IV SIMTEC
Simpósio de Tecnologia da Fatec Taquaritinga

TÉCNICAS DE ESTIMATIVA DE CUSTO DE SOFTWARE: CONTRIBUIÇÕES PARA O GERENCIAMENTO DE CUSTOS EM PROJETOS

SOFTWARE COST ESTIMATION TECHNIQUES: CONTRIBUTIONS TO COST MANAGEMENT IN PROJECTS

Pedro Henrique Camargo de Abreu – phcamargo1997@gmail.com
Fernando Rodrigues de Amorim – fernando.amorim@fatectq.edu.br
Faculdade de Tecnologia de Taquaritinga (FATEC) – São Paulo – Brasil

RESUMO

O custo representa uma preocupação para qualquer tipo de projeto, pois é um elemento responsável por determinar a viabilidade, a atratividade e o sucesso do projeto. Quando o foco é direcionado para projetos de software, é evidente que a viabilidade se faz presente, mas os mecanismos e ferramentas para atingir este objetivo, devem ser compreendidos de forma diferente em comparação com os demais projetos, pois o desenvolvimento de software é composto por algumas atividades gerenciáveis e quantificáveis, onde a geração de estimativas pode representar uma alternativa atrativa para maximizar as chances de sucesso acerca dos custos no projeto, tanto que existem algumas técnicas já consolidadas na literatura. Dessa forma, o objetivo do presente trabalho é abordar e analisar as influências e as contribuições que tais técnicas promovem para o controle e monitoramento dos custos em projetos de software. A pesquisa decorreu de forma exploratória, conduzida por meio de uma revisão bibliográfica, a fim de proporcionar o levantamento de informações e a investigação das técnicas de COCOMO, Putnam e Análise de Pontos de Função. Os resultados obtidos, mostram a importância da aplicação destas técnicas para a criação de estimativas e indicadores capazes de contribuir com a análise dos riscos operacionais e promover o alinhamento com o gerenciamento dos custos.

Palavras-chave: Estimativa. Gestão de projetos. Viabilidade. Riscos em projetos.

ABSTRACT

Cost is a concern for any type of project, because it is an element responsible for determining the viability, attractiveness and success of the project. When the focus is directed to software projects, it is clear that this need for viability is present, but the mechanisms and tools to achieve this goal need to be understood differently in comparison with other projects, because software development is composed of some manageable and quantifiable activities, where the generation of estimates may represent an attractive alternative to maximize the chances of success related to project costs, and there are some techniques already consolidated in the literature. Thus, the objective of the present work is to approach and analyze the influences and the contributions that these techniques promote for the control and monitoring of costs in software projects. The research was conducted in an exploratory way, through a bibliographical review, to provide the information acquisition and the investigation of the techniques of COCOMO, Putnam and Function Point Analysis. The results demonstrate the importance of applying these techniques to create estimates and indicators capable of contributing to the analysis of operational risks and to promote alignment with cost management.

Keywords: Estimation. Project management. Viability. Risks in projects.

1 INTRODUÇÃO

Espera-se que um gerente de projetos seja capaz de gerenciar efetivamente o custo, o tempo e os riscos de todas as mudanças no projeto, contemplando-os dentro das restrições e esteja pronto para lidar com qualquer desafio. Para coordenar projetos de forma eficaz, é necessário que os gestores tenham conhecimento de boas práticas de gerenciamento de projetos, sendo que o gerente do projeto deve entender as implicações das mudanças a fim de gerenciá-las, de tal forma que todos os objetivos sejam obtidos dentro do tempo, orçamento e restrições de qualidade (SCHOONWINKEL et al., 2016).

Pinto e Slevin (1988) identificaram duas faces no gerenciamento de projetos. A face interna, leva em consideração os objetivos de manutenção em relação aos custos, tempo e qualidade, e está intimamente ligada às ações do gerente de projeto e da equipe do projeto. A face externa, está relacionada ao usuário, através do uso do produto e a resolução do problema que originou o projeto.

De acordo com Munns e Bjeirmi (1996), a definição do projeto e a tomada de decisão no início do processo, são fundamentais para o sucesso geral. Os esforços da equipe, não redimirão o projeto que está condenado a falhar por causa da fraca tomada de decisão inicial. Existe, no entanto, a possibilidade de que um gerenciamento de projetos fraco, possa ameaçar um projeto potencialmente bom. A viabilidade não deve ser limitada à viabilidade do processo de desenvolvimento, mas deve ser estendida ao uso subsequente.

Segundo Shenrar e Dvir (2010), alcançar os objetivos de tempo e orçamento, indica que o projeto foi gerenciado de forma eficiente. No entanto, a maioria dos projetos faz parte da gestão estratégica de suas organizações e devem ser avaliados com base em suas contribuições para os resultados do negócio (efetividade). De acordo com o PMI (2013), para cumprir o custo aprovado, é necessário que se promova o gerenciamento de custos, a partir da execução de práticas de planejamento, estimativas, orçamentos, financiamentos, gerenciamento e controle de custos.

O processo de gerenciamento de projetos de desenvolvimento de software, segue as regras básicas de gerenciamento de projetos, mas também possui características específicas. Um gerente de projeto de desenvolvimento de software tem que lidar com desafios e

contratempos que são específicos do setor de TI (DESPA, 2014). De acordo com Shekhar e Kumar (2016), os métodos algorítmicos de estimação de custos em projetos de software fazem uso de cálculos e equações para a elaboração das estimativas, que se apoiam em características importantes e específicas deste âmbito de projetos, tais como: linhas de código, pontos de função, linguagens de programação, etc.

Despa (2014) afirma que o campo de desenvolvimento de software é caracterizado por uma alta dinâmica de tecnologias e padrões. As linguagens de programação evoluem, os novos *frameworks* surgem com uma velocidade surpreendente, as interfaces de usuário tornam-se cada vez mais diversas, exigindo que os softwares trabalhem com uma grande variedade de dispositivos.

Sendo assim, o objetivo do presente trabalho é abordar e analisar as influências e contribuições que as técnicas de estimativa de custo promovem para o controle e monitoramento dos custos em projetos de software, abordando tanto as boas práticas a serem adotadas do ponto de vista gerencial, quanto ao esforço e maturidade necessários para o planejamento operacional em projetos de software.

2 PLANEJAMENTO ESTRATÉGICO DE PROJETOS

Porter (2004) afirma que é preciso analisar e alinhar uma companhia ao seu meio ambiente para consolidar uma estratégia competitiva. Dessa forma, é fundamental que os gestores estejam cientes do ambiente em que a organização está inserida, a fim de garantir e assegurar condições suficientes para propor estratégias adequadas e que potencializem a sua ação no mercado.

De acordo com Tiffany e Peterson (1998), o planejamento estratégico não é uma ciência que mostra o certo e o errado em relação ao futuro, e sim, uma ferramenta que fornece à organização, uma visão do futuro, permitindo que a empresa aproveite as oportunidades e explore suas potencialidades no mercado.

Em um ambiente organizacional, os projetos são a maneira de implementar estratégias. Portanto, os objetivos do projeto devem estar diretamente ligados aos objetivos estratégicos. A eficácia em qualquer projeto, é determinada pela forma como o projeto contribui para o alcance dos objetivos estratégicos da organização, e a eficiência é determinada pelo quão bem o projeto foi conduzido (PATAH, 2010). Segundo Porter (2004), é imprescindível que sejam aceitas alternativas estratégicas que ofereçam algumas relações possíveis entre a parcela de mercado e

a rentabilidade necessária. O fato é que a adoção de determinada estratégia deve refletir as potencialidades da organização, a fim de avaliar seus riscos e garantir a eficácia da sua aplicação.

Todo projeto é único, pois possuem objetivos específicos e riscos que, na maioria das vezes, não foram vivenciados anteriormente. Dessa forma, se as condições e os eventos incertos ocorrerem, gerarão riscos ao projeto que irão afetar os seus objetivos. Naturalmente, em um novo gerenciamento de projetos, podem ser resolvidos riscos potenciais com base em experiências de outros projetos, sob declarações de similaridade (IACOB, 2014).

No entanto, a fim de apoiar uma gestão eficaz do risco, o processo de gestão do conhecimento deve ir além da coleta e estruturação de informações. Um aspecto crucial é a capacidade desse processo para orientar a escolha das técnicas que devem ser aplicadas em diferentes contextos, dependendo do próprio projeto e da maturidade que a empresa apresenta (KARADSHEH et al., 2012).

3 GERENCIAMENTO DE CUSTOS EM PROJETOS

Segundo Tichacek (2006), o custo é um denominador comum de fácil compreensão, usado para racionalizar a decisão em meio a custos adicionais de aceleração de um cronograma existente ou para aceitar custos maiores no desenvolvimento de um plano de execução otimizado.

De acordo com Hansen e Mowen (2006), o gerenciamento de custos identifica, coleta, mede, classifica e relata informações que são úteis aos gerentes para determinar os custos de produtos, clientes, fornecedores e outros objetos relevantes, além de contribuir com o planejamento, controle, melhorias contínuas e tomada de decisão no projeto.

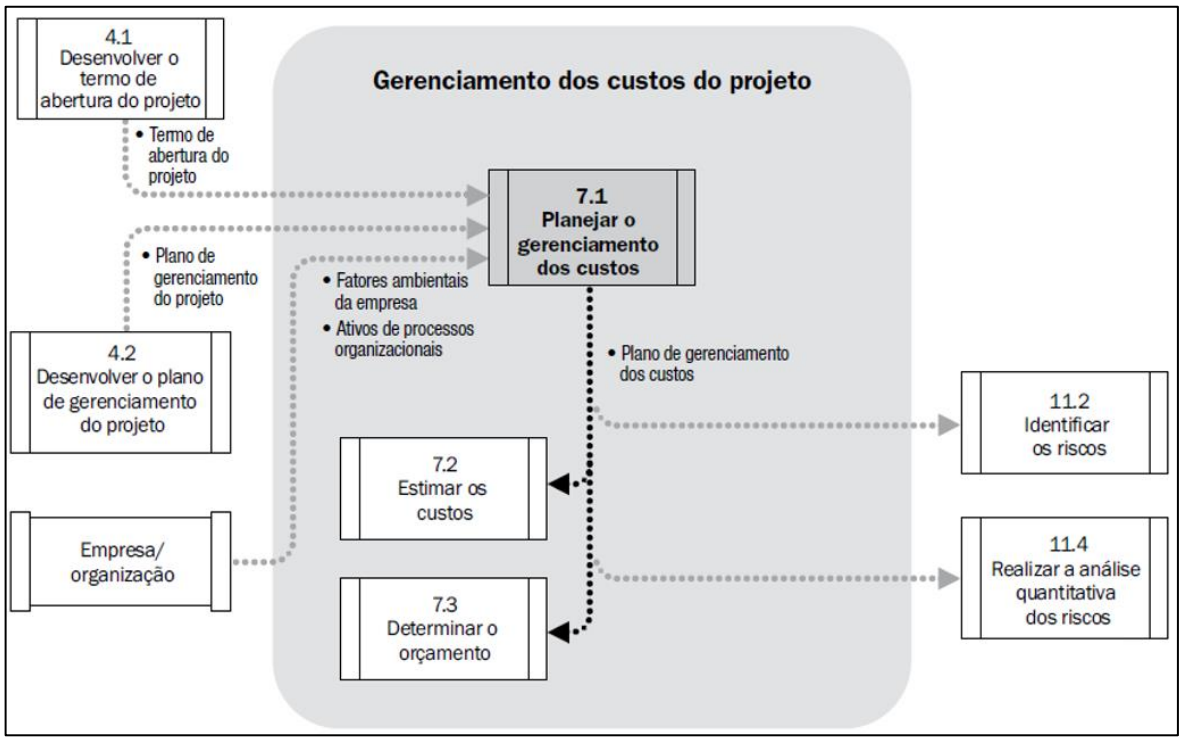
Todas as organizações identificam o gerenciamento de custos como um objetivo e uma prática. O gerenciamento de custos, no entanto, difere muito da contabilidade para eles, e, embora muitas organizações acreditam que estão gerenciando os custos, estão, de fato, apenas contabilizando-os, sendo que tais organizações se privam dos benefícios que os processos de gerenciamento oferecem (TICHACEK, 2006). Segundo Kaplan e Cooper (1998), o gerenciamento de custos é fundamental para a sustentação da competitividade das empresas no mercado atual.

Planejar o gerenciamento dos custos é o processo responsável por definir as políticas, os procedimentos e os documentos necessários para o fortalecimento do planejamento,

gerenciamento, despesas, e controle dos custos relacionados ao projeto. O principal benefício obtido com a consolidação deste processo, é a obtenção de orientações e instruções acerca dos riscos a serem gerenciados ao longo do projeto (PMI, 2013).

A Figura 1 a seguir demonstra o fluxo de dados para planejar o gerenciamento dos custos em projetos.

Figura 1 – Planejar o gerenciamento dos custos: diagrama do fluxo de dados



Fonte: PMI (2013, p. 196)

Os processos de gerenciamento dos custos do projeto, juntamente com as suas ferramentas e técnicas associadas, devem ser documentados no plano de gerenciamento dos custos, que é um importante componente do plano de gerenciamento do projeto (PMI, 2013).

Segundo Kujala et al. (2014), a gestão dos custos de um projeto, começa durante a fase de marketing e vendas do projeto, onde são criadas as primeiras estimativas de custo. Durante esta fase, as estimativas são especificadas iterativamente para fornecer informações precisas e confiáveis para serem usadas na definição do preço do projeto a ser entregue. Durante a fase de especificação do projeto, um orçamento provisório é definido, uma abordagem de especificação de custo é explorada para garantir especificações econômicas e, normalmente, as letras de intenções com os principais fornecedores são assinadas. Antes da fase de implementação, o orçamento do projeto é criado com base nas estimativas de custo mais recentes.

Na fase de planejamento do projeto, os recursos para as tarefas de trabalho são alocados, os fluxos de caixa são planejados e, normalmente, as contingências de custos são definidas. Durante as fases de planejamento e implementação, os custos atuais do projeto são monitorados. Além disso, as estimativas e previsões de custos são constantemente atualizadas e comparadas com o orçamento do projeto. Na fase de implementação, as receitas do projeto são monitoradas, o faturamento é analisado e as contingências de custos são executadas. Em alguns projetos, o financiamento também é assegurado durante a fase de implementação. Após a entrega do sistema, os custos operacionais são monitorados juntamente com o cliente, os custos da garantia são verificados, e, por fim, o ciclo de aprendizado das fases anteriores deve ser criado (KUJALA et al., 2014).

De acordo com Venkataraman e Pinto (2008), o gerenciamento efetivo de custos em projetos, é um processo extremamente complexo que começa muito cedo durante o ciclo de vida do projeto. Dentre os fatores que influenciam o sucesso, se mostra fundamental a utilização de um sistema razoável e preciso para estimar os custos.

Turner e Simister (2000) destacam algumas considerações importantes para que se possa criar um sistema de estimativa de custos, dentre eles: uma definição clara, completa e inequívoca do projeto e o escopo de trabalho envolvido; um cronograma de trabalho priorizado e detalhado, extraído da a partir da divisão do trabalho, a fim de possibilitar o acompanhamento do progresso das tarefas individuais; orçamentos significativos, onde cada pacote de trabalho é alocado à sua parcela adequada do orçamento total, proporcional ao trabalho envolvido; e um sistema de contabilidade de custos que irá acumular custos e alocá-los às suas contas relevantes.

4 PROCEDIMENTOS METODOLÓGICOS

Tendo em vista a proposta e o objetivo geral, o presente trabalho foi conduzido por meio de uma pesquisa exploratória, a fim de proporcionar o levantamento de informações capazes de contribuir com a análise do tema proposto. De acordo com Gil (1999), este tipo de estudo visa proporcionar um maior conhecimento para o pesquisador acerca do assunto, a fim de que esse possa formular problemas mais precisos ou criar hipóteses que possam ser pesquisadas por estudos posteriores.

Dessa forma, foi realizada uma revisão da literatura, com o intuito de contribuir com uma aproximação e consolidar um relativo questionamento acerca dos objetos analisados. Moreira (2004) afirma que estudos desta natureza, analisam a produção bibliográfica em

determinada área temática, a fim de proporcionar uma visão geral ou fornecer um relatório capaz de evidenciar novas ideias, métodos, ou investigar subtemas que tenham recebido maior ou menor ênfase na literatura selecionada.

O estudo e a compreensão do referencial abordado, foram desempenhados de maneira sistemática, a fim de consolidar a identificação dos benefícios e contribuições a serem obtidas por meio da aplicação de técnicas algorítmicas para estimação de custos em projetos de desenvolvimento de software, enfatizando assim, as práticas a serem necessárias para a redução dos riscos de viabilidade nesses projetos.

5 RESULTADOS E DISCUSSÃO

5.1 Método COCOMO

A construção de estimativas acerca dos esforços de desenvolvimento de software, pode ser compreendida como um processo de previsão do uso mais realista do esforço necessário para se desenvolver um produto de software baseado em insumos incompletos, incertos e/ou duvidosos. Estimativas de esforço podem ser usadas como insumos para planos de projetos, planos de iteração, orçamentos e análises de investimentos (RAMACHARAN; RAO, 2013).

O *Constructive Cost Model* (COCOMO) ou Modelo de Custo Construtivo foi proposto por Barry Boehm em 1981. O foco principal do COCOMO é a estimativa da influência de 15 fatores de custo ao processo de desenvolvimento (SINGH et al., 2013). A intenção deste modelo, é fornecer uma abordagem capaz de lidar com problemas de engenharia de software, por meio da perspectiva da economia humana, bem como da percepção de programação (RAMACHARAN; RAO, 2013).

De acordo com Singh et al. (2013), a palavra "construtivo" fornece a ideia de que a complexidade do modelo pode ser facilmente compreendida devido à sua abertura e adaptação, sendo que estas características estão diretamente relacionadas à capacidade estimativa do modelo. Desde o início das técnicas de desenvolvimento de software, muitos foram os esforços desempenhados a fim de contribuir com a geração de estimativas em projetos; sendo que o COCOMO possui uma boa documentação para auxílio, além de sua transparência de uso e capacidade de refletir as práticas de desenvolvimento de software que perduram até os dias atuais.

O COCOMO foi derivado de um banco de dados de 63 projetos, executados entre 1964 e 1979 pela empresa americana TRW Systems, Inc. Os projetos considerados durante esse período, distinguiam-se em relação ao tipo de aplicação, tamanho e linguagem de programação utilizadas (HEEMSTRA, 1990).

De acordo com Singh et al. (2013), Boehm introduziu três níveis do modelo de estimativa:

- COCOMO 81 Básico: computa o esforço e o custo de desenvolvimento, considerando uma estimativa do tamanho do programa (em LOC). A fórmula usada é:

$$E = a * (KLOC)^b \quad (1)$$

Em que:

- E: é o esforço aplicado (pessoa-mês);
- KLOC: tamanho do código (mil linhas de código-fonte);
- a: parâmetro específico do domínio, depende do tipo de projeto, isto é, se é orgânico, semi-destacado ou incorporado;
- b: parâmetro específico do domínio, depende do tipo de projeto, isto é, se é orgânico, semi-destacado ou incorporado;

- COCOMO 81 Intermediário: computa o esforço e o custo de desenvolvimento considerando uma estimativa do tamanho do programa e um conjunto de direcionadores de custo (avaliações subjetivas do produto, do hardware, do pessoal e dos atributos do projeto). A fórmula usada é:

$$E = a * (KLOC)^b * EAF \quad (2)$$

Em que:

- EAF: é o fator de ajuste do esforço. Valores típicos para o EAF variam de 0,9 a 1,4.

- COCOMO 81 Avançado (ou Detalhado): além das características do COCOMO intermediário, inclui uma avaliação do impacto dos direcionadores de custo sobre cada etapa do desenvolvimento.

Como a estimativa no modelo COCOMO é realizada nos estágios iniciais do desenvolvimento de software, muitas vezes pode levar a falhas de estimativa. Como resultado desses problemas, a nova versão do COCOMO, denominada de COCOMO II, foi desenvolvida em 1990 e usa um conjunto mais amplo de dados. Ela usa linhas de código fonte, pontos de função e pontos de objeto como entradas, além de incluir algumas modificações aos drivers multiplicadores de custos do COCOMO anterior. O resultado é obtido na forma de estimativas de tamanho e esforço posteriormente desenvolvidas em um cronograma do projeto (SHEKHAR; KUMAR, 2016).

5.2 Modelo de Putnam

De acordo com Pillai e Nair (1997), o modelo de Putnam pode ser utilizado na análise de processos, a fim de avaliar o impacto de um cronograma apertado, e prever custos a partir da evolução do projeto. Segundo Shekhar e Kumar (2016), este modelo examina muitos projetos de software e depende da distribuição de mão-de-obra.

O modelo prevê, de forma simples, uma estimativa de custos para o desenvolvimento de um software, porém sua capacidade em prever o tempo de desenvolvimento e requisitos de quantidade de mão-de-obra em um estado inicial do desenvolvimento, não são satisfatórios, acarretando em uma previsão inicial (*early prediction*) falha. Os modelos lineares simplificados falham em capturar a variação inicial das observações, sendo que existem duas formas de tentar resolver essa questão: ajustar a curva com um grau mais elevado para o conjunto de dados e/ou encontrar uma alternativa de transformação para os eixos (PILLAI; NAIR, 1997).

De acordo com Haufe (2001), a estimativa de Putnam possibilita a troca do tempo pelo esforço. Ao acrescentar pessoas num projeto, o tempo necessário para o desenvolvimento é reduzido, entretanto, aumenta-se o custo do projeto. Ao ultrapassar o ponto máximo para esta troca, o projeto torna-se incontrolável. Por outro lado, é possível que os administradores simulem outras situações, com a intenção de verificar o tempo, o esforço e o custo do projeto.

Dizaji e Khalilpour (2014) descrevem a equação usada neste modelo, da seguinte forma:

$$S = E * Effort^{\frac{1}{3}} * td^{\frac{4}{3}} \quad (3)$$

Em que:

- S: tamanho do software em LOC;

- E: fatores do meio ambiente;
    - E = 2.000 (ambiente de desenvolvimento pobre);
    - E = 8.000 (métodos em prática, documentação e revisões adequadas);
    - E = 11.000 (ambiente ótimo, ferramentas e técnicas automatizadas);
- *Effort*: é o esforço aplicado (pessoa-ano);
- td: tempo de desenvolvimento (anos).

A equação empregada pela estimativa de Putnam, apresenta como resultado uma função, permitindo que os administradores definam valores para determinar a curva (Rayleigh-Norden – Tempo x Esforço), a fim de estabelecer qual será o esforço e o tempo necessários para que o desenvolvimento do software seja concretizado (HAUFE, 2001).

De acordo com Shekhar e Kumar (2016), a vantagem do modelo de Putnam, é que ele se baseia, essencialmente em apenas duas variáveis, que são tempo e tamanho, o que torna a aquisição de dados mais fácil para a construção de estimativas. Entretanto, esta característica também pode ser compreendida como sua desvantagem, uma vez que ele não leva em consideração outros aspectos do ciclo de vida do desenvolvimento de software.

5.3 Análise de Pontos de Função

A Análise de Pontos de Função (APF) foi desenvolvida por Allan Albrecht em 1979, como uma forma de quantificar as funcionalidades em projetos de software (SHEKHAR; KUMAR, 2016). Este método mede o tamanho do software, levando em consideração arquivos lógicos internos, arquivos de interface externa, entrada-saída externa e consultas externas a partir da métrica do ponto de vista funcional (MUSTAFA et al., 2005).

Ponto de função é a unidade de medida empregada nesta técnica, que possui o objetivo de contribuir com a medição do software, independentemente das tecnologias a serem utilizadas para sua construção, ou seja, a APF busca medir o que o software faz, sem tratar da forma como ele foi construído (VAZQUEZ et al., 2013). Segundo Mustafa et al. (2005), existem 5 componentes principais da análise de pontos de função que capturam a funcionalidade do aplicativo, que são: Entradas Externas, Saídas Externas, Consultas Externas, Arquivos Lógicos Internos e Arquivos de Interface Externa. Os três primeiros são tratados como tipos de funções transacionais e os dois últimos são compreendidos como tipos de função de dados.

A seguir, o Quadro 1 descreve os principais componentes da análise de pontos de função.

Quadro 1 – Principais componentes da Análise de Pontos de Função

COMPONENTES	DESCRIÇÃO
Entradas Externas (EE)	Funções que movem dados para o aplicativo sem apresentar manipulação de dados.
Saídas Externas (SE)	Funções que movem dados para o usuário e apresentam algumas manipulações de dados.
Consultas Externas (CE)	Funções que movem dados para o usuário sem apresentar manipulação de dados.
Arquivos Lógicos Internos (ALI)	A lógica sob a forma de dados fixos gerenciados pelo aplicativo através da utilização de Entrada Externa (EI).
Arquivos de Interface Externa (AIE)	A lógica na forma de dados fixos utilizados pela aplicação, mas que não foram executados nela.

Fonte: Elaborado pelos autores a partir de Berg et al. (2005)

De acordo com Vazquez et al. (2013), os dados para a contagem por pontos de função, são coletados a partir de casos de uso, onde apenas os requisitos funcionais são analisados para a medição, sendo que essa dimensão recebe o nome de tamanho funcional. Caso os requisitos não sejam levantados de forma clara e assertiva, isso implicará negativamente na contagem dos pontos funcionais do sistema a ser desenvolvido.

É possível destacar vários benefícios da aplicação da APF nas organizações, tais como: possibilita a contagem de todas as funções incluídas e que correspondem especificamente aos seus requisitos; análise de produtividade e qualidade; gerenciamento de escopo; gerenciamento dos requisitos; permite estimar custo e recursos; possibilita que a negociação de contratos seja fundamentada; além da normalização para a comparação de software (VAZQUEZ et al., 2013).

Shekhar e Khumar (2016) apontam algumas das vantagens da utilização deste método, tais como: não depende da linguagem de programação, ferramentas e métodos de implementação; os custos de desenvolvimento podem ser estimados em fases iniciais do desenvolvimento de software; os resultados são mais fáceis de serem tratados do que se fossem em LOC.

Por outro lado, a aplicação da análise requer um trabalho manual, o que acaba consumindo mais tempo. Além disso, é difícil para um novo desenvolvedor estimar o tamanho do software, pois o uso do ponto de função exige experiência (SHEKHAR; KUMAR, 2016).

6 CONCLUSÃO

A partir de uma análise do contexto de projetos de software, é evidente que estes projetos exigem um grande envolvimento e uma relativa maturidade de planejamento e gerenciamento dos ativos, a fim de alcançar as metas e os objetivos previamente estabelecidos pelos gestores. Entretanto, o gerenciamento do ciclo de vida de um projeto de software, pode ser complementado com a aplicação de técnicas algorítmicas capazes de proporcionar estimativas valiosas e disponibilizar dados importantes tanto do ponto de vista operacional quanto do ponto de vista gerencial do projeto.

Apesar do foco do presente trabalho estar na análise das contribuições para o gerenciamento de custos, é importante ressaltar que a aplicação de técnicas como o método COCOMO, o modelo de Putnam e a Análise de Pontos de Função, colaboram com o gerenciamento de tempo e escopo em projetos de software. Realizar a análise de fatores como: tamanho da aplicação (KLOC), recursos necessários, parâmetros e indicadores de tempo e esforço, representa uma tarefa fundamental para que o projeto atinja os resultados esperados sem exceder as metas previamente estabelecidas, pois é perceptível que o custo, o prazo e o escopo do projeto, estão diretamente alinhados, e que se um deles for comprometido, os demais serão fortemente influenciados.

Dessa forma, o gerenciamento de custos representa uma grande preocupação para as organizações e, consequentemente para os gerentes de projetos, pois o cumprimento das especificações econômicas só será alcançado através do controle e monitoramento de diversos fatores ligados aos elementos internos ou externos à organização. Sendo assim, o gerenciamento de custos pode ser compreendido como uma preocupação de alto nível (gerencial), mas é evidente que, para obter sucesso em um projeto de software, o nível operacional exerce uma grande influência e exige a criação de planos de contingência capazes de minimizar, controlar e evitar os riscos e incertezas do projeto.

Por meio de uma análise das diferentes atividades realizadas por profissionais de Tecnologia da Informação no desenvolvimento de software, tais como: documentação, programação, análise, testes, verificação e validação, é possível compreender que estas práticas devem estar alinhadas entre si, a fim de contribuir com a evolução do produto como um todo. Entretanto, o sucesso no projeto não será alcançado apenas com a simples realização destas etapas e com a consolidação do produto final, pois é necessário que o projeto atinja a eficiência, eficácia e efetividade necessárias para não ultrapassar o prazo e o custo estabelecidos pelo nível

gerencial, o que justifica a importância da aplicação de técnicas algorítmicas para a estimativa de custo, pois através da quantificação do esforço necessário e da análise das funcionalidades a serem implementadas, o nível de maturidade da equipe de desenvolvimento pode ser otimizado e gerenciado através de indicadores valiosos, que devem ser utilizados para contribuir com os processos do gerenciamento de custos do projeto.

Como sugestão para trabalhos futuros, é possível indicar a realização de estudos experimentais no âmbito de projetos de software, propondo a análise e a comparação entre técnicas algorítmicas de estimativa de software, com o intuito de contribuir com o gerenciamento não apenas de custo, mas de tempo e escopo em projetos.

REFERÊNCIAS

BARDSIRI, V. K.; JAWAWI, D. N. A.; HASHIM, S. Z. M.; KHATIBI, E. Increasing the accuracy of software development effort estimation using projects clustering. **IET Software**, v. 6, n. 6, p. 461-473, 2012. Disponível em: <http://ieeexplore.ieee.org/document/6353331/>. Acesso em: 19 ago. 2017.

BERG, K. V. D.; DEKKERS, T.; OUDSHOORN, R. Functional size measurement applied to UML-base user requirements. In: SOFTWARE MEASUREMENT EUROPEAN FORUM, 2.,. **Proceedings of the 2nd Software Measurement European Forum**. Rome: SMEF, 2005. Disponível em: <https://research.utwente.nl/en/publications/functional-size-measurement-applied-to-uml-based-user-requirement>. Acesso em: 27 ago. 2017.

DESPA, M. L. Comparative study on software development methodologies. **Database Systems Journal**, v. 5, n. 3, p. 37-56, 2014. Disponível em: <http://dbjournal.ro/archive/17/17_4.pdf>. Acesso em: 22 ago. 2017.

DIZAJI, Z. A.; KHALILPOUR, K. Particle swarm optimization and chaos theory based approach for software cost estimation. **International Journal of Academic Research**, v. 6, n. 3, p. 130-135, 2014.

GIL, A. C. **Métodos e técnicas de pesquisa social**. São Paulo: Atlas, 1999.

HANSEN, D. R.; MOWEN, M. M. **Cost management:** accounting and control. 6. ed. Mason, OH: South-Western College, 2006.

HAUFE, M. I. **Estimativa da produtividade no desenvolvimento de software**. 2001. 108 f. Dissertação de Mestrado. UFRGS – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001. Disponível em: <http://www.lume.ufrgs.br/handle/10183/1651>. Acesso em: 17 ago. 2017.

HEEMSTRA, F. J. Software Cost Estimation Models, University of Technology Department of Industrial Engineering. In: IEEE Information Technology, 1990. 'Next Decade in

Information Technology. **Proceedings...** Jerusalem Conference on (Cat. No.90TH0326-9), 5., Jerusalem: IEEE, 1990. Disponível em: <http://ieeexplore.ieee.org/document/128297/>. Acesso em: 20 ago. 2017.

IACOB, V. S. Risk management and evaluation and qualitative method within the projects. **Ecofurum Journal**, v. 3, n. 1, 2014

KAPLAN, R. S.; COOPER, R. **Custo e desempenho: administre seus custos para ser mais competitivo**. São Paulo: Futura, 1998.

KARADSHEH, L.; ALHAWARI, S.; TALET, A.N. The support of knowledge process to enhance risk analysis in Jordanian telecommunication companies. **Journal of Information & Knowledge Management**, v. 11, n. 2, 2012.

KUJALA, J.; BRADY, T.; PUTILA, J. Challenges of cost management in complex projects. **International Journal of Business and Management**, v. 9, n. 11, p. 48-58, 2014. Disponível em: <http://www.ccsenet.org/journal/index.php/ijbm/article/view/39454/22715>. Acesso em: 14 ago. 2017.

MOREIRA, V. Revisão de literatura e desenvolvimento científico: conceitos e estratégias para confecção. **Janus**, v. 1, n. 1, p. 19-30, 2004. Disponível em: <http://publicacoes.fatea.br/index.php/janus/article/view/1/1>. Acesso em: 28 jul. 2017.

MUNNS, A. K.; BJEIRMI, B. F. The role of project management in achieving project success. **International Journal of Project Management**, v. 14, n. 2, p. 81-87, 1996. Disponível em: <http://www.sciencedirect.com/science/article/pii/0263786395000577>. Acesso em: 02 ago. 2017.

MUSTAFA, K.; GOWTHAMAN, K.; KHAN, R. A. Measuring the function points for migration project: a case study. **American Journal of Applied Science**, v. 2, n. 8, p. 1218-1221, 2005. Disponível em: <http://thescipub.com/PDF/ajassp.2005.1218.1221.pdf>. Acesso em: 28 jul. 2017.

PATAH, L. A. **Avaliação da relação do uso de métodos e treinamentos em gerenciamento de projetos no sucesso dos projetos através de uma perspectiva contingencial**: uma análise quantitativa. 2010. 234 f. Tese de Doutorado. USP – Universidade de São Paulo, São Paulo, 2010. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3136/tde-17082010-123256/pt-br.php>. Acesso em: 24 ago. 2017.

PILLAI, K.; NAIR, V. S. S. A model for software development effort and cost estimation. **IEEE Transactions on Software Engineering**, v. 23, n. 8, p. 485-497, 1997. Disponível em: <https://dl.acm.org/citation.cfm?id=265174>. Acesso em: 02 ago. 2017.

PINTO, J. K.; SLEVIN, D. P. Critical success factors across the project life cycle. **Project Management Journal**, v. 19, n. 3, p. 67-75, 1988.

PORTER, M. E. **Estratégia Competitiva:** técnicas para análise de indústrias e da concorrência. Rio de Janeiro: Campus, 2004.

PMI. **Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK ®)**. 5. ed. Project Management Institute, 2013.

RAMACHARAN, S.; RAO, K. V. G. Parametric models for effort estimation for global software development. **Lecture Notes on Software Engineering**, v. 1, n. 2, p. 178-182, 2013. Disponível em: <http://www.lnse.org/papers/40-IE1011.pdf>. Acesso em: 30 jul. 2017.

SCHOONWINKEL, S.; FOURIE, C. J.; CONRADIE, P. D. F. A risk and cost management analysis for changes during the construction phase of a Project. **Journal of the South African Institution of Civil Engineering**, v. 58, n. 4, p. 21-28, 2016. Disponível em: <http://www.scielo.org.za/pdf/jsaice/v58n4/03.pdf>. Acesso em: 26 jul. 2017.

SHEKHAR, S.; KUMAR, U. Review of Various Software Cost Estimation Techniques. **International Journal of Computer Applications**, v. 141, n. 11, p. 31-44, 2016. Disponível em: <http://www.ijcaonline.org/archives/volume141/number11/shekhar-2016-ijca-909867.pdf>. Acesso em: 12 jul. 2017.

SHENRAR, A.; DVIR, D. **Reinventando o gerenciamento de projetos**: a abordagem diamante ao crescimento e inovação bem-sucedidos. São Paulo: M. Books, 2010.

SINGH, B. K.; TIWARI, S.; MISHRA, K. K.; MISRA, A. K. Tuning of cost drivers by significance occurrences and their calibration with novel software effort estimation method. **Advances in Software Engineering**, v. 2013, n. 1, p. 1-10, 2013.

TICHACEK, R. L. Effective cost management - Back to basics. **Cost Engineering**, v. 48, n. 3, p. 27-33, 2006.

TIFFANY, P.; PETERSON, S. D. **Planejamento estratégico**: o melhor roteiro para um planejamento estratégico eficaz. Rio de Janeiro: Campus, 1998.

TURNER, J. R.; SIMISTER, S. J. **Gower handbook of project management**. 3. ed. Aldershot, HA: Gower, 2000.

VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. **Análise de pontos de função**: medição, estimativas e gerenciamento de projetos de software. 13. ed. São Paulo: Érica, 2013.

VENKATARAMAN, R. R.; PINTO, J. K. **Cost and value management in projects**. Hoboken, NJ: John Wiley & Sons, Inc., 2008.