

# DEFINITION OF A PROCESSES MODEL FOR SOFTWARE DEVELOPMENT PROJECTS IN A PUBLIC COMPANY.

## DEFINIÇÃO DE UM MODELO DE PROCESSOS PARA PROJETOS DE DESENVOLVIMENTO DE SOFTWARES EM UMA EMPRESA PÚBLICA

Caetano B. Pedroso<sup>1✉</sup>, Ana C. C. Coimbra<sup>2</sup>, Francisco V. Maracci<sup>1</sup>

<sup>1</sup>Universidade do Oeste Paulista (UNOESTE), Presidente Prudente, São Paulo, Brasil

<sup>2</sup>Etec Professor Adolpho Arruda Mello, Presidente Prudente, São Paulo, Brasil

✉ [pedroso@unoeste.br](mailto:pedroso@unoeste.br)

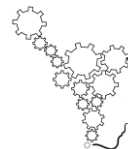
Recebido: 25 maio 2018 / Aceito: 16 junho 2018 / Publicado: 20 junho 2018

**ABSTRACT.** The development of software is a complex activity that has well defined steps to obtain a good product. The objective of this study was to identify how software is developed in a public company and propose improvements in the development process. Analyses of the existing development process and bibliographical research were done, in order to allow a processes model for the management of software development projects to be proposed. With a well-defined process model managing the development projects there is a decrease in problems and an increase in the satisfaction with the final product.

**Keywords:** project management, software development, communication and information technology, quality control.

**RESUMO.** O desenvolvimento de softwares é uma atividade complexa que possui etapas bem definidas para a obtenção de um bom produto. O objetivo deste estudo foi identificar a forma como eram desenvolvidos os softwares em uma empresa pública e propor melhorias no processo. Foram feitas análises dos processos de desenvolvimento existentes e pesquisas bibliográficas, de forma a permitir que um modelo de processos para a gerência de projetos de desenvolvimento de software fