

ANÁLISE DO PADRÃO “SOFTWARE EXTENSION TO THE PMBOK GUIDE FIFTH EDITION”

Adriano Simões
Arnaldo Lyrio, DSc
André Barcaui, DSc

Há muito tempo as metodologias tradicionais de gerenciamento de projetos (MGP) vêm sendo empregadas nos projetos de desenvolvimento de software, mas, de uma forma geral, continua-se a entregar projetos com escopos incompletos, fora do tempo e do orçamento planejados. Desta forma, conforme observa-se pelas evidências abaixo, a área de engenharia de software carece de uma metodologia de gerenciamento de projetos mais adequada às suas peculiaridades. Segundo dados do “*Chaos Manifesto*”ⁱ de 2013:

- Somente 39% dos projetos de software são encerrados com sucesso;
- 43% dos projetos de software terminam com custo e tempo superiores ao inicialmente planejado;
- 18% dos projetos de software falham.

Já a pesquisa “*2014 IT Leadership Survey*”ⁱⁱ, indica que:

- 89% dos projetos de software não cumprem regularmente seu orçamento;
- 59% dos projetos normalmente são entregues com atraso;
- O retrabalho é responsável por pelo menos 25% do orçamento, segundo 33% dos líderes de TI entrevistados.

Com base nesses dados, o objetivo geral deste artigo foi de fazer uma análise crítica do guia SWX-PMBOK5, desde a sua conceituação de projetos, passando pela descrição do ciclo de vida de projetos diante de metodologias ágeis e análise do ciclo de vida de projetos dentro das organizações. Serão destacados também, de forma geral, os

processos e áreas de gerenciamento de projetos descritos no PMBOK 5ª edição, que foram adaptados para projetos de desenvolvimento de software.

Segundo Pressman (1995), para que um projeto de software seja bem-sucedido, é necessário que algumas dimensões sejam cuidadosamente analisadas, entre elas, o escopo do software, os riscos envolvidos, os recursos necessários, as tarefas a serem realizadas, os indicadores a serem acompanhados, os esforços e custos aplicados e a sistemática a ser seguida. A análise de todas estas dimensões é função típica do gerenciamento de projetos, a qual se inicia antes do trabalho técnico e prossegue à medida que a entrega do software vai sendo construída.

Ao mesmo tempo, Martins (2006) postula que desenvolvimento de software é uma atividade criativa, diferente dos projetos tradicionais que são baseados em atividades produtivas e processos administrativos que possuem fluxo de trabalho relativamente fixo. Sendo difícil planejar o trabalho que deverá ser feito, antes de saber mais precisamente o que vai ser produzido. Desta forma, a engenharia de software é um domínio de alto risco e requer uma abordagem diferenciada para gerenciamento de projetos.

Pelo que já foi relatado, este tipo de projeto clama por uma metodologia de gerenciamento de projetos própria, que leve em consideração suas peculiaridades, mas sem deixar de lado o formalismo e os preceitos fundamentais do gerenciamento de projetos. Desta forma, o PMI (*Project Management Institute*) em conjunto com o IEEE (*Institute of Electrical and Electronics Engineers*) em 2013 publicou uma extensão do PMBOK 5ª edição para projetos de desenvolvimento de software. Seu

objetivo foi expandir e aperfeiçoar os conceitos de gerenciamento de projetos em geral, ferramentas, técnicas, e vocabulário encontrados no “A Guide to the Project Management Body of Knowledge” (Guia PMBOK) - quinta edição, proporcionando uma definição mais específica de termos, conceitos e métodos de gestão de projetos adequados à realidade de empreendimentos de desenvolvimento de software. Nesta extensão são descritos fatores que explicam os desafios únicos para esta indústria, além de uma metodologia e sugestão de boas práticas a serem empregadas no gerenciamento de projetos desta natureza.

O PMBOK representa um marco na história da ciência gerenciamento de projetos, procurando contemplar os principais aspectos que podem ser abordados no gerenciamento de um projeto genérico. Define e promove um vocabulário comum para se discutir, escrever e aplicar em gerenciamento de projetos, possibilitando a troca eficiente de informações entre os profissionais de gerência de projetos. Essa abordagem se assemelha à empregada por outras normas como a ISO 9000iii e o CMMI^{iv}, do *Software Engineering Institute*. Os processos identificados se relacionam e interagem durante a condução do trabalho.

A extensão de software para o PMBOK 5^a ed., intitulada “*Software Extension to the PMBOK Guide Fifth Edition*”, foi um trabalho desenvolvido pelo PMI em conjunto com o “*IEEE Computer Society, Software Extension committee*”, publicado em 2013, que descreve práticas comumente aceitas para o gerenciamento de projetos de software. Aborda as práticas aplicáveis para gerenciamento de projetos desta natureza (desenvolvimento de novos softwares e modificação de software

existente). O objetivo desta extensão é ampliar e aprofundar os processos de gerenciamento do projeto, as ferramentas e as técnicas e o vocabulário encontrado no PMBOK e para fornecer termos, processos e métodos mais específicos e precisos para gerenciar projetos de software. Gerentes de projetos, incluindo profissionais certificados pelo PMI, podem aperfeiçoar suas habilidades de gerenciamento de projetos de desenvolvimento de software, melhorando seus conhecimentos relacionados aos processos, métodos, ferramentas e técnicas usadas para gerenciar projetos desta natureza.

Gerentes de projetos de software enfrentam desafios crescentes, como projetos que crescem e se tornam mais complexos à um ritmo cada vez maior, juntamente com o aumento das expectativas dos clientes e usuários; ou a necessidade de respeito à regras governamentais, à indústria e às políticas organizacionais; os desafios tecnológicos de se manter atualizados e com frequência as plataformas de hardware e software; ou ainda com a crescente inter-relação entre desenvolvimento de hardware, desenvolvimento de firmware, e desenvolvimento de software; e as considerações relacionadas com a ergonomia dos elementos humanos destes sistemas. Além disso, os projetos de software geralmente envolvem questões de segurança, confiabilidade e outros requisitos de qualidade. Deve-se considerar ainda como desafio a necessidade de expansão dos mercados mundiais e o fornecimento de produtos de software para uma ampla variedade de culturas, idiomas e formas de vida, aumentando assim o escopo e a complexidade do software a ser desenvolvido e modificado.

O principal objetivo dessa extensão para projetos de desenvolvimento de

software é completar o PMBOK com conhecimentos e práticas que podem melhorar a eficiência e a eficácia do gerenciamento de projetos de software, as suas equipes de gestão, e os membros do projeto. Tal como referido no PMBOK (2013, p.34) "boas práticas para a maioria dos projetos, na maior parte do tempo", não significa que o conhecimento descrito deve sempre ser aplicado uniformemente em todos os projetos; a organização e/ou equipe de gerenciamento de projetos é responsável por determinar o que é apropriado para um determinado projeto ou situação. O mesmo se aplica a esta extensão para projetos de software.

PORQUE O GERENCIAMENTO DE PROJETOS DE SOFTWARE É UM DESAFIO?

Toda disciplina tem aspectos únicos que a diferenciam das outras. Muitos fatores tornam os projetos de software e gerenciamento de projetos de software desafiadores. Alguns desses fatores são os seguintes:

- Software é um produto intangível e maleável; código-fonte de software é texto escrito. Na maioria dos casos, as equipes de desenvolvedores de software produzem e revisam documentos compartilhados (por exemplo, requisitos, especificações de projeto, código, e planos de teste). Desenvolvimento de Software é muitas vezes caracterizado como um processo de aprendizagem, na qual o conhecimento é adquirido e as informações são geradas durante o projeto.
- A complexidade do projeto e do produto é não linear no dimensionamento de recursos, na medição de projetos e produtos, na incerteza inicial do projeto e escopo de produto, pois os conhecimentos adquiridos vão evoluindo ao longo do projeto.
- Requisitos de Software mudam frequentemente durante um projeto de software à medida que o conhecimento é adquirido e o escopo do projeto e do produto vão surgindo.
- Requisitos de softwares novos e/ou modificados muitas vezes influenciam e são influenciados por uma organização, processos de negócio e os processos de fluxo de trabalho.
- O capital intelectual do pessoal de software é o principal bem de capital para projetos de software e organizações de desenvolvimento de software, porque o software é um produto direto de processos cognitivos humanos.
- Comunicação e coordenação entre as equipes de software e com as partes interessadas do projeto muitas vezes são carentes de clareza. Muitas das ferramentas e técnicas utilizadas na engenharia de software são destinadas a melhorar a comunicação e a coordenação.
- Criação de software requer soluções inovadoras para os problemas, a fim de criar soluções únicas. A maioria dos projetos de software desenvolve produtos exclusivos, pois a replicação de software existente é um processo simples se comparado à reprodução de artefatos físicos. Os projetos de Software são mais parecidos com os projetos de P&D (Pesquisa e Desenvolvimento) do que de construção ou projetos de fabricação.
- Os projetos de Software envolvem riscos e incertezas porque requerem inovação, o produto é intangível, e as partes interessadas não podem articular efetivamente ou acordar as necessidades a serem satisfeitas pelos produtos de software.
- O planejamento inicial e a estimativa de projetos de software são um desafio porque essas atividades

dependem das necessidades, que são muitas vezes imprecisas ou parte dos dados históricos, muitas vezes são ausentes ou inaplicáveis. A preparação de estimativas precisas também é desafiadora porque a eficiência e eficácia dos desenvolvedores de software são amplamente variáveis.

- A complexidade do produto transforma o desenvolvimento e a modificação do software e uma atividade desafiadora, devido à enorme quantidade de caminhos lógicos nos módulos do programa, combinados com valores de dados e peculiaridades de interfaces com o usuário.
- Exaustivos testes de software são impraticáveis devido à falta de tempo que seria necessário para testar todos os caminhos lógicos e interfaces com todas as combinações de entrada de dados e outros estímulos.
- O desenvolvimento de Software muitas vezes envolve inclusão dos diferentes produtos de fornecedores e desenvolvimento de interfaces com outros softwares; isso pode resultar questões de integração e desempenho.
- Em função da maior parte do software ser interligada, as informações técnicas de segurança são necessárias. Software de segurança é um grande e crescente desafio.
- Quantificação objetiva e medição de qualidade de software são difícil, devido ao carácter imaterial de software.
- Os desenvolvedores de software utilizam processos, métodos e ferramentas que estão evoluindo constantemente e são atualizados com frequência.
- Muitas vezes, o software é o elemento fundamental de um sistema que é alterado em funcionalidade,

comportamento, ou atributos de qualidade estão a ser alteradas.

- Para um mesmo produto de software pode ser necessário que opere em uma variedade de plataformas de hardware e softwares de infraestrutura.
- Software executável não é um produto independente. É executado em hardware de computação e é muitas vezes um elemento de um sistema composto de hardware diferente, outros softwares e procedimentos manuais.
- Plataformas de tecnologias, software de infraestrutura, e fornecedores de software são frequentemente alteradas ou atualizadas, o que pode exigir mudanças no software que está sendo desenvolvido.

Software é imaterial; não é uma entidade física que pode ser avaliada por medidas tradicionais (massa, volume, condutividade, gravidade específica). É, no entanto, limitada por fatores, tais como o processamento de hardware, memória disponível, e largura de banda de comunicação.

O carácter imaterial de software cria desafios para medir o estado atual do produto, o que por sua vez dificulta fiscalização e controle de um projeto de software. As abordagens tradicionais, como EAP, redes de precedência e análise de valor agregado, são adaptadas para as necessidades dos projetos de software. Essas técnicas tradicionais são ampliadas e adaptadas a outras técnicas, tais como o método iterativo e/ou desenvolvimento incremental, com as frequentes demonstrações do software parcialmente concluído.

CICLOS DE VIDA DE PROJETOS

A criação de entregáveis de software geralmente requer uma variedade de processos de ciclo de vida do projeto. Em conformidade com a norma

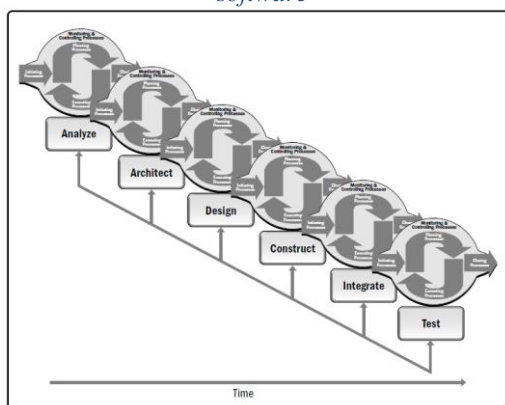
ISO/IEC/IEEE Standard 12207^v, o desenvolvimento de software inclui os seguintes processos:

- Análise: Processo de análise de requisitos de software,
- Arquitetura: Processo de projeto de arquitetura de software,
- Design: Processo de design detalhado de software,
- Construção: Processo de construção de software,
- Integração: Processo de integração de software e
- Teste: Processo de qualificação do software.

CICLOS DE VIDA PREDITIVOS

Ciclos de vida preditivo ou completamente planejados são caracterizados pela determinação precoce do escopo, tempo e custos para se executar o projeto. Como pode ser visto na figura abaixo os projetos que seguem este tipo de ciclo de vida evoluem através do sequenciamento ou sobreposição de fases, sendo que cada uma destas fases representa um subconjunto de atividades de projeto e processos de gerenciamento de projetos.

Figura 1. Sobreposição de fases sequenciais do Ciclo de Vida Preditivo de um projeto de software



Fonte: SOFTWARE EXTENSION TO THE PMBOK GUIDE FIFTH EDITION, PMI, 2013

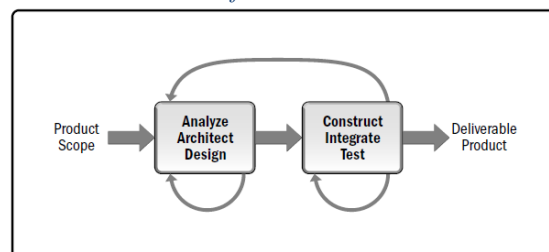
CICLOS DE VIDA ITERATIVOS E INCREMENTAIS

Ciclos de vida iterativos e incrementais são os ciclos em que as fases do projeto ou iterações, repetem uma ou mais atividades do projeto, de forma intencional, à medida que o entendimento do produto aumenta a cada iteração. O produto será obtido através de sucessivas iterações, quantas forem necessárias para o atendimento total dos requisitos do produto.

Projetos que seguem este tipo de ciclo de vida são especialmente úteis quando se pretende fazer entregas parciais e intermediárias do produto. Projetos grandes e complexos são muito beneficiados por ciclos iterativos, já que este tipo de ciclo reduz os riscos ao permitir que a equipe assimile o “feedback” e lições aprendidas nas iterações anteriores.

Estes ciclos de vida são aplicáveis em projetos de software cujas exigências são muitas vezes modificadas ao longo do projeto, além de termos as dimensões “tempo” e “escopo” modificando constantemente as estimativas de “custo”. Desta forma faz-se necessário limitar um ou mais dos três fatores de restrição, diminuindo-se assim os “trade-offs” de requisitos, tempo e custo, à medida que a equipe do projeto vai melhorando sua compreensão do produto.

Figura 2. Um projeto de software com ciclo de vida dividido em com duas etapas, cada uma com três fases iterativas



Fonte: SOFTWARE EXTENSION TO THE PMBOK GUIDE FIFTH EDITION, PMI, 2013

CICLOS DE VIDA ADAPTATIVOS

Ciclos de vida adaptativos (também chamados de direcionados à mudança ou utilizadores de métodos ágeis) têm como principal característica serem reativos (e adaptativos) a altos níveis de mudanças e requer o envolvimento contínuo das partes interessadas. São também interativos e incrementais.

- O termo “ágil” não caracteriza um ciclo de vida do projeto; é um termo utilizado para caracterizar determinados atributos que ciclos de vida adaptativos compartilham em diferentes graus, tais como:
- Entregas de incrementos de software são produzidas de uma forma periódica.
- Os tempos de cada iteração variam de diária, semanal ou mensal, mas geralmente não mais que mensalmente.
- Ciclos de iteração adaptativos são, muitas vezes, de igual duração (ou seja, é "time boxed"), mas alguns ciclos podem ser de maior ou de menor duração em casos de exceção.
- Incrementos de entregáveis não são necessariamente produzidos por cada ciclo de iteração. Desta forma, incrementos e iterações são distintos.
- Os requisitos, design, e o produto de software surgem à medida que o projeto evolui.
- O cliente, um representante do cliente, e/ou usuário experiente está constantemente envolvido. Sua participação inclui a observação de manifestações periódicas de trabalho, avaliações das entregas de software no término de cada ciclo de desenvolvimento iterativo com seus incrementos, de forma periódica (ou seja, diária, semanal, quinzenal ou mensal). Além disso, este ator oferece orientação para o desenvolvimento de novos produtos com base nas demonstrações do produto e nas limitações de escopo

do projeto (cronograma, orçamento e recursos).

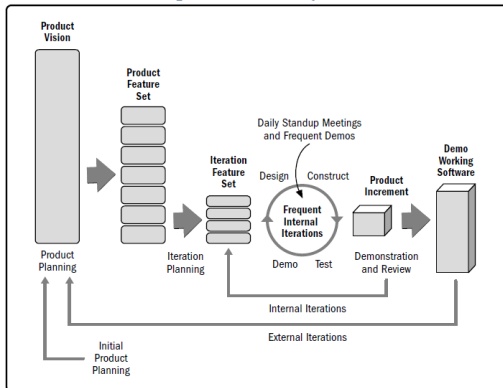
- As equipes de desenvolvimento de software adaptativo são pequenas (com 10 ou menos componentes) e são auto organizadas. Neste modelo, grandes projetos podem ser compostos por várias pequenas equipes.
- Todos os membros de cada equipe de desenvolvimento de software são atribuídos a um projeto por vez.
- Cada equipe de desenvolvimento de software possui profissionais generalistas e especialistas na quantidade necessária para realizar as atividades de trabalho; especialistas funcionais podem ser envolvidos periodicamente ou sempre que necessário.

Para projetos de software do tipo adaptativos, o produto a ser desenvolvido é frequentemente caracterizado por Histórias (“*Stories*”), Casos de Uso (“*Use cases*”) ou funcionalidades implementadas no lugar de módulos ou linhas de código. Uma equipe de projetos adaptativos aprende rapidamente a mensurar a quantidade de trabalho que deve ser executado durante o ciclo iterativo. A experiência acumulada também permite que as equipes possam estimar com bastante precisão a quantidade de tempo necessário para completar a implementação de um conjunto de funcionalidades. Uma medida de produtividade é a **velocidade**, que é a relação de produtos de trabalho executados pela quantidade de esforço dispendido por uma equipe durante um ciclo iterativo. A medida de velocidade pode ser acumulada durante as iterações e usada para rastrear o progresso planejado versus o atual, permitindo-se assim prever o custo final e a data de encerramento do projeto. Isto é similar à forma como o conceito de valor agregado é usado para rastrear a fase de

construção de projetos de ciclo de vida preditivos.

Fonte: SOFTWARE EXTENSION TO THE PMBOK GUIDE FIFTH EDITION, PMI, 2013

Figura 3. Método para o desenvolvimento adaptativo de software



Fonte: SOFTWARE EXTENSION TO THE PMBOK GUIDE FIFTH EDITION, PMI, 2013

Desta forma as principais características do ciclo de vida externo altamente adaptativo do ciclo de vida interno adaptativo são:

- O cliente fica muito mais próximo do projeto.
- O cliente pode obter demos diários do produto.
- O cliente fornece constantemente insumos para a próxima iteração.
- É o cliente que aceita, solicita revisão e rejeita funcionalidades adicionadas.
- Os incrementos de softwares podem estar disponíveis para entrega no ambiente do usuário diariamente, se for desejado.

CICLO DE VIDA ALTAMENTE ADAPTATIVOS

A Figura 4 ilustra um método de desenvolvimento de software altamente adaptativo, onde diariamente demonstrações de software são produzidas para o cliente ou alguém indicado por ele que passa a estar completamente envolvido no processo de desenvolvimento. Seu envolvimento é tão grande que frequentemente passa a fazer parte da equipe de desenvolvimento. Diariamente ele descreve uma “User Story” ou um cenário que deva ser desenvolvido para se obter determinadas funcionalidades. As novas funcionalidades são então adicionadas e os cenários são aplicados.

CONCLUSÃO

Como podemos observar ao analisar os dados da Tabela 1, a equipe que formulou as adaptações do Guia PMBOK para projetos de desenvolvimento de software, propôs um número maior de alterações nas áreas de gestão de Tempo, Custos e Escopo, nesta ordem.

Tabela 1 - Quantitativo de alterações propostas na Extensão de Software para o PMBOK 5

Capítulo do PMBOK 5	Área de conhecimento	Alterações propostas			Total por área de conhecimento	% por área de conhecimento
		Entradas	Ferramentas e técnicas	Saídas		
6	Gerenciamento do tempo do projeto	12	21	8	41	28%
7	Gerenciamento dos custos do projeto	8	18	4	30	20%
5	Gerenciamento do escopo do projeto	4	9	6	19	13%
10	Gerenciamento das comunicações do projeto	3	6	5	14	10%
11	Gerenciamento dos riscos do projeto	2	8	3	13	9%
4	Gerenciamento da integração do projeto	4	1	5	10	7%
9	Gerenciamento dos recursos humanos do projeto	2	4	1	7	5%
13	Gerenciamento das partes interessadas no projeto	2	4	1	7	5%
8	Gerenciamento da qualidade do projeto	1	1	4	6	4%
12	Gerenciamento das aquisições do projeto	0	0	0	0	0%
TOTAL		38	72	37	147	100%
		%	26%	49%	25%	100%

Figura 4. Ciclo externo de desenvolvimento para um modelo adaptativo de desenvolvimento de software

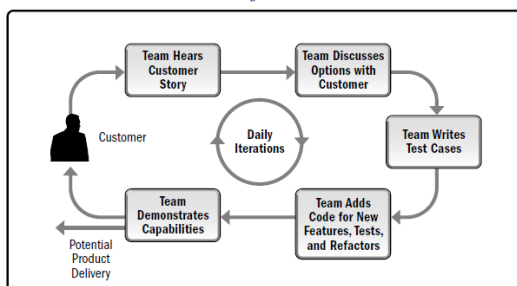
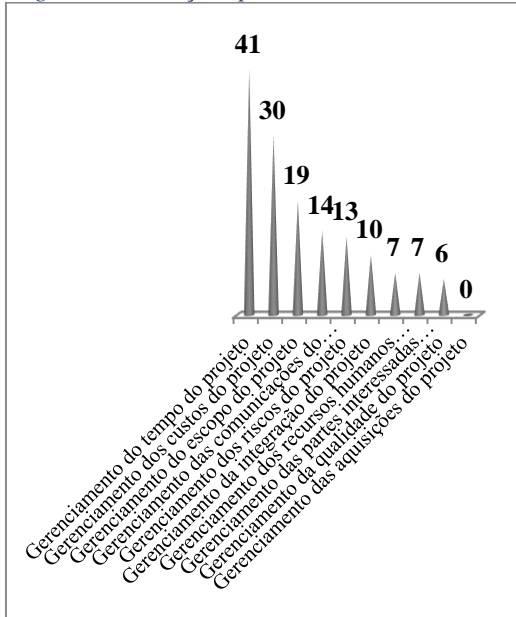


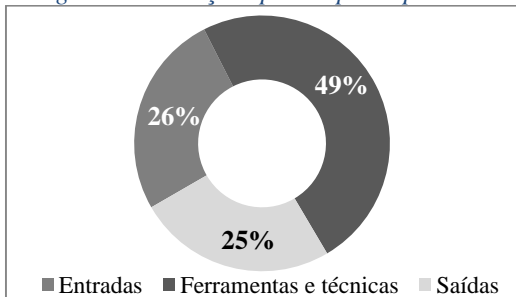
Figura 5. Alterações por área de conhecimento



Observa-se que as áreas de gestão de Tempo, Custos e Escopo foram responsáveis por 61% do total de alterações. Isto pode ser facilmente explicado como sendo uma resposta do PMI e do IEEE às frequentes constatações de insucessos em projetos de software, justamente por problemas de gestão nas três áreas citadas, como podemos observar nos dados apresentados abaixo.

Percebemos também que as alterações propostas na extensão foram concentradas nas "Ferramentas e Técnicas" dos processos, sendo responsável por 49% das alterações.

Figura 6. Alterações por etapa do processo



Isto nos sugere uma preocupação em propor um melhor ferramental para a gestão de projetos, admitindo-se que aquele sugerido no PMBOK é

insuficiente para projetos de desenvolvimento de software.

Diante deste cenário pode-se constatar que a SWX-PMBOK5 é uma tentativa de reação à baixa popularidade e participação efetiva do Guia PMBOK em projetos de software. É possível atribuir esta baixa utilização e popularidade à natureza genérica do padrão, que visa atender projetos das mais diversas naturezas, não se prendendo a especificidades das diferentes naturezas de projetos. A questão é que pessoas com pouca informação sobre a real utilidade do PMBOK, e que possuem uma visão "engessada" sobre a utilidade do mesmo, estão gerenciando e liderando projetos, sendo muitos deles formadores de opinião. Deve-se mais uma vez ratificar que o PMBOK é uma orientação genérica e que nem todos os projetos precisam seguir "à risca" todos os preceitos e preenchimentos de documentos, adoção de ferramentas e "burocracia" que os 47 processos do guia sugerem. O gerente de projetos de desenvolvimento de software comumente utiliza metodologias e boas práticas de desenvolvimento de software consolidadas no mercado, estas boas práticas não necessariamente estão "preocupadas" em propor melhorias na gestão do projeto, mas sim em propor melhores metodologias de desenvolvimento do produto (software). Dentre estas metodologias podemos citar as metodologias ágeis, conceitos de manufatura "lean", entre outros.

A questão é que estas metodologias não orientam os líderes e gerentes de projetos a conduzir "melhor" um projeto desta natureza, e desta forma observa-se a ocorrência de um "gap" entre boas práticas de desenvolvimento e ciclos de produção e a gestão destes projetos. Desta forma a SWX-PMBOK é uma tentativa de integrar as boas práticas de desenvolvimento de

software com os 47 processos do PMBOK. Sendo assim, podemos observar que as adaptações foram em sua totalidade de acréscimo de informações, ou seja, em momento algum houve retirada de algum artefato ou ferramenta/técnica proposta para o planejamento, condução e acompanhamento dos processos de gestão de projetos, apresentados no PMBOK. Um fato curioso sobre a obra analisada é o baixíssimo grau de citações à mesma, em publicações, fóruns e outros canais de discussões sobre o tema em pauta. Isto pode ser comprovado fazendo-se uma busca ao scholar.google.com.br pelo nome da obra. Segue abaixo a URL com o nome da obra no termo de busca, acessada no dia 26 de fevereiro de 2015, que retornou apenas 5 resultados:

<https://scholar.google.com.br/scholar?q=Software+Extension+to+the+PMBOK>

CONCLUSÕES

Embora, dada à importância do tema e a relevância da obra em análise, não há estudos que se baseiam nesta proposta. O que reforça a ideia de que o padrão não vem sendo adotado pela comunidade de gestão de projetos de desenvolvimento de software.

Nota-se ainda uma baixíssima adesão ao padrão SWX-PMBOK em projetos de desenvolvimento de software e ainda a massiva inobservância ao mesmo em artigos acadêmicos, possivelmente relacionado ao fato de que gerentes de projetos e líderes da área de

desenvolvimento de software não “acreditam” nos benefícios que o PMBOK possam proporcionar em detrimento a adoção de metodologias próprias para a condução de projetos desta natureza. Este comportamento representa um engano, já que a adoção de metodologias próprias e mais adequadas para a execução desses projetos não invalidam a utilidade de se adotar boas práticas de gerenciamento de projetos.

O problema que estigmatiza a área de desenvolvimento de software, composta por equipes pouco produtivas, produtos de baixa qualidade e com o não cumprimento das metas de cronograma e custos está mais relacionada à baixíssima qualificação de profissionais gerentes de projetos que conduzem tais empreendimentos. Observa-se

comumente no mercado a escolha de gerentes de projetos com pouca qualificação em gestão de projetos e menos ainda na área de desenvolvimento de software, para conduzir projetos desta natureza. Desenvolvimento de software é uma área extremamente complexa tanto pelas suas peculiaridades técnicas quanto pela natureza intangível do produto gerado. O produto gerado (software) é fruto de trabalho intelectual de profissionais com motivações e anseios bem peculiares e diferentes daqueles outros profissionais que desenvolvem seu trabalho de uma forma padronizada, coordenada e que raramente dependem de “insights” e inspirações alimentadas por sua vasta e multidisciplinar “bagagem” técnica.

ⁱ THE STANDISH GROUP. The 2013 Chaos Manifesto. In:

<http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>, acessado em 20 de fevereiro de 2015.

ⁱⁱ BLUEPRINT SOFTWARE INC. 2014 IT Leadership Survey. In:

<http://www.blueprintsys.com/lp/it-leader-2014/>, acessado em 20 de fevereiro de 2015.

ⁱⁱⁱ ISO 9000. In: http://www.iso.org/iso/iso_9000, acessado em 24 de maio de 2015.

^{iv} CMMI for Development, Version 1.3, Carnegie Mellon University, ano 2010. p. 5.

^v ISO/IEC/IEEE Standard 12207:2008. Systems and Software Engineering—Software Life Cycle Processes.