IV SINGEP
Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade
International Symposium on Project Management, Innovation and Sustainability
ISSN: 2317 - 8302

IMPLEMENTAÇÃO DE UM MODELO DE GERENCIAMENTO DE PROJETOS DE SOFTWARE COMERCIALIZADOS DE FORMA STANDARD E CUSTOMIZADA EM UMA INDÚSTRIA METALÚRGICA

LUIZ FERNANDO LIMA
UNINOVE – Universidade Nove de Julho
luizlima.engenheiro@gmail.com

FILIPE QUEVEDO PIRES DE OLIVEIRA E SILVA
UNINOVE – Universidade Nove de Julho
admquevedo@hotmail.com

IV SINGEP
Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade
International Symposium on Project Management, Innovation and Sustainability
ISSN: 2317 - 8302

IMPLEMENTAÇÃO DE UM MODELO DE GERENCIAMENTO DE PROJETOS DE SOFTWARE COMERCIALIZADOS DE FORMA STANDARD E CUSTOMIZADA EM UMA INDÚSTRIA METALÚRGICA

Resumo

Os projetos de software são cada vez mais uma realidade nas organizações, independente do seguimento. Para atender aos requisitos dos clientes as empresas vem adotando metodologias de gerenciamento de projetos muitas vezes prontas e algumas vezes adaptadas a realidade de cada empresa. Este relato tem o objetivo de identificar a existência de contribuições na utilização de um modelo de gestão de projetos para desenvolvimento de softwares comercializado de forma *standard* e customizada no atendimento ao requisito de prazo de entrega. Utilizando estudo de caso como estratégia, esta pesquisa concluiu haver contribuições, no entanto a existência de um modelo não é um fator determinante no atendimento a este requisito.

Palavras-chave: Gestão de Projetos; Projetos de Software; Metodologias de Gestão de Projetos.

Abstract

The software projects are increasingly a reality in organizations regardless of the kind. To meet customer requirements the company has adopted project management methodologies often ready and sometimes adapted to the reality of each company. This report aims to identify the existence of contributions to the use of a model of project management for software development commercialized standard and customized way in meeting the requirement of delivery. Using case study as strategy, this research concluded that there are contributions, however the existence of a model is not a determining factor in meeting this requirement.

Keywords: *Project Management; Software Projects; Project Management Methodologies.*

ISSN: 2317 - 8302

1 Introdução

A demanda por softwares que automatizem os processos industriais minimizando intervenções humanas é crescente na indústria. Com a disseminação do conhecimento para desenvolvimento de software e as perspectivas de aumento de receitas, um número cada vez maior de empresas vem ingressando no mercado de desenvolvimento e fornecimento de softwares. Segundo Tonini, Carvalho e Spinola (2008), os consumidores do produto software querem ver suas necessidades atendidas com qualidade, dentro do prazo acordado e sem custos adicionais.

Uma das características dos projetos de software é o dinamismo. No ambiente de desenvolvimento de software os requisitos sofrem alterações frequentes (Rising & Janoff, 2000). Entende-se como requisitos do cliente os aspectos técnicos, de qualidade, custo e prazo, entre outros. Para atender as necessidades dos clientes, as organizações começaram a adotar as práticas de gerenciamento de projetos. Uma das justificativas para o crescimento do uso da gestão de projetos pode ser encontrada em uma de suas definições, onde gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas para atender aos requisitos do projeto (PMI, 2013).

Com o passar dos anos diversos modelos para gerenciamento de projetos de desenvolvimento de produtos foram criados. As empresas têm adotado estes modelos dentre os quais se destaca pela sua difusão o PMBOK® (Carvalho & Rabechini Jr., 2011). Um dos mais tradicionais, é o guia de melhores práticas de gestão de projetos do *Project Management Institute*. Outro modelo amplamente utilizado é o *Scrum*, que se enquadra nos chamados métodos ágeis. Estes métodos são considerados mais flexíveis e indicados para projetos onde ocorrem constantes mudanças de requisitos (Carvalho & Mello, 2012), assim como nos projetos de software. Existem ainda empresas que desenvolvem a sua própria metodologia de gestão de projetos, adequadas as características da organização.

