintes processos (Figura 4):

- Criar as Tarefas: Nesse processo os documentos de Especificação de Caso de Uso, Diagrama de Classes e *Product Backlog* são utilizados para criar uma lista de tarefas de acordo com a priorização dos *backlogs*.
- Estimar as Tarefas: Nesse processo, o time scrum, estima o tempo e o esforço necessário para realizar cada tarefa da lista de tarefas, o resultado desse processo é a lista de tarefas com esforço e tempo estimados.
- Criar o *Backlog* da *Sprint*: Nesse processo acontece a organização dos itens que serão desenvolvidos durante a *sprint*.



Fonte: Autores.

Para que se consiga criar a lista de tarefas de forma coerente, segundo o SBOK [SBOK 2016], o time da FTT utiliza as seguintes ferramentas:

- Decomposição: Utilizada na divisão de tarefas de complexidade alta, em tarefas mais detalhadas, de complexidade mais baixa;
- Determinação de Dependência: Uma vez que tenha a lista de atividades selecionada o time separa as atividades entre as equipes considerando as dependências das atividades, disponibilidade de pessoas, comprometimento e habilidades técnicas para realização da atividade;

3.3. Metodologia, Coleta e Análise de dados

Pode-se classificar o presente trabalho como de natureza aplicada, devido ao interesse prático na aplicação de seus resultados e na resolução de problemas reais. Quanto aos seus procedimentos, a presente pesquisa utilizou o estudo de caso. Segundo Zelkowitz e Wallace (1998), estudo de caso é

um estudo observacional em que o pesquisador monitora projetos em profundidade e coleta dados ao longo do tempo.

Para coletar dados foram feitas entrevistas e um questionário com alguns alunos estagiários que trabalham na própria Fábrica de Tecnologia Turing. Neste questionário foi feita perguntas tanto relacionadas à estimativa, quanto à perspectiva pessoal e profissional dos estagiários, bolsistas e voluntários. As perguntas relacionadas às estimativas foram feitas com o fim de avaliar, na perspectiva de cada entrevistado, o nível de conhecimento sobre o que é estimativa, a sua importância e como é aplicada nos projetos de software na FTT. Segundo Andrade (2004), saber sobre o lado pessoal e profissional é ponto chave no processo de estimativa, pois as estimativas de esforço basicamente são dadas através dessas áreas, saber sobre motivações, objetivos, dificuldades e conflitos ajuda a saber sobre o perfil de cada entrevistado.

Foram entrevistados desenvolvedores, analistas e orientadores, totalizando 10 pessoas de um grupo de 19 pessoas, para se ter uma maior cobertura e visão de como a estimativa funciona para cada área do processo de desenvolvimento.

Em cada projeto, em cada etapa do processo analisado e em cada papel entrevistado foi constatado que, apesar de definido no processo, não havia a utilização sistematizada de método ou técnica para extrair as estimativas das atividades. Mesmo que haja um campo na documentação, especialmente para relatar a estimativa daquela atividade, esta não era preenchida como deveria ser. Quando era preenchida, a informação se baseava unicamente na experiência vivida. Sabe-se que técnicas que dependem somente de métricas históricas tendem a funcionar só quando o processo é estável, já possuir informações colhidas ao longo do tempo e que todos os dados sejam cuidadosamente guardados para serem analisados e validados.

Outro aspecto levantado durante as entrevistas foi a falta de maturidade de alguns integrantes das equipes. Falta de maturidade tanto na própria execução de seu papel, quanto nas estimativas quando estas são realizadas. Não foram detectados problemas relacionados à desavença com os próprios companheiros de equipe, nem com os orientadores e a direção da FTT. Contudo, houve a constatação de que há um bom nível de conhecimento sobre estimativas por parte dos entrevistados, levando a se questionar sobre o porquê deste conhecimento não estar sendo aplicado ou explorado.

O projeto selecionado para a realização do estudo de caso desta pesquisa foi o SGafa (Sistema de gerenciamento acadêmico e financeiro de Angola). A escolha se deu por alguns fatores: é o único projeto que está em fase inicial; possui características de complexidade relevantes para fundamentar os dados da pesquisa; a equipe é composta por profissionais com diferentes níveis de conhecimento e experiência; projeto de escopo internacional. Os demais projetos da FTT atualmente encontram-se finalizados ou em fase de manutenção.

O SGafa pretende desenvolver um sistema para gestão de processos acadêmicos e financeiro, para as instituições ISPVIDA (Instituto Politécnico Vida de Angola) e ISTEEL (Instituto Superior Teológico em Lubango), ambos localizados na Angola, continente Africano.

Para este projeto houve uma reorganização das equipes da FTT, onde alguns membros saíram e novos membros chegaram. No período da coleta dos dados, a equipe do projeto estava composta por 19 pessoas. Por ser um projeto com um grau de complexidade maior, a arquitetura e as tecnologias envolvidas no desenvolvimento foram atualizadas. Para tanto, antes de dar início ao SGafa houve um processo seletivo e o time passou por um período de treinamento.

Como visto anteriormente estimar nunca foi uma tarefa fácil para indústria de *software* [PRESSMAN 2006], e se torna ainda mais difícil com times inexperientes e com instabilidades de conhecimento acerca de arquitetura e linguagens de programação. Portanto a técnica de estimativa ágil a ser levantada deveria ser de fácil entendimento para todos os integrantes, considerar as

incertezas do projeto, ser de fácil aplicação e não ocupar muito tempo, visto que há variações de horas trabalhadas entre integrantes do time. Os estagiários trabalham 6 horas/dia, bolsistas 5 horas/dia e os voluntários 4 horas/dia.

Ao analisar várias técnicas de estimativas ágeis chegou-se à conclusão, junto com envolvidos no projeto, que a técnica PERT seria a mais adequada nestas primeiras *Sprints*, pois com ela poderia avaliar três pontos de vista para uma estimativa: Otimista, Mais Provável e Pessimista. Assim, com essa técnica poderia se considerar os níveis de incerteza de entendimentos de requisitos e de impedimentos nas tecnologias sem deixar de observar a produtividade. Por ser uma técnica estatística, o PERT não deixa de avaliar o conhecimento histórico, pois para recolher as estimativas Otimistas, Mais Prováveis e Pessimistas é necessário olhar para projetos ou bases históricas já desenvolvidas. Portanto, para os envolvidos poderem estimar, foi considerado as experiências durante o treinamento como base para as estimativas, visto que os itens selecionados para a primeira *Sprint* eram semelhantes aos itens desenvolvidos nos treinamentos.

Antes de estimar as atividades, foi ministrada uma breve formação sobre a importância das estimativas e sobre a técnica PERT. Após todos entenderem como funcionava a estimativa com a técnica PERT, deu-se início ao processo de estimar as atividades. As estimativas foram separadas em quatro grupos, da mesma forma em que as equipes foram organizadas para compor os times. A primeira equipe a estimar foram os responsáveis pelo *front-end*, seguida pela equipe de *back-end*, a equipe de documentação e requisitos e por último a equipe de testes.

Dentro de cada grupo, os integrantes já haviam selecionado as atividades de acordo com a competência de cada um. Foi questionado sobre as estimativas de tempo e esforço nas perspectivas Otimistas, Mais Prováveis e Pessimistas, para cada uma das atividades selecionadas, conforme indica a Tabela 1. O processo se repetiu para as demais equipes.

Tabela 1. Estimativas de uma determinada equipe.

Responsável	Atividade	Otimista	Mais Provável	Pessimista	Estimativa Real	Total	Desvio Padrão
Responsável A	Atividade 1	0,83	1,2	1,5	1,19	7,28	-0,11
	Atividade 2	0,83	1,2	1,5	1,19		-0,11
	Atividade 3	1,4	2,2	2,4	2,1		-0,167
	Atividade 4	1	1,4	2	1,43		-0,17
	Atividade 5	1	1,3	2	1,37		-0,17
Responsável B	Atividade 6	1,3	2	2,3	1,93	5,72	-0,17
	Atividade 7	1,3	1,4	2	1,48		-0,12
	Atividade 8	1	1,4	1,2	1,3		-0,03
	Atividade 9	1,3	2	2,3	1,93		-0,17
	Atividade 10	0,83	1	1,2	1,005		-0,06
	Atividade 11	0,83	1	1,2	1,005		-0,06
Responsável C	Atividade 12	2,3	3,2	4	3,18	8,60	-0,28
	Atividade 13	2,3	3,2	4	3,18		-0,28
	Atividade 14	0,5	1	0,67	0,86		-0,03
	Atividade 15	1	1,3	2	1,367		-0,17

Fonte: Autores.

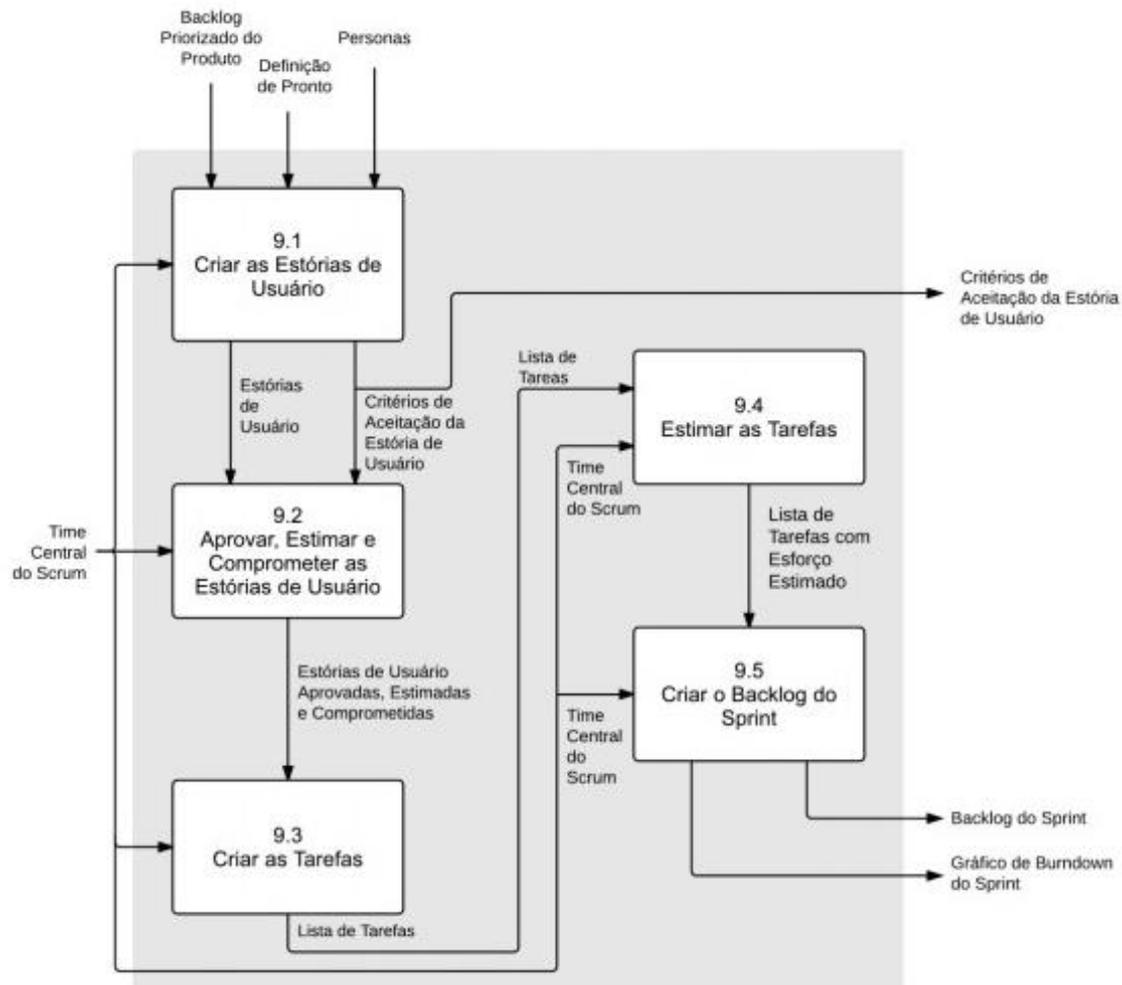
3.4. Resultados e Conclusões

O objetivo deste trabalho foi melhorar as estimativas de tempo e esforço dos projetos de desenvolvimento de *software* realizados pela FTT, com a expectativa inicial de se propor um método para tal melhora. Fez-se necessário o estudo de técnicas de estimativas ágeis, do processo utilizado pela FTT, do conhecimento do perfil dos envolvidos e dos projetos desenvolvidos nesse ambiente.

Ao se aprofundar no processo da FTT, percebeu-se que já constava itens para a estimativa, embora não utilizados adequadamente pelas equipes. Essa pesquisa sistematizou o que a FTT já

possuía em seu processo, conforme apresentado na Figura 5 e acrescentou a técnica PERT, que se mostrou adequada para este ambiente.

Figura 5. Processo de estimativa de acordo com SBOK.



Fonte: SBOK.

Como foi constatado no levantamento de dados que muitos integrantes não tinham consciência da relevância de se estimar e de se registrar os dados do estimado e do executado. Como resultado desse trabalho foi realizado um treinamento aos integrantes objetivando conscientizá-los do papel da estimativa em projetos de software, bem como propiciar o conhecimento adequado sobre o uso da técnica PERT.

Após todas as equipes estimarem houve um período para análise das estimativas coletadas e percebeu-se que a maioria das estimativas foram consideravelmente subestimadas. Sabia-se que as estimativas poderiam ser estimadas um pouco altas ou um pouco baixas. Porém, o desvio foi muito maior do que o esperado. Percebeu-se que essa ocorrência se deu em alguns casos pela falta de experiência do membro em estimar, em outros pela falta de experiência com as tecnologias envolvidas e em outros pela falta de comprometimento.

As estimativas realizadas pela equipe de *front-end* foram as estimativas com menores valores, as mais subestimadas. Houve caso de se estimar atividades com o prazo de 20 (vinte) minutos para serem concluídas e mesmo sob orientações os membros resistiram e não alteraram a estimativa. Como resultado, o estimado não foi realizado e o tempo final da atividade ficou em 5 horas. Considerando que a *Sprint* tem em média 75 (setenta e cinco) horas e levando em

consideração que os colaboradores das equipes não trabalham a mesma quantidade de horas por dia, o somatório das estimativas não chegou à metade da quantidade de horas de duração da *Sprint*.

A equipe de teste é a equipe que tem mais atividades, pois devem realizar Casos de Teste, Testes Funcionais e Reteste para cada atividade da *sprint backlog*, de todos os projetos da FTT. É a equipe que possui mais integrantes, porém com menor experiência, pois a maioria dos integrantes dessa equipe são novatos, sendo apenas o líder mais experiente.

Alguns membros dessa equipe não souberam estimar um valor para término das atividades, mesmo baseando no treinamento que tiveram e mesmo sendo encorajados a estimar. Logo nessa equipe houve a intervenção do líder, ou seja, ao solicitar uma estimativa Otimista, Mais Provável e Pessimista para um determinado integrante que não sabia qual valor estimar, o líder analisava a atividade e estimava para este integrante, portanto a atividade de um integrante que não soubera estimar quem realizava a estimativa era o líder.

Para as equipes de desenvolvimento o fator que mais atrapalhou tanto nas estimativas quanto na implementação foi a falta de conhecimento prévio sobre as tecnologias necessárias para o desenvolvimento de suas funções. Como se trata de uma fábrica de software acadêmica, na qual um dos seus objetivos é melhorar a qualificação profissional de seus envolvidos, essa é uma situação que comumente pode ser encontrada nesse ambiente. Nesta equipe, ocorreram vários impedimentos em relação à arquitetura do projeto, pois avaliaram uma tecnologia para ser utilizada e ao início do desenvolvimento percebeu-se que esta não seria adequada por não se integrar adequadamente a outras tecnologias. Sugere-se que tenha um melhor planejamento antes de começar as *Sprints* tendo em vista acompanhar e monitorar riscos desenvolvendo um plano de ação para que os impactos não sejam tão grandes.

Como visto anteriormente, as estimativas de todas as equipes foram abaixo do esperado e mesmo com atrasos e impedimentos no desenvolvimento a *Sprint* não foi cancelada, somente o objetivo foi alterado. Se anteriormente o objetivo era entregar todas as atividades finalizadas ou funcionais, o objetivo passou a ser de inspeção sobre o que funcionou e o que não funcionou a respeito das tecnologias envolvidas.

Em relação a maturidade ao estimar o time se comportou diferente do que foi relatado nas entrevistas, ou seja, ao se adotar uma técnica de estimativa ágil que seja rápida, de fácil entendimento para todos e que seja efetivamente acompanhada e não somente coletar valores aleatórios, percebeu-se uma mudança de comportamento.

Manter as equipes motivadas é fator primordial para que todos estejam motivados a melhorar tanto o processo quanto sua atuação individual.

Ao final deste levantamento, percebe-se que o processo adotado pela FTT prevê estimativas de tempo nas etapas de definição visão e planejamento da *Sprint*. Como contribuição dessa pesquisa, a técnica PERT foi inserida no processo e o método para se utilizá-la foi conscientizado e sistematizado, conforme apresenta a Tabela 1. Ressalta-se que o foco desta pesquisa está nos aspectos que influenciam a produtividade das equipes, limitando-se nas estimativas de tempo e esforço para desenvolvimento das atividades selecionadas na etapa de planejamento para *Sprint*. Outro ponto a se destacar é que na coluna Técnica, foi sugerido outros métodos de estimativa como ponto de referência para as equipes da FTT. Espera-se que os dados que constam na Tabela 2 promovam maior visibilidade sobre as técnicas de estimativa ao longo de todo o processo de gestão e desenvolvimento de sistemas.

4. Conclusões

A técnica PERT aborda pontos importantes nas estimativas, tal como os riscos e tem a finalidade de direcionar, viabilizar e ajudar a controlar os ciclos de desenvolvimento de *software*.

Tabela 2. Relação de etapas do processo, possibilidades de estimativas e suas respectivas técnicas.

Etapa	Necessita Estimativa?	Tipo de Estimativa	Técnica
Definição da Visão	Sim	Tempo; Tamanho; Custo	Análise de Pontos de Função; Pontos por caso de Uso; COCOMO;
Desenvolvimento do Backlog do Produto	Não		
Reunião de Planejamento da Sprint	Sim	Tempo; Esforço;	Planning Poker; Ideal Day; Análise de Pontos de Função; Pontos por caso de Uso; COCOMO; PERT;
Desenvolvimento do Produto	Não		
Revisão	Não		
Retrospectiva	Não		

Fonte: Autores.

O sucesso de um processo de estimativas depende de uma série de fatores externos, conhecimento prévio sobre técnicas de estimativas, conscientização dos envolvidos, manutenção e monitoramento do processo entre outros fatores. Tais fatores como conhecimentos sobre as tecnologias envolvidas e entendimento dos requisitos. Este trabalho foi realizado em um momento bastante inicial do projeto, onde alguns dos envolvidos no projeto não tinham experiências, as tecnologias envolvidas para o desenvolvimento estavam em testes e alguns requisitos ainda não tinham sido bem definidos. Mesmo que as tecnologias envolvidas foram as mesmas apresentadas no treinamento não foram suficientes para simular um ambiente real de produção.

Estimar traz benefícios para os envolvidos no projeto à medida que possibilita dimensionar melhor a carga de trabalho de forma a garantir melhor a qualidade e a performance. Com estas informações o time poderá requerer mais recursos, seja ele tempo ou de tecnologias, além de estimar com maior precisão o tempo necessário para desenvolver suas atividades. Para a gestão, terá informações de como as equipes estão se saindo, bem como estabelecer cronogramas com entregas em prazos possíveis.

Conforme Andrade (2004), a confiança nas estimativas aumenta à medida que o projeto é desenvolvido, pois os envolvidos obtêm informações que antes não eram muito claras ou não tinham informações suficientes. Este artigo foi elaborado para orientar profissionais e iniciantes tanto nas áreas de desenvolvimento de *software* quanto nas áreas de gestão de projetos sobre a importância de estimar através da proposta de melhorar um processo existente onde havia falhas e/ou não era executado de maneira correta.

Foi constatado que ao se adotar o uso formal de técnica de estimativa ágil houve um crescimento na maturidade em alguns integrantes da equipe, visto que com a adoção de uma técnica de estimativa foi possível cobrar pelos rendimentos de todos os integrantes das equipes. Houve também uma motivação para estimar para as próximas *Sprints*, conscientização sobre o processo de estimar e uma melhoria na documentação das estimativas. Para o lado profissional do time saber sobre estimativas, processo de estimar e sua importância é muito valioso, visto que muitas empresas utilizam metodologias para gerenciar seus projetos e a gestão de tempo e esforço estão presentes na maioria dessas metodologias.

Como sugestão de trabalhos futuros, as equipes da FTT podem aplicar outras técnicas de estimativas ou junções de duas ou mais técnicas com o fim de se obter dados históricos para comparações. O processo poderá ser ajustado à medida que as equipes adquirirem experiências com as técnicas e com o desenvolvimento dos projetos. Também é importante ressaltar que é necessário

ter uma boa gestão de configuração para poder armazenar essas informações, pois assim será possível melhor utilizar técnicas que se baseiam em dados históricos.

Referências

- AGILEMANIFESTO (2001). “Manifesto for Agile Software Development”, <http://agilemanifesto.org>, Outubro.
- Ambler, S. (2002). Agile modeling. New York: Wiley Computer Publishing.
- Andrade, E.L.P., Oliveira, K.M. (2004). Uso Combinatório de Análise de Pontos de Caso de Uso na Gestão de Estimativas de Tamanho de Projeto de Software Orientado a Objetos. In III Simpósio Brasileiro de Qualidade de Software. Brasília.
- Carvalho, B.V., Mello, C.H.P. (2009). Revisão, análise e classificação da literatura sobre o método de desenvolvimento de produtos ágil Scrum. In: Simpósio de Administração da Produção, Logística e Operações Internacionais – SIMPOI, 12., 2009, São Paulo. Anais... São Paulo.
- Dantas, V.F. (2003). Uma metodologia para o desenvolvimento de aplicações Web num cenário global. 2003. Dissertação (Mestrado) - Centro de Ciências e Tecnologia, Universidade Federal de Campina Grande, Campina Grande.
- Fowler, M. (2000). “Put your process on a diet. Software Development”, <http://www.drdoobs.com/put-your-process-on-a-diet/184414675>, Outubro.
- Sanches, L.A., Fernandes, N.T. (2013). Pert/Cpm Probabilístico Utilizando a Simulação de Monte Carlo, FATEC Bragança Paulista;
- Schwaber, K., Sutherland, J. (2013). “The Scrum Guide of Scrum.org”, <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf>, Setembro.
- Sommerville, I. (2003). Engenharia de software. 6º ed. Tradução Maurício de Andrade. São Paulo: Ed Addison-Wesley.
- SBOK. (2016). Um Guia para o Conhecimento em Scrum. Título original: A Guide to the SCRUM Body of Knowledge.
- Leishman, C. (2008). “The 3 Ps of Estimation: Why we estimate in Agile projects”, <https://medium.com/@cleishm/the-3-ps-of-estimation-aee352cf9b16#.7sa58argz>, Agosto.
- Pressman, R.S. (2006). Engenharia de software. 6. ed. São Paulo: Mcgraw-hill.
- Prikladnicki, R., Willi, R., Milani, F. (2014). Métodos Ágeis para Desenvolvimento de Software. Porto Alegre: Bookman. 289 p.
- Takeuchi, H., Nonaka, I. (1986). The new Product development game. Harvard Business Review.
- Tavares, H.C.A.B., Carvalho, A.E.S., Castro, J.F.B. (2004). Medição de Pontos por Função a Partir da Especificação de Requisitos. Pernambuco.
- Zanatta, E.M. (2004). Planejamento de práticas pedagógicas inclusivas para alunos surdos numa perspectiva colaborativa. 2004. Tese (Doutorado em Educação Especial). Universidade Federal de São Carlos.
- Zelkowitz, M.V., Wallace, D. (1998). Experimental models for validating computer technology. IEEE Comput. 31 (5), 23–31.