

# PROPOSTA DE ADEQUAÇÃO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE PARA A MELHORIA DA ESTIMATIVA DE HORAS

Evandro Franzen<sup>1</sup>, Helena Barden<sup>2</sup>

**Resumo:** Este artigo trata da análise e proposição de melhorias nos processos de desenvolvimento de *software* da empresa Interact Solutions Ltda. O trabalho desenvolvido visa a melhorar a estimativa de horas dos projetos e também identificar uma forma de mensurar as atividades dos colaboradores envolvidos no processo. A evolução da tecnologia da informação, aliada ao crescimento da organização, redundando em maiores desafios no gerenciamento e execução dos projetos. O desenvolvimento das atividades se deu por meio de análise de documentos fornecidos pela empresa, dos processos atualmente em execução, de observações e entrevistas informais com gerentes, analistas e desenvolvedores. Por meio da fundamentação teórica relacionada à Análise de Pontos de Função e Gerência de Projetos, foram propostas adequações no fluxo das atividades e foi introduzida uma técnica para mensurar a produtividade. O presente artigo descreve as observações relacionadas ao processo atual e as modificações propostas.

**PALAVRAS-CHAVE:** Projeto. Estimativas. Produtividade. Processo.

## 1 INTRODUÇÃO

As diversas demandas do mercado consumidor vêm exigindo, por parte das empresas, uma atuação flexível, ágil e eficiente, sendo essencial a implementação de novas ferramentas que visam ao aprimoramento da gestão de projetos.

O termo Gerenciamento de Projetos não é um conceito novo, porém vem evoluindo muito nos últimos anos, alcançando patamares mais sofisticados. Aspectos como prazo e custo, que antigamente eram secundários, hoje são essenciais para que o projeto tenha sucesso.

Segundo Keelling (2002, p. 3), a definição para Gerenciamento de Projetos adotada pelo Project Management Institute (Instituto de Administração de Projetos) – PMI é: “Um esforço temporário empreendido para criar um produto ou serviço único”. Assim um projeto visa a satisfazer um cliente ou usuário, possuindo exigências quanto a qualidade e um limite no orçamento e no prazo, além de enfrentar desafios na comunicação e no trabalho em equipe.

Os projetos em geral possuem muitas incertezas, incógnitas e suposições. Para os projetos de desenvolvimento de *software*, há um desafio ainda maior a ser enfrentado, a identificação e análise dos requisitos. Esse processo, para a maioria dos tipos de projetos, costuma ser de fácil especificação e poucos erros, o que não ocorre nos projetos de *software*. Além da incerteza quanto aos requisitos, as empresas muitas vezes enfrentam dificuldades para encontrar mão de obra especializada e também na adoção de boas práticas. Essas dificuldades são causadas pela velocidade de evolução da Tecnologia da Informação e pelo fato de o produto final do projeto, o *software*, ser intangível.

---

1 efranzen@univates.br

2 hbarden@universo.univates.br

Na área da tecnologia da informação é comum haver modificações dos requisitos nas fases de construção ou produção, que, em conjunto com outros desafios, caracteriza os projetos por atrasos, problemas relacionados ao orçamento e resultados aquém das expectativas dos clientes e usuários finais. Algumas características dos projetos de TI, como estado final do projeto não bem definido e má estimativa de horas, contribuem para esses fracassos.

Dessa forma, este estudo tem como objetivo elaborar uma proposta de aprimoramento do processo de desenvolvimento de *software* da Interact Solutions Ltda. efetuando uma melhoria nas suas estimativas. Para atingir esse objetivo, são analisados processos da área de desenvolvimento de *software* da Interact Solutions Ltda. e após propostas melhorias a esta, além de identificada uma forma de mensurar o trabalho dos envolvidos no processo de desenvolvimento de *software*.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Engenharia de *Software*

O *software* é o grande objetivo buscado pela engenharia, visando a obter um produto que seja confiável, economicamente viável e eficiente, ou seja, atenda as necessidades do usuário (Tonsig, 2008). Ainda esse autor afirma que atualmente os *softwares* desenvolvidos envolvem alto grau de complexidades e muitas variáveis, sendo fundamental o planejamento da aplicação, trazendo a utilização de um método muitas vantagens, como padronização, maior controle, maximização de resultados e uniformidade de linguagem.

Os métodos de engenharia de software fornecem a técnica de 'como fazer' para construir software. Eles abrangem um amplo conjunto de tarefas que incluem comunicação, análise de requisitos, modelagem de projeto, construção de programas, testes e manutenção. Os métodos de engenharia de software repousam num conjunto de princípios básicos que regem cada área da tecnologia e incluem atividade de modelagem e outras técnicas descritivas (PRESSMAN, 2010, p. 18).

Conforme Pressmann (2010), a engenharia de requisitos é uma tarefa da engenharia de *software*, sendo seu entendimento uma das tarefas mais difíceis que um engenheiro de *software* precisará enfrentar.

Por isso Pressmann (2010) afirma que muitas vezes os clientes não sabem exatamente o que desejam, e, se sabem, essas necessidades se modificam no decorrer do projeto. Essas modificações devem ser controladas por meio de mecanismos a serem estabelecidos, antes que elas controlem o projeto.

Segundo Sommerville (2007), quando os requisitos do sistema são redigidos em diferentes níveis de especificação, permitem melhor entendimento aos vários tipos de leitores. Os requisitos de usuário são redigidos com linguagem natural e diagramas, utilizando alto nível. Já os requisitos de sistema visam a detalhar o sistema e suas funções, definindo exatamente o que será implementado.

### 2.2 Métodos ágeis

Conforme Pressmann (2010), a agilidade é uma resposta efetiva a modificações no *software*, na equipe, nas tecnologias utilizadas ou em quaisquer outras mudanças. Ela torna a comunicação entre as pessoas envolvidas mais fácil e adota os clientes como parte da equipe, além de flexibilizar o plano de projeto.

Sommerville (2007) afirma ainda que os métodos ágeis propõem processos diferentes para o desenvolvimento, porém compartilham dos mesmos princípios, tendo muitos pontos em comum. Exemplos de princípios são: o envolvimento do cliente, entregas incrementais, desenvolvimento das

habilidades da equipe, desenvolvimento de maneiras próprias para trabalho, aceite de mudanças e simplicidade do *software*.

Segundo Pressmann (2010), o Scrum foi desenvolvido por Jeff Sutherland e sua equipe no início da década de 1990. Esse modelo ágil permite a construção de *softwares* mais flexíveis, enfatizando o uso de padrões de processo de *software*, sendo efetivos para prazos apertados.

### 2.3 Projetos

Conforme Keelling (2002, p. 3), a definição para projeto adotada pelo PMI resume as várias definições existentes, dizendo ser “Um esforço temporário empreendido para criar um produto ou serviço único”. Sendo assim, há um prazo limitado com data de conclusão estipulada e um resultado específico para ele, o que também é afirmado por Newton (2011).

Newton (2011) complementa observando que projeto é um modo de trabalho, de organizar pessoas e gerenciar atividades. Quando o seu resultado é alcançado, o projeto é finalizado, deixando de ser necessário. Há um ponto definido para o projeto começar e terminar, existindo recursos geralmente limitados para a sua execução, como de dinheiro, tempo e de pessoas. Cada projeto é específico, tem um conjunto de atividades único, mesmo que possua características comuns com projetos passados.

Os projetos contemporâneos apresentam-se em muitas formas e tamanhos. Alguns são de curta duração, empreendimentos baratos que duram apenas alguns dias e necessitam de recursos mínimos. Projetos de médio ou longo prazo, por outro lado, podem representar empreendimentos ambiciosos que se estendem por muitos anos e exigem grandes recursos financeiros e materiais, altos níveis de habilidade técnica e estruturas de administração complexas (KEELLING, 2002, p. 4).

Conforme Vieira (2003), pelo fato de projetos terem o objetivo de obter um serviço ou produto único, são envolvidos pelas incertezas, que são agravadas em projetos de tecnologia da informação. A administração dessas incertezas é um grande desafio no gerenciamento de projetos.

Vieira (2003, p. 15) ainda afirma que “o objetivo do gerenciamento de projetos é garantir o cumprimento do escopo, dos prazos, dos custos e da entrega dos produtos com a qualidade esperada pelo cliente”. Phillips (2003, p. 2) complementa dizendo que “[...] a gerência de projetos é a capacidade de administrar uma série de tarefas cronológicas que resultam em uma meta desejada”.

### 2.4 Gestão de projetos

Vieira (2003, p. 15) salienta que “o objetivo do gerenciamento de projetos é garantir o cumprimento do escopo, dos prazos, dos custos e da entrega dos produtos com a qualidade esperada pelo cliente”. Phillips (2003, p. 2) acrescenta que “[...] a gerência de projetos é a capacidade de administrar uma série de tarefas cronológicas que resultam em uma meta desejada”.

Conforme Keelling (2002, p. 9), “Conduzir um projeto exige capacitação em todas as disciplinas da administração, mas a natureza de curto prazo desta espécie de trabalho exige aplicação especial, disciplina e técnica”. Vieira (2003) complementa afirmando que o ponto principal do gerenciamento dos projetos é o controle, pois sem este não é possível atender aos requisitos do cliente.

Segundo Vazquez, Simões e Albert (2003), entende-se por requisito uma característica, condição ou capacidade que o usuário necessita em seu negócio, e que deve ser atendida.

Ainda Vazquez, Simões e Albert (2003) afirmam que existem muitos processos que buscam permitir a gerência de projetos, e são agrupados em grupos básicos, dentre os quais três se destacam:

- planejamento: definição de objetivos em busca da melhor forma de ação;
- execução: gerenciamento de pessoas e demais recursos para que seja efetuada a execução;

- controle: por meio do monitoramento e medição do processo, garantir que os objetivos serão alcançados e tomar as devidas ações corretivas.

## 2.5 Métricas de software

Segundo Kerzner (2006), apenas o fato de seguir uma metodologia para a gestão de projetos não garante que esta obtenha sucesso e excelência, pois fatores externos podem influenciar fortemente e levar ao fracasso. As metodologias precisam evoluir juntamente com as mudanças que ocorrem nas organizações, e estas exigem muita dedicação dos administradores visando ao desenvolvimento de uma sistemática de gestão de projetos que seja integrada com os demais sistemas da organização.

Segundo Vazquez, Simões e Albert (2003), a análise de pontos de função é um processo padronizado para a medição do desenvolvimento de um *software* sob a ótica do usuário. O levantamento de requisitos nesse processo é fundamental, pois é por meio dele que as necessidades dos usuários são mapeadas e após medidas ou contadas. Não existe nenhum método mágico. Se os dados não forem adequados, a estimativa também será inadequada.

A métrica ponto por função (*Function point – FP*), inicialmente proposta por Albrecht [ALB79], pode ser usada efetivamente como um meio para medir a funcionalidade entregue por um sistema. Usando dados históricos, o FP pode então ser usado para (1) estimar o custo ou esforço necessário para projetar, codificar e testar o software; (2) prever o número de erros que vão ser encontrados durante o teste; e (3) prever o número de componentes e/ou o número de linhas de código projetadas no sistema implementado (PRESSMANN, 2010, p. 357).

## 3 MÉTODO DE PESQUISA

Segundo Vergara (2004, p. 12), “Método é um caminho, uma forma, uma lógica de pensamento”. Assim, este capítulo apresenta os procedimentos metodológicos, buscando fornecer uma ordem lógica para o desenvolvimento da pesquisa.

A natureza deste estudo é a pesquisa aplicada, a fim de adquirir conhecimentos e dar soluções para os problemas específicos, ou seja, analisar o processo atual de desenvolvimento de *software* e, por meio de uma metodologia, buscar aprimoramentos para este, visando a um maior índice de sucesso nos projetos.

Esta pesquisa caracteriza-se como qualitativa visto que o conteúdo das entrevistas individuais e das observações realizadas foi avaliado pela qualidade e interpretação das respostas.

Para a realização deste estudo, a pesquisa é de caráter exploratório, pois é a melhor e a mais adequada maneira de se adquirirem informações precisas, aumentar o conhecimento e atender aos objetivos propostos neste projeto.

A pesquisa neste estudo foi efetuada por meio de levantamento bibliográfico, sendo analisados dados a partir de material já publicado, como bibliografias e monografias, visando a um melhor entendimento do estudo.

A presente pesquisa também pode ser classificada como estudo de caso. Para Beuren (2003, p. 84), o estudo de caso “caracteriza-se principalmente pelo estudo concentrado de um único caso. Esse estudo é preferido pelos pesquisadores que desejam aprofundar seus conhecimentos a respeito de determinado caso específico”.

Os dados coletados são analisados e interpretados de forma qualitativa. Pesquisa qualitativa é basicamente aquela que busca entender um fenômeno específico em profundidade. Ao invés de estatísticas, regras e outras generalizações, a qualitativa trabalha com descrições, comparações e interpretações para posteriormente identificar os possíveis problemas que deverão ser modificados

na empresa, com a finalidade de propor um aprimoramento do processo de desenvolvimento de *software*.

#### 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

O presente trabalho foi efetuado na empresa Interact Solutions Ltda., que atua no ramo de Tecnologia da Informação desde 1999. Atualmente ocupa reconhecida posição de liderança no Brasil em pesquisa e desenvolvimento de *softwares* para a gestão corporativa.

A empresa utiliza um processo de desenvolvimento de software que visa a obter um produto de qualidade, atendendo a padrões previamente estabelecidos. O processo foi concebido para atender a realidade da Interact, sendo consideradas metodologias reconhecidas e amplamente utilizadas, como RUP, OpenUP, modelo CMMI e PMI.

O processo de desenvolvimento de *software* da Interact (PDS Interact) é dividido em fases, que indicam a ênfase dada ao projeto no decorrer do seu ciclo de vida:

- definição: ênfase no planejamento do projeto;
- análise: ênfase na análise do projeto, sendo envolvidas questões de arquitetura, requisitos, protótipos e funcionalidades, cronogramas e riscos do projeto;
- construção: ênfase na construção do projeto, sendo realizados a implementação do código, os testes específicos para cada requisito implementado e iniciada a documentação do projeto;
- transição: ênfase na transição do projeto do ambiente de desenvolvimento para produção, sendo efetuados testes integrados, documentação e a garantia da qualidade;
- produção: ênfase na passagem do *software* para o ambiente de produção, sendo este disponibilizado para o cliente, e também efetuado o término do projeto.

Antes da formalização do projeto, são realizadas as atividades de identificação dos requisitos e formalização do projeto por meio de documentos. Após inicia-se o ciclo do Processo de Desenvolvimento de *Software*.

A fase de definição tem como objetivo definir o escopo do projeto a partir da lista de requisitos, aprovada pelo gerente de produtos, bem como criar o plano de projeto e obter a aprovação deste pelos envolvidos.

A fase de análise tem como objetivos a elaboração e aprovação das especificações funcionais técnicas dos requisitos (EFTR), conforme o escopo do plano de projeto.

A fase de construção objetiva implementar as melhorias, criar o plano de testes e iniciar a documentação de acordo com a especificação contida nos artefatos, visando sempre a cumprir as exigências e os prazos estipulados no Plano de Projeto.

A fase de transição visa a garantir a conformidade da implementação em relação ao que foi especificado.

A fase de produção tem como objetivo disponibilizar o produto ao cliente e finalizar o projeto.

#### 5 SUGESTÕES DE MELHORIA

Nesta sessão são apresentados os processos das fases com suas respectivas melhorias, objetivando introduzir a análise de pontos de função aos processos. Também é apresentada uma proposta para o processo da metodologia Scrum, que já é utilizada pela Interact Solutions, porém até o momento não possui um processo formalizado.

### 5.1 Fase 1 - Definição

A fase de definição tem como objetivo definir o escopo do projeto a partir da lista de requisitos, bem como criar o plano de projeto.

Segundo entrevistas e documentos coletados na empresa, um projeto é iniciado oficialmente com o recebimento de documento em que constam todas as informações necessárias para iniciar o projeto, como, por exemplo: lista dos requisitos, definição do gerente do projeto e do analista de sistemas.

O gerente de projeto fica encarregado de definir a especificação dos requisitos de *software* (ERS), criar plano de projeto, definir o cronograma, estimar o esforço, homologar a ERS.

Para a definição do documento com a lista de requisitos, se necessário, o analista de sistemas pode vir a obter maior entendimento dos requisitos juntamente com o seu fornecedor, sendo este entendimento devidamente registrado, seja por ata ou *e-mails* de aceitação pelo fornecedor.

Com o objetivo de detalhar a especificação técnica dos requisitos, um novo processo que anteriormente era efetuado, não obrigatoriamente, na fase de análise foi adicionado à fase de definição: especificação da arquitetura. Essa especificação abrangerá, além da estrutura do banco de dados, as principais classes e pacotes a serem criados. Deve ser efetuada antes da estimativa de esforço, pois os levantamentos efetuados nesse processo serão utilizados para essa estimativa.

A estimativa de esforço dos requisitos a serem implementados deve ser efetuada na Planilha de Estimativas utilizada pela empresa. Nesta é apenas necessário informar os requisitos do projeto, definir a complexidade e os atores deles para obter a visão geral do esforço bruto. A metodologia utilizada para a estimativa de esforço anteriormente era *Use Case Points* (Pontos por caso de uso), agora passando a utilizar análise de pontos de função e possuindo dois passos a serem seguidos:

- efetuar a contagem dos pontos, com base nos requisitos, arquitetura, casos de uso, desenhos de telas e demais documentos que o projeto possuir. Quanto mais clara estiver a necessidade do usuário, a visão sobre o sistema, mais precisa será a contagem, que varia conforme o tipo do projeto (vide exemplo na sessão “estudo de caso”);
- calcular prazo e custos utilizando os pontos de função. Para que esse procedimento seja efetuado, é necessário que a empresa já possua uma base para o cálculo. Apenas os números de pontos de função não são o suficiente para determinar prazos e custos. A eficácia desse método depende de dados históricos. Para iniciar a formação da base, a empresa pode efetuar o cálculo dos pontos de função do projeto, porém utilizar seu método tradicional de estimativa de horas e custo enquanto uma base é formada.

Prosseguindo com o processo, é efetuada uma reunião visando a homologar a lista de requisitos e as estimativas de esforço. Ainda nessa reunião define-se a necessidade de criação da arquitetura do projeto ou não.

Com os documentos já homologados, estes são submetidos ao cliente para avaliação, sendo solicitada uma aprovação formal deles. Caso o cliente solicite modificações, retorna-se à etapa de definição dos requisitos. Neste ponto ele também pode vir a solicitar o cancelamento do projeto. Acontecendo isso, a equipe alocada até o momento é comunicada.

Após a aprovação do cliente, é necessário obter a aprovação da equipe sobre o cronograma do projeto, validando o envolvimento de cada membro nas atividades, o plano de riscos e o plano de comunicação.

## 5.2 Fase 2 – Análise

A fase de análise tem como objetivo a elaboração e a aprovação das EFTRs, conforme o escopo do Plano de Projeto.

A primeira tarefa a ser realizada nesta fase seria especificar a arquitetura, caso esta atividade tenha sido prevista na fase anterior. No novo modelo, a tarefa se tornou requerida na fase de definição, não sendo mais executada na fase de análise.

Com a especificação dos requisitos, também desenvolve-se a especificação funcional, utilizando um template fornecido pela empresa. Se houver alguma dúvida sobre um requisito, pode-se entrar novamente em contato com o fornecedor dos requisitos para saná-las, sendo tudo registrado formalmente.

Após a especificação funcional dos requisitos ser concluída, é necessário obter a aprovação do documento na equipe e posteriormente com o fornecedor. Se houver alterações e estas afetarem o escopo do projeto, é efetuada a gerência de mudanças.

Na etapa anterior, mudanças podem ter sido efetuadas, necessitando de alterações no banco de dados, propriedades, estrutura das classes, as quais devem ser devidamente documentadas na EFTR e nos demais documentos necessários. Concluída essa atividade, é necessário aprovar novamente os documentos na empresa.

Após o término do detalhamento dos requisitos, é refeita a estimativa de esforço, custo e prazo do projeto, buscando também a aprovação das modificações na empresa. Caso haja alterações nas estimativas, são revisados os valores em relação aos definidos no início do projeto, verificando se é possível encaixar as mudanças. Não sendo possível, é efetuada uma reunião interna na empresa para se chegar a um consenso.

## 5.3 Fase 3 – Construção

Na fase de construção são efetuadas tarefas de testes, documentação e implementação de requisitos. A Interact Solutions adotou a metodologia ágil Scrum em seu processo, a qual ainda não foi formalmente descrita. Nesta sessão utiliza-se uma proposta de processo baseada nessa metodologia.

A primeira tarefa a ser efetuada nesta fase é a definição do time de trabalho para o Sprint, que pode ser composto por desenvolvedores, testadores, documentadores e analistas de sistemas. Idealmente os times do Scrum devem ser formados por colaboradores de uma mesma função, ou seja, somente de documentadores, somente de testadores ou desenvolvedores, para facilitar na mensuração dos trabalhos.

Após essa definição, é efetuada a reunião que define quais são as características ou requisitos pendentes que devem ser implementados com prioridade por fornecerem maior valor de negócio ao cliente. São participantes dessa reunião o gerente de projetos e o cliente.

Com a lista de atividades definidas em ordem decrescente de prioridade, nova reunião é realizada, agora entre o time do Scrum anteriormente definido. Essa reunião tem como objetivo expandir os itens de pendência, os dimensionando em horas de acordo com a experiência dos integrantes. Um item da lista não necessariamente será representado por uma única atividade, pode-se dividi-lo em várias atividades menores. A dimensão final de uma determinada atividade deve ser de comum acordo com todos os presentes na reunião.

Algumas atividades que o processo anterior impunha à execução, como, por exemplo, realização dos testes unitários por parte dos desenvolvedores, de teste específico por parte dos

testadores e análise da EFTR por parte dos documentadores, agora devem se tornar tarefas a serem adicionadas no sprint para a execução quando necessário.

O time deve dimensionar atividades para serem executadas nos próximos onze dias úteis, efetuando uma média de seis horas diárias de trabalho por integrante. No decorrer da reunião, deve adicioná-las estas a uma planilha padrão utilizada pela empresa. Nessa reunião também é definido o Scrum Master da equipe, que irá representar a equipe quando necessário e a liderar.

Após as atividades serem definidas, novas tarefas não devem ser inclusas. É necessário adicioná-las na lista de pendências para o próximo sprint. Tarefas que necessitam de urgência devem ser requisitadas ao Scrum Master, que lhes dará o devido encaminhamento.

A próxima atividade refere-se aos pontos de função definidos na fase de análise do projeto. Caso não haja pontos de função para o projeto em questão, ou o corrente sprint é somente destinado a testes e ou a documentação e estas atividades não possuam pontos definidos, pula-se esta etapa do processo. Com o término da definição das atividades, deve-se analisar a contagem de pontos de função do projeto juntamente com a lista de atividades definida pelo time, para efetuar o cálculo de pontos de função que o sprint irá contemplar e também adicionar os pontos que cada item da planilha das atividades representa. Caso um item da lista de pendências seja representado por várias atividades, deve-se particionar os pontos de função de acordo com o peso de cada atividade, tendo como base suas horas estimadas.

O Scrum Master definido na reunião de planejamento é o responsável pelo quadro de tarefas, que é utilizado na empresa para gerenciar o andamento delas. Nesse quadro existem três colunas, que determinam o estado em que a tarefa pode estar: "A fazer" (TODO), "Em execução" (DOING) e "Feito" (Done). As tarefas contidas na planilha são escritas em post-its coloridos e inicialmente são inseridas na primeira coluna do quadro. À medida que elas vão sendo desenvolvidas, avançam até chegar à última coluna. Diariamente o Scrum Master deve retirar as tarefas que estão na última coluna e atualizar com os dados destas a planilha de atividades.

Para iniciar a execução das tarefas, verifica-se se já há um repositório de trabalho criado. Caso não haja, é necessário solicitar a criação, sendo esses sempre efetuados via *e-mail*.

Com o repositório já definido é necessário preparar o ambiente para a execução das tarefas. Se esse ainda não está preparado, efetuam-se as configurações necessárias e os demais ajustes relacionados, como base de dados, por exemplo.

Após todos esses procedimentos, inicia-se a execução das tarefas, que devem ser efetuadas na ordem em que estão definidas na planilha do sprint e dispostas no quadro de tarefas. Ao iniciar uma tarefa, o colaborador deve mover o respectivo post-it para a próxima coluna, "Em execução", preenchendo-o com a abreviatura desta, data e hora de início da atividade. Ao finalizar a tarefa deve-se mover o post-it para a última coluna e preenchê-lo com a data e a hora de término da atividade, o tempo que demorou para finalizar a tarefa.

Caso o executor se depare com a necessidade de realizar uma tarefa que não foi planejada na reunião, deve adicionar um post-it ao quadro de tarefas, utilizando uma cor diferenciada da que vem sendo usada por padrão. Esse post-it, ao chegar à última coluna, será adicionado à planilha do sprint pelo Scrum Master como uma tarefa extra.

Diariamente uma reunião deve ocorrer em um horário e local a ser determinado pelo próprio time. Essa reunião deve ser efetuada em pé e durar aproximadamente quinze minutos. Nela cada integrante deve comentar sobre o que fez desde a última reunião diária, o que vai fazer hoje e se está com alguma dificuldade. Membros de outras equipes podem assistir a reunião, porém não podem se pronunciar.



Chegando ao décimo primeiro dia de sprint, organiza-se a reunião de finalização dele, na qual será apresentado ao cliente o resultado obtido. Caso não for possível o cliente participar, a gerência efetua o papel do cliente.

Com a finalização do sprint, atividades não devem mais ser executadas, somente após serem novamente definidas tarefas. Muitas vezes nem todas as atividades são executadas, por motivos de má estimativa de horas ou então atividades extras demasiadas que atrapalham o andamento do sprint. Outras vezes são estimadas menos atividades do que realmente se poderia executar, o que acarreta em adição de atividades extras. Independente do ocorrido, após a finalização do sprint, o gerente de projeto deve efetuar a verificação de pontos de função efetivamente implementados, o que acarreta em somar na planilha de atividades os pontos das atividades executadas. Este resultado deve ser mantido na planilha. Esta atividade também só deverá ser executada se houver pontos de função já registrados na planilha de atividades.

Caso o sprint finalizado possuir implementações que necessitam de testes, como, por exemplo, equipes que desenvolveram rotinas, telas para o sistema, efetuaram correções ou qualquer outro caso, o analista de sistemas deve criar um plano de testes que instrua a equipe que irá efetivamente testar os requisitos implementados, para que tomem conhecimento dos pontos e como devem se comportar.

Finalizando a fase de construção, há nova reunião entre o time que executou as tarefas do sprint – reunião de Análise Crítica. Nela os integrantes devem informar, na planilha de atividades, os pontos que avaliaram como “bons” durante o sprint e os que “poderiam melhorar”, sempre apresentando no mínimo dois itens por categoria. Após devem ler os apontamentos que definiram como “poderiam melhorar” e entre eles escolher dois nos quais efetivamente estão dispostos a trabalhar, adicionando-os em nova categoria, “melhorar”.

### 5.3.1 Monitoramento da produtividade

Com as informações de horas realizadas e pontos de função correspondentes por tarefa efetuado no sprint, pode-se mensurar a produtividade média em determinado período.

Após a finalização do sprint, indicadores podem ser alimentados, visando a construir uma base sólida de mensuração de horas primeiramente por equipe, podendo vir a ser por usuário, se for efetuada a separação de atividades executadas. Além de serem adicionadas as horas trabalhadas, também acrescentar informações sobre os pontos de função efetuados.

Exemplo: em um sprint com um time de três pessoas, foram realizadas 30 atividades, utilizando 180 horas para a execução, totalizando 111 pontos de função. Nesse período pode-se chegar a algumas conclusões:

- produtividade média – dividindo os pontos de função efetuados pelo número de integrantes, a produtividade média alcançada foi de 37 pontos de função, ou seja,  $111 / 3 = 37$ ;
- tempo por ponto de função – dividindo o número de horas para execução pelos pontos de função efetuados, resulta em 1,62 horas (aproximadamente 1 hora e 37 minutos) por ponto de função efetuado, ou seja,  $180 / 111 = 1,62$ .

Com esses dados a empresa inicia sua base histórica para gradativamente ir melhorando as estimativas de projetos.

A comparação de médias de produtividade e tempo utilizado para implementação de ponto de função entre os times pode fornecer à empresa diagnósticos de andamento dos projetos,

identificando problemas e possibilitando à empresa corrigi-los. Esse diagnóstico só será possível se um padrão de contagem de pontos de função for seguido.

Outra forma de controle que a técnica permite é a variação da produtividade nos períodos do ano, como, por exemplo, uma rentabilidade constante de janeiro a outubro e uma queda em novembro e dezembro.

#### **5.4 Fase 4 – Transição**

A fase de transição tem como objetivo garantir a conformidade da implementação em relação ao que foi especificado. Para tanto, eram efetuados a documentação e testes da implementação. Porém, com o novo processo de construção, a fase de transição não contará mais com tarefas de documentação e testes, uma vez que estas devem ser planejadas e executadas nos sprints da construção.

Inicia-se esta fase efetuando um teste de integridade do pacote gerado para entrega ao cliente, que deve ser o produto final gerado em um dos sprints. Se este não estiver correto, a equipe que o gerou é comunicada e uma atividade extra de urgência é adicionada ao sprint, para que o problema seja resolvido.

Caso uma homologação do produto no cliente tenha sido planejada, executa-se o teste e o gerente de projetos solicita ao analista de sistemas a execução desta atividade, que consiste em efetuar teste integrado juntamente com o cliente. Se o cliente não homologar o produto e sugerir alterações, deverá ser executado o processo de gerência de mudanças, complementando a especificação da arquitetura e revendo as estimativas, além de adicionar os itens à lista de pendências do produto, com sua devida prioridade.

#### **5.5 Fase 5 - Produção**

A fase de produção tem como objetivo disponibilizar o produto ao cliente e finalizar o projeto. Para isso, é efetuada a distribuição do pacote de entrega ao cliente e também aos colaboradores. O suporte da empresa e interessados no projeto também recebem um *e-mail* informativo sobre a entrega do produto, juntamente com propriedades ou configurações necessárias para a sua instalação.

Após a entrega do produto, é efetuada uma reunião com a equipe para realizar uma análise final, avaliando o andamento do projeto, os objetivos alcançados, dificuldades enfrentadas e as sugestões de melhorias. Outra reunião é efetuada na sequência, para marcar o encerramento do projeto e levantar as Lições Aprendidas.

#### **5.6 Comparação de processos**

Com as sugestões de melhoria já efetuadas, nesta seção é apresentado um quadro comparativo entre as macro-tarefas realizadas no PDS Interact e no processo proposto neste projeto. Nele é possível observar de forma resumida as adequações propostas no processo.

Quadro 1 – Comparação dos processos

	<b>Processo Interact Solutions</b>	<b>Processo proposto</b>
Fase 1 - Definição	Plano do projeto, lista de requisitos, estimativa de esforço com a utilização de UseCasePoints, formalização do projeto.	Plano do projeto, lista de requisitos, especificação da arquitetura, estimativa de esforço com a utilização de análise de pontos de função efetuando a contagem dos pontos de função, formalização do projeto.
Fase 2 - Análise	Especificação da arquitetura não obrigatória, especificação funcional técnica dos requisitos (EFTR), aprovação da especificação e, se necessário, revisão das estimativas na gerência de mudanças utilizando de UseCasePoints.	Especificação funcional técnica dos requisitos (EFTR), aprovação da especificação e, se necessário, revisão das estimativas na gerência de mudanças, utilizando análise de pontos de função e efetuando novamente o cálculo dos pontos de função.
Fase 3 - Construção	Elaboração do plano de testes, elaboração da documentação, implementação dos requisitos, realização dos testes.	Formação de times de trabalho, realização de tarefas de testes, documentação e implementação dos requisitos utilizando a metodologia ágil scrum e, após o sprint, contagem de pontos de função realizados.
Fase 4 - Transição	Finalização da documentação dos requisitos, realização de testes integrados, homologação do produto pelo cliente, se necessário, realização da gerência de mudanças, revisando a especificação da arquitetura, caso exista, e a estimativa de horas utilizando UseCasePoints.	Finalização da documentação dos requisitos, realização de testes integrados, homologação do produto pelo cliente, se necessário, realização da gerência de mudanças, revisando a especificação da arquitetura e a estimativa de horas, utilizando análise de pontos de função e efetuando novamente o cálculo dos pontos de função.
Fase 5 - Produção	Entrega do produto ao cliente e encerramento do projeto.	Entrega do produto ao cliente e encerramento do projeto.
Monitoramento da produtividade	Utilização de horas como métrica.	Utilização de pontos de função como métrica

Fonte: Elaborado pelos autores.

## 6 CONSIDERAÇÕES FINAIS

Com o desenvolvimento deste trabalho foi possível atingir os objetivos propostos por meio da análise dos processos atuais da Interact Solutions Ltda. e das sugestões de melhoria.

O desenvolvimento de uma proposta de processo aprimorado buscou auxiliar a empresa em problemas comuns enfrentados pela gerência de projetos. Segundo Vazquez, Simões e Albert (2003), a indústria de *software* desde, seu surgimento, possui deficiências para gerar estimativas confiáveis. Apesar de os métodos de estimativa terem avançado muito, a falta de disseminação dos métodos entre os profissionais não permitiu que o problema fosse solucionado. Com dados inadequados a estimativa fornecida também será inadequada.

Dessa forma, para que a metodologia proposta seja efetiva, é necessário que todos os colaboradores, principalmente a gerência, auxiliem na sua disseminação, efetuando melhorias no

processo quando necessário. A sua correta utilização também é um ponto crítico para o processo, visando à geração de dados reais e íntegros.

A estimativa dos projetos irá gradativamente se tornando mais precisa e simples à medida que o processo for sendo executado e os colaboradores obtiverem experiência na contagem de pontos de função. Para os primeiros projetos que utilizarem esse processo, é provável que haja maior dificuldade tendo em vista que é um fato novo para todos.

A solução proposta possivelmente necessitará de ajustes no processo, uma vez que até o momento não foi implementada. Somente após a sua utilização poderão ser avaliados precisamente sua eficiência e os benefícios por ela gerados.

O nível de detalhamento exigido pela análise de pontos de função na contagem dos pontos pode vir a beneficiar também o escopo dos projetos, em que muitas vezes alguns pontos não são tratados, causando problemas posteriormente.

Outro benefício que a metodologia irá proporcionar é na mensuração do trabalho dos envolvidos, pois com a utilização de pontos de função a estimativa passa a ser efetuada por esforço, e não mais por horas, possivelmente gerando mais precisão e padronização, sem variações. Com isso a empresa irá poder trabalhar em estatísticas de horas x esforço, gerando uma base sólida de produtividade.

Já possuindo dados históricos de projetos, experiência na contagem de pontos de função e registros de produtividade dos colaboradores, a empresa conseguirá atingir um nível satisfatório de estimativas.

Pelo conhecimento adquirido na organização, uma dificuldade que pode ser encontrada para a implantação dessa metodologia é o nível necessário de detalhamento dos requisitos para a contagem dos pontos de função e, por consequência, a disponibilidade de tempo dos colaboradores para essa tarefa. Um fato favorável que pode ser apontado para implementação da metodologia é o interesse que alguns membros da empresa demonstraram para o uso de análise por pontos de função, como um experimento visando à melhoria das estimativas dos projetos.

## REFERÊNCIAS

- BEUREN, I. M. **Como elaborar Trabalho Monográfico em Contabilidade: teoria e prática**. São Paulo: Atlas, 2003.
- KEELLING, Ralph. **Gestão de Projetos: uma abordagem global**. São Paulo: Saraiva, 2002.
- KERZNER, Harold. **Gestão de projetos: as melhores práticas**. 2. ed. Porto Alegre: Bookman, 2006.
- NEWTON, Richard. **O Gestor de Projetos**. 2. ed. São Paulo: Pearson Prentice Hall, 2011.
- PHILLIPS, Joseph. **Gerência de Projetos de Tecnologia da Informação: no caminho certo, do início ao fim**. 2. ed. Rio de Janeiro: Campus, 2003.
- PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. Porto Alegre: AMGH, 2010.
- SOMMERVILLE, Ian. **Engenharia de software**. 8. ed. São Paulo: Pearson Addison, 2007.
- TONSIG, Sérgio Luiz. **Engenharia de Software: análise e projeto de sistemas**. 2. ed. Rio de Janeiro: Ciência Moderna Ltda, 2008.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. **Análise de Pontos de Função**: medição, estimativas e gerenciamento de projetos de software. São Paulo: Érica, 2003.

VERGARA, Sylvia Constant. **Projetos e Relatórios de Pesquisa em Administração**. 5. ed. São Paulo: Atlas, 2004.

VIEIRA, Marconi Fábio. **Gerenciamento de Projetos de Tecnologia da Informação**. 3. ed. Rio de Janeiro: Elsevier, 2003.