Este relato tratará o caso de uma indústria metalúrgica, que vislumbrando a oportunidade de comercializar softwares que automatizassem a operação do seu produto, decidiu entrar no mercado de desenvolvimento e comercialização de softwares. O produto de software foco deste relato é comercializado por esta empresa como um produto *standard* e também de forma customizada. O produto *standard* é aquele onde não há necessidade de se fazer adequações para atender as necessidades do cliente, já o produto customizado sofre alterações de acordo com as necessidades do cliente. Devido a esta característica e com o aumento do número de clientes interessados no produto esta empresa começou a enfrentar problemas no cumprimento do prazo de entrega.

A busca por um modelo de gestão de projetos que atendesse as necessidades da organização, levou ao desenvolvimento do seu próprio modelo de gestão de projetos. Este modelo teve base em modelos existentes e tinha como foco principal minimizar os problemas com atendimento a prazos no desenvolvimento. Diante deste cenário, surge a questão de pesquisa central do relato: Como um modelo de gerenciamento de projetos pode contribuir na gestão de projetos de softwares desenvolvidos em uma indústria metalúrgica e comercializados de forma *standard* e customizada?

Com base no exposto, o objetivo central deste estudo é identificar como a implementação de um modelo de gerenciamento de projetos de software contribui no atendimento ao requisito de prazo de entrega em projetos de software comercializados como descrito. A busca pela resposta da questão de pesquisa se deu através de entrevistas e análise de documentos e registros. Na sequência encontra-se uma revisão da literatura que dará suporte ao modelo proposto e a análise dos resultados encontrados.

ISSN: 2317 - 8302

2 Referencial Teórico

2.1 Gerenciamento de Projetos de Software

O gerenciamento de um projeto de software é o processo de tomar decisões que envolvem o uso de recursos materiais e humanos, para coordenar, controlar e realizar atividades temporárias, com o objetivo de fornecer um resultado (Casey, 2010). Ainda segundo Casey (2010), isso precisa ser realizado dentro dos limites de recursos disponíveis, tanto financeiros como tecnológicos, utilizando pessoas, a capacidade técnica, recursos e tempo disponível levando em conta as características da organização. Em complemento pode-se dizer que a gestão de projetos de software é composta por um conjunto de processos que contêm os procedimentos que podem ser utilizados pelos gestores em qualquer tipo de projeto de desenvolvimento de software (Amescua *et al.*, 2004). Um gerenciamento de projeto bem conduzido não garante o sucesso do projeto, no entanto projetos onde o gerenciamento é mal executado normalmente resultam em não atendimento aos requisitos do projeto (Sommerville, 2007).

Gerenciar projetos de software não é uma tarefa fácil, a interação entre tecnologia, aspectos econômicos e humanos no contexto de projetos de software elevam a complexidade do gerenciamento deste tipo de projeto (Boehm, 1989). Com o advento da Internet, a melhoria dos hardwares, softwares, redes e grupos de trabalho interdisciplinares globais, mudaram o ambiente de trabalho e o gerenciamento de projetos de software (Nienaber & Barnard, 2007). A necessidade de satisfazer usuários, clientes, desenvolvedores, patrocinadores e gerenciadores contribuem para dificultar o gerenciamento de projetos de software. Estes conflitos influenciam diretamente no atendimento aos requisitos do projeto (Boehm, 1989).

Nos últimos anos uma serie de modelos de gestão de projetos e serviços de software foram desenvolvidos (Andrade & Tait, 2012). Dentre as várias metodologias desenvolvidas, existem as convencionais, que são orientadas a documentação, e as metodologias ágeis, que trabalham com o mínimo de documentação (Soares, 2003). Para Williams (2005) as metodologias convencionais de gerenciamento de projetos podem ser inapropriadas e potencialmente não vantajosas para projetos que possuem estrutura complexa, incertezas e restrições pesadas de tempo de projeto, onde estão enquadrados a maior parte dos projetos de software, quando comparadas as metodologias ágeis.

Considerado como um dos modelos tradicionais de gestão de projeto, o PMBOK® é baseado na realização das atividades em cascata e resulta em grande número de documentos (Silva, Souza, & Camargo, 2013). Ele surgiu na década de 1990, quando diversas associações lançaram as primeiras edições de seus guias de conhecimento em gerenciamento de projetos, entre estes guias está o PMBOK® (Carvalho & Rabechini Jr., 2011). O PMBOK® fornece diretrizes para o gerenciamento de projetos e define os conceitos de gestão relacionados com projetos. Ele também descreve o gerenciamento e seus processos relacionados, bem como o ciclo de vida do projeto. O PMBOK® divide a gestão de projetos em dez áreas de conhecimento, são elas: integração, escopo, tempo, custo, qualidade, recursos humanos, comunicação, riscos, aquisições e *stakeholders*. Cada uma destas áreas de conhecimentos possui seus documentos de entrada e saída (PMI, 2013).

De acordo com Fernandez, Worth e Fernandez (2009) a teoria subjacente ao PMBOK® é baseada na visão de transformação da produção e em algumas teorias da administração, como a gestão e planejamento. Para Koskela e Howell (2002) a teoria presente no PMBOK® está obsoleta e as teorias ágeis, como o *Scrum*, possuem uma abordagem mais adequada a projetos inovadores e complexos como a maioria dos projetos de software, permitindo uma gestão com maior fluência e geração de valor.

2.2 Método Ágil *Scrum*

A insatisfação com os métodos de desenvolvimento de software na década de 1990 levou a alguns desenvolvedores a propor uma alternativa aos métodos tradicionais, os chamados métodos ágeis (Sommerville, 2007). Existem diversos métodos ágeis, como por exemplo a Programação Extrema, o *Feature Driven Development*, *Dynamic Systems Development Method*, *Adaptive Software Development*, *Crystal*, *Pragmatic Programming*, *Scrum* e *Test Driven Development* (Carvalho & Mello, 2012). Para Sommerville (2007) existem algums princípios que são ligados aos métodos ágeis, como a participação ativa do cliente no processo de desenvolvimento, a priorização de mudanças deve ser realizada de forma racional, os sistemas devem ser desenvolvidos mantando a simplicidade necessária para atender ao requisito especificado, sofisticações desnecessárias podem comprometer os prazos e custos. Dentre os métodos ágeis que se destacam encontra-se o *Scrum* (Silva *et al*, 2013). O *Scrum* é um processo incremental para desenvolvimento de softwares em ambientes complexos. Ele fornece conhecimentos empíricos que permite o desenvolvimento ocorrer o mais próximo do caos que a organização pode tolerar (Rising & Janoff, 2000).

De acordo com Carvalho e Mello (2012), o *Scrum* estabelece um conjunto de práticas e artefatos gerenciais que devem ser seguidas para o sucesso de um projeto, sendo que o ponto inicial do *Scrum* é *Backlog* do Produto, nesta prática através de reuniões com os principais *stakeholders* é gerada uma lista de funcionalidades a serem desenvolvidas ordenada por prioridade de desenvolvimento. Outra prática bem característica do *Scrum* é o *Daily Scrum* ou *Stand Up Meeting*, a reunião diária entre os membros da equipe onde são discutidas as tarefas do dia anterior e o que será executado no decorrer do dia, para Rising e Janoff, neste encontro deve ser respondido: O que foi feito ontem? O que será feito Hoje? Há algum obstáculo para à realização das atividades?. Considerada a principal prática do *Scrum* por Carvalho e Mello (2012), o *Sprint* é o período de tempo para implementação de funcionalidades que dura normalmente de uma a quatro semanas. Ligado ao *Sprint* temos o *Backlog* do *Sprint*, lista de atividades a serem desenvolvidas durante o *Sprint*.

Por fim, Carvalho e Mello (2012) apresentam o artefato do *Burndown*, representação gráfica do trabalho realizado versus o trabalho restante, ferramenta que contribui no acompanhamento do andamento do projeto, se o realizado está de acordo com o planejado. Para Lacerda, Barbosa e Ribeiro (2011) o *Scrum* permite a criação de equipes auto-organizadas, incentivando a comunicação verbal entre os membros da equipe e todas as áreas envolvidas no projeto.

2.2 Gerenciamento do Tempo em Projetos de Software

No capitulo destinado a gerenciamento do tempo do PMBOK®, o PMI (2013) define esta área de conhecimento como os processos necessários para gerenciar a conclusão do projeto dentro do prazo planejado. Para uma boa gestão do tempo é necessário que os requisitos tenham sido bem definidos, pois todo o gerenciamento de prazo é balizado pelas decisões tomadas durante a definição do escopo (Carvalho & Rabechini Jr., 2011).

A gestão de prazos é uma das tarefas mais difíceis para um gerente de projetos. Estimar o tempo e os recursos necessários para a execução de uma atividade e organiza-las em uma sequência coerente é um grande desafio. A estimativa de prazo é ainda mais complicada em projetos de software, quando se trata de algo novo e que utilizara uma linguagem para desenvolvimento desconhecida (Sommerville, 2007).

De acordo com Sommerville (2007), a elaboração do cronograma do projeto envolve a separação do trabalho total em atividades menores, estimativa de tempo para execução destas atividades e a organização delas em uma sequência lógica. Normalmente, algumas dessas atividades são realizadas em paralelo. É necessário coordenar essas atividades paralelas e organiza-las de modo que a força de trabalho seja utilizada de forma otimizada. É importante evitar uma situação em que todo o projeto é adiado porque uma tarefa crítica está atrasada.

Para PMBOK® o gerenciamento do tempo é formado pelos processos planejar o gerenciamento do tempo, definir as atividades, sequenciar as atividades, estimar os recursos necessários, estimar o tempo necessário para execução, elaborar o cronograma e controlar o cronograma (PMI, 2013). É importante lembrar que estes processos interagem entre si e com as demais áreas que complementam o gerenciamento de um projeto, principalmente escopo e custos (Carvalho & Rabechini Jr., 2011). Já no *Scrum*, as atividades menores resultantes da divisão do trabalho total são chamadas de *Sprint*, esta é considerada a principal pratica do *Scrum*. É através das interações propostas pelo *Scrum*, durante as reuniões diárias, é que os problemas encontrados são tratados e realimentam o grupo de desenvolvimento contribuindo para o cumprimento do prazo de entrega (Carvalho & Mello, 2012).

3 Metodologia

A estratégia de pesquisa designa as diferentes formas de se abordar e analisar dados empíricos em ciências sociais. As pesquisas científicas caracterizadas por descrição, compreensão e interpretação dos fatos são denominadas qualitativas. A estratégia de pesquisa estudo de caso vai ao encontro da avaliação qualitativa, uma vez que seu objetivo é analise de uma unidade social em profundidade (Martins & Theóphilo, 2009). Para Eisenhardt (1989) a estratégia de pesquisa estudo de caso foca o entendimento da dinâmica presente dentro de configurações individuais.

Questões de pesquisa do tipo “como” são mais explanatórias e conduzem ao uso de estudo de caso (Yin, 2001). Pela característica da questão de pesquisa deste relato e por se tratar de uma avaliação qualitativa dos resultados, a estratégia de pesquisa indicada para este trabalho é o estudo de caso. O estudo de caso tipicamente combina a coleta de dados através de métodos como entrevistas, arquivos, questionários e observações conforme Eisenhardt (1989) menciona, neste relato utilizaremos entrevistas, observação e a análise de arquivos.

A principal forma de coleta de dados para este relato foi através de entrevistas. Participaram das entrevistas o Gerente do Projeto, Gerente Funcional da área de desenvolvimento do software e dois Desenvolvedores membros da equipe do projeto analisado neste relato. O Gerente de Projetos tem 30 anos, formado em Engenharia Elétrica, atua na área de gestão de projetos a 5 anos. O Gerente Funcional tem 47 anos, também formado em Engenharia Elétrica e atua no desenvolvimento de software a 20 anos. Os Desenvolvedores membros da equipe possuem 28 e 31 anos, são formados em Ciência da Computação e atuam no desenvolvimento de software a 5 e 7 anos respectivamente.

Foram elaboradas 6 questões com base no referencial teórico, onde foram abordados os problemas no cumprimento de prazos, o modelo de gestão de projetos praticado atualmente e o praticado anteriormente, qual foi a base para o desenvolvimento da metodologia de gestão de projetos utilizada, e a existência ou não de contribuições no atendimento do prazo de entrega com a aplicação de um método de gestão de projetos adequado ao perfil da organização. Além das entrevistas foram analisados os documentos internos da organização que descrevem o modelo e os procedimentos que são adotadas na gestão de projetos praticada, e os registros referentes ao planejamento inicial e a conclusão dos projetos, antes e depois da implementação da metodologia de gestão de projetos utilizada atualmente. A análise dos

dados da pesquisa foi essencialmente qualitativa não sendo aplicado nenhum tratamento quantitativo. De acordo com Eisenhardt (1989) estudos de caso tipicamente utilizam múltiplas fontes de dados, a triangulação com múltiplas fontes possibilita uma forte comprovação dos dados apurados. Ainda segundo Yin (2001), a utilização de diferentes fontes de dados, dão o suporte e fortalecem o estudo de caso. Desta forma a triangulação de entrevista, registros e análise de documentos garantem confiabilidade dos resultados apresentados neste relato.

4 O ESTUDO DE CASO

A empresa ao qual este relato se refere é uma indústria metalúrgica líder no seu segmento. Esta indústria possui aproximadamente 1800 funcionários, um faturamento anual bruto da ordem de R$ 400 milhões por ano e investe aproximadamente 4% do seu faturamento em desenvolvimento de novos produtos. A engenharia de desenvolvimento possui em torno de 100 funcionários, trabalhando em projetos interdisciplinares que abrangem as áreas de software, hardware e mecânica. A unidade de análise deste relato é o projeto de um software aplicativo que roda em plataforma Windows. Este software faz o controle dos processos de recebimento e expedição de mercadorias através da integração dos equipamentos desenvolvidos pela empresa e o ERP utilizado pelo cliente. Como já mencionado este software possui a característica de ser comercializado de forma *standard*, ou seja, quando não é realizada adequações especificas a necessidade do cliente, ou de forma customizada, onde são realizadas adequações que visam a atender necessidades especificas de cada cliente. A figura 1 apresenta o modelo de gestão de projetos deste software adotada pela organização, no início desta forma de trabalho com este produto.

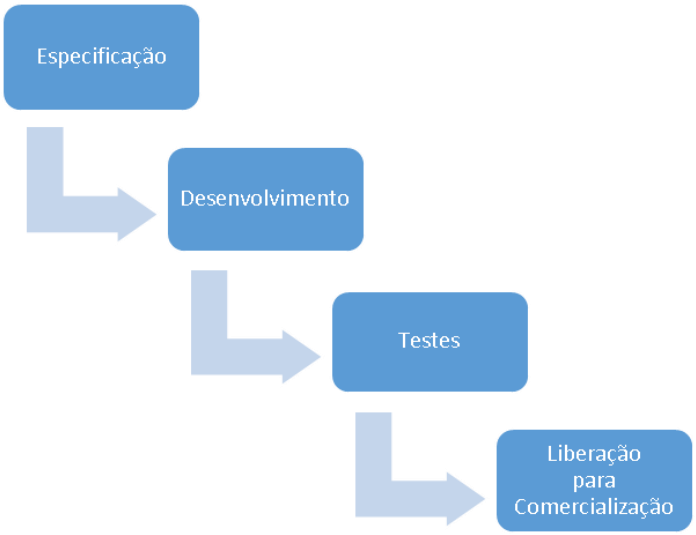

Figura 1 – Modelo de Gestão do Projeto de Software Adotado Inicialmente pela Empresa
Fonte: Autor

Neste formato, a liberação para comercialização de funcionalidades implementadas na versão *standard* ou de customizações seguiam o mesmo fluxo. Este fluxo era baseado no PMBOK® e reforça a ideia de Silva *et al* (2013) de serialização das atividades. O crescimento da comercialização deste software fez aumentar o número de customizações, cada uma era tratada como um projeto e a sequência de desenvolvimento acontecia de acordo com o prazo acordado com o cliente, sem considerar a similaridade de requisitos. Com aumento do volume

de projetos os prazos acordados começaram a sofrer atrasos, tornando-se um problema para organização.

Diante dos problemas enfrentados com prazo de entrega, o time de desenvolvimento, elaborou um modelo próprio para os projetos do software em questão. Este modelo possui aspectos descritos no PMBOK® e também no *Scrum*. Neste modelo os projetos de customização passaram a ser agrupados, cada grupo de customizações era tratado como um *Sprint* do *Scrum*. Porém, diferentemente do indicado no *Scrum*, onde normalmente cada *Sprint* possui algumas semanas, no modelo implementado os *Sprints* poderiam chegar a ter até alguns meses. A liberação da comercialização da customização e da versão standard passou a ocorrer em momento diferentes. Uma etapa de *Backlog* do Produto passou a ocorrer com todas as customizações solicitadas e também para as implementações na versão *standard*. As funcionalidades solicitadas pelos clientes passaram a ser analisadas e agrupadas por similaridade e os prazos acordados passaram a levar em conta esta análise. Também vindo do *Scrum*, foi adotado a realização de reuniões diárias, artefato para identificar problemas e soluções com maior antecedência e dinamismo.

Em relação ao modelo utilizado anteriormente, com base no PMBOK®, a característica mais marcante ainda presente era o controle documental. Boa parte dos documentos sugeridos no PMBOK® foi mantido, a estrutura de registros documentais existente na organização foi aproveitada, foram realizadas apenas adequações pontuais, porém sempre com base no PMBOK®. A figura 2 apresenta a metodologia desenvolvida e utilizada pela organização atualmente na gestão do projeto de software.

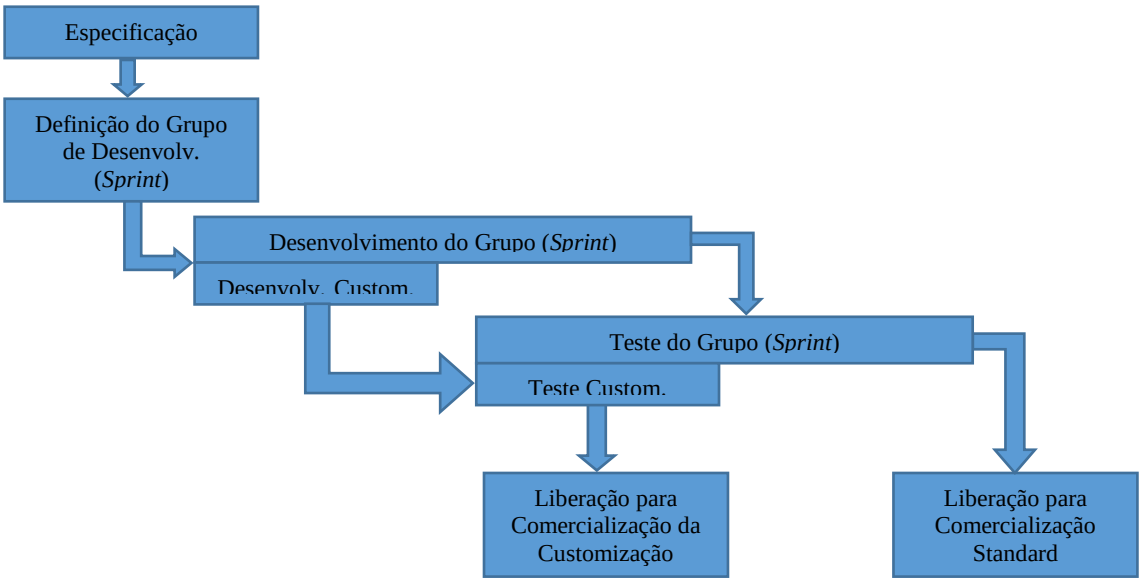

Figura 2 – Modelo de Gestão do Projeto de Software Desenvolvido pela Empresa
Fonte: Autor

6 Análise dos Resultados Obtidos

Os dados coletados na entrevista apontam que o gerente de projeto, gerente funcional e os desenvolvedores consideram que a utilização de um modelo próprio para gestão do projeto do software unidade de analise deste relato trouxe melhorias no atendimento do prazo de entrega. Eles salientam que o modelo utilizado anteriormente com base no PMBOK® não possuía a flexibilidade necessária para o tipo de projeto, ou seja, projetos de software,

reforçando a ideia apresentada por Williams (2005). A utilização de um modelo de gestão de projetos adequado as características do tipo de desenvolvimento que está sendo realizado e também da organização foi um facilitador na implementação, dando flexibilidade para solução de problemas e as tornando mais dinamicas. De acordo com os entrevistados a escolha por utilizar o *Scrum* como base para metodologia utilizada se deu devido a sua difusão entre os projetos de software conforme comenta Silva *et al* (2013) e por esta metodologia ser indicada a projetos em ambientes dinâmicos como o desenvolvimento de software, indo ao encontro do exposto por Rising e Janoff (2000). A decisão por continuar utilizando parte dos processos do PMBOK® segundo os entrevistados foi pois ele já era utilizado e para atender as necessidades documentais da organização.

Ambos disseram que a opção por adaptar modelos existentes as necessidades da organização tiveram grande importância nas melhorias alcançadas. Porém apesar das melhorias no atendimento aos prazos, relatadas pelos entrevistados, os problemas com cumprimento de prazo não foram sanados e ainda eram uma realidade, no entanto, em menor número. Isso pode ser confirmado através do registro analisados onde foi possível verificar alteração nos prazos acordados no decorrer do desenvolvimento do projeto e reforça o que diz Sommerville (2007) que a gestão do tempo em projetos é um dos pontos mais dificeis da gestão de projetos como um todo.

De acordo com os entrevistados a pratica de *Backlog* do Produto possibilitou que os projetos fossem iniciados com suas especificações dentro de um nível de detalhamento que reduziu o surgimento de dúvidas e por consequência de interrupções por indefinições e retrabalho, isso demonstra a importância desta pratica conforme indica Carvalho e Mello (2012). Outro artefato implementado e que segundo os entrevistados trouxe ganhos significativos, foi o agrupamento dos projetos, ou seja a utilização dos *Sprints*, com os projetos sendo agrupados por similaridade, o desenvolvimento do software foi otimizado, a lógica utilizada na construção das rotinas passou a levar em conta todas as funcionalidades previstas para aquele *Sprint*, reduzindo assim o tempo de desenvolvimento e a necessidade de reescrever códigos em função de funcionalidades que não haviam sido pensadas anteriormente. A prática dos *Sprints* segundo os entrevistados contribui de forma significativa para a redução dos problemas com prazo.

Todos os entrevistados também deram ênfase as reuniões diárias, o *Daily Scrum* ou *Stand Up Meeting*, pratica do *Scrum*, que segundo os entrevistados possibilita de forma rápida a tomada de decisão e realização de ações sobre possíveis problemas. Com o *Daily Scrum* as interrupções devido a problemas diários foram reduzidas, a equipe passou a saber o *status* do andamento do projeto, com as últimas ações realizadas e quais seriam próximos passos, uma vez que as respostas as questões levantadas por Rising e Janoff (2000) e que devem ser tradas nesta reunião levavam a isso. Ações também eram tomadas a fim de evitar problemas ocorridos com outros desenvolvedores, uma vez que todos estavam cientes dos acontecimentos, este ponto vai ao encontro do que Carvalho e Mello (2012) dizem sobre os problemas discutidos servirem para realimentar a equipe de projetos. No entanto, não existe uma prática de registro dos assuntos discutidos e das ações tomadas durante as reuniões diárias, desta forma as informações relatadas nas entrevistas não puderam ser confirmadas através registros documentais.

7 Conclusões

Os resultados encontrados neste relato refletem o exposto por Sommerville (2007) de que uma gestão de projetos eficiente, contribui no atendimento aos requistos do cliente, dentre eles o prazo de entrega planejado, no entanto não é garantia de que o objetivo venha a ser

alcançado. Foi possivel obervar através dos dados coletados, tanto as entrevistas como a analise documentale de registros, que um gerenciamento de projeto mais eficaz, com a utilização de um modelo adequado as necessidades da organização, tanto no que diz respeito ao tipo de projeto desenvolvido quanto as necessdidade institucionais, são fatores que contribuem para o atendimento ao requisito prazo de entrega da unidade de análise deste estudo de caso.

Fica claro através dos resultados obetidos que a simples utilização de um modelo de gerenciamento de projetos não elimina os probloemas de atendimento ao requistos de prazo de entrega do projeto. Isso corrobora com Carvalho & Rabechini Jr. (2011) que enfatizam a importancia da definição do escopo na gestão do tempo, mesmo a implementação da prática de *Backlog* do Produto, que trouxe melhoria significativa na elaboração da especificação do produto, não foi capaz de sanar os problemas de definição do escopo. Também não foi detectada nenhum potilica voltada para o gerenciamento de risco. Apesar de as reuniões diarias servirem como uma ferramenta para tratar os acontecimentos, elas não realizam o papel de identificação e prevenção ou de elaboração de medidas mitigadoras.

A utilização de um modelo de gestão moldado as necessidades da organização também mostrou contribuições interessantes em detrimento a utilização de modelos prontos. Mesmo modelos reconhecidos podem ser adaptados de forma individual ou com a integração com outros modelos, desta forma se enquadrando a realidade e caracteristica da empresa ou projeto. Os modelos prontos muitas vezes acabam tornando o processo rigido, não permitindo a flexibilidades necessária. Eles são criados muitas vezes a partir de um segmento especifico e são tornados mais generalistas com o intuito de abranger a qualquer tipo de projeto. Esta caracteristica acaba por vezes projudicando o desempenho destes modelos quando aplicados em outras organizações. Foi possivel perceber durantes as entrevistas que a implementação do modelo de gestão de projetos tornou o desenvolvimento mais fluido e eficaz, os entrevistados atribuiram isso em grande parte pelo modelo esta baseado no *Scrum*, reforçando assim o que foi apresentado por Koskela e Howell (2002). Um método mais alinhado ao dinamismo necessário para a gestão de projetos de software facilitou a utilização do modelo, e recebeu a contribuição dos membros da equipe do projeto, o que facilitou a sua implementação. Os resultados positivos que foram alcançados também foi um fator incentivador para o fortalecimento da metodologia própria de gestão de projetos.

Com base nos resultados alcançados pela organização estudada neste relato, apresenta-se como uma área a ser explorada a utilização de modelos próprios de gestão de projetos nas organizações ao invés de modelos prontos. Pesquisas adicionais neste sentido possibilitarão analisar e avaliar como as empresas vem utilizando as metodologias de gestão de projetos existentes e em que nível estas metodologias estão sendo adaptadas para a realidade de cada organização. Outra linha de pesquisa a ser seguida como complemento a este relato, é a implementação do modelo de gestão utilizado neste estudo em outra organização com características de projetos semelhantes. Com isso poderia se verificar em que grau o modelo desenvolvido é aplicável em outras organizações, quais os resultados ele apresenta e em que nível ele precisa ser adaptado. Isso mostraria se este modelo poderia servir como base para o desenvolvimento de modelos próprios de gestão de projetos em outras organizações, e os benefícios gerados por ele.

8 Referências

Amescua, A., García, J., Velasco, M., Martínez, P., Ruiz, B., Llorens, J., … Feliu, T. S. (2004). A Software Project Management Framework. INFORMATION SYSTEMS MANAGEMENT, 21(2), 78–86.

Andrade, S. C., & Tait, T. F. (2012). Uma aplicação do guia PMBOK na gestão de projetos de software. Revista Brasileira de Computação Aplicada, 4, 2-11.

Boehm, B. (Julho de 1989). Theory-W Software Project Management: Principles and Examples. IEEE Transactions on Software Engineering, 15(7), 902-916.

Carvalho, B. V., & Mello, C. H. (2012). Aplicação do método ágil scrumno desenvolvimento de produtos de softwareem uma pequenaempresa de base tecnológica. G&P, 19, 557-573.

Carvalho, M. M., & Rabechini Jr., R. (2011). Fundamentos em Gestão de Projetos: Contruindo Competências Para Gerenciar Projetos. São Paulo: Editora Atlas S.A.

Casey, V. (2010). Virtual Software Team Project Management. Journal of the Brazilian Computer Society, 16, 83-96.

Coser, A., & Morales, A. B. (2013). A Influência do Capital Intelectual sobre a Performance dos Projetos de Software. Perspectivas em Gestão & Conhecimento, 3, 196-217.

Eisenhardt, K. M. (1989). Building Theories From Case Study Research. Academy of Management Review, 14(4), 532–550.

Fernandez, D. J., Worth, F., & Fernandez, J. D. (2009). Agile project management–agilism versus traditional approaches. Journal of Computer Information Systems, 49(2), 10–18.

Koskela, L., & Howell, G. (2002). The underlying theory of project management is obsolete 1. Proceedings of the PMI Research Conference, 293–302.

Lacerda, G. S., Barbosa, A. B., & Ribeiro, V. G. (2011). Adoção do CMMI e das metodologias ágeis em empresas brasileiras. Revista Avances En Sistemas E Informática, 8(3), 33–41.

Martins, G. d., & Theóphilo, C. R. (2009). Metodologia da Investigação Científica para Ciências Sociais Aplicadas. São Paulo: Editora Atlas.

Nienaber, R. C., & Barnard, A. (2007). A Generic Agent Framework to Support the Various Software Project Management Processes. Interdisciplinary Journal of Information, 2, 149–162.

PMI. (2013). A Guide to the Project Management Body of KnowledGe (5 ed.). Newtown Square: Project Management Institute.

Rising, L., & Janoff, N. (2000). The Scrum Software Development Process for Small Teams. IEEE SOFTWARE, 17, 26-32.

Silva, D. E., Souza, I. T., & Camargo, T. (2013). Metodologias Ágeis para Desenvolvimento de Software: Aplicação e o Uso da Metodologia Scrum em Contraste ao Modelo Tradicional de Gerenciamento de Projetos. Revista Computação Aplicada, 2, 39-46.

Soares, M. d. (2003). Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software.

Sommerville, I. (2007). Software Engineering (8 ed.). Harlow: Pearson Education Limited.

Tonini, A. C., Carvalho, M. M., & Spinola, M. d. (2008). Contribuição dos modelos de qualidade e maturidade. Produção, 18, 275-286.

Williams, T. (2005). Assessing and moving on from the dominant project management discourse in the light of project overruns. Engineering Management, IEEE Transactions on, 52(4), 497–508.

Yin, R. K. (2001). Estudo de Caso Planejamento e Métodos (2 ed.). Porto Alegre: Bookman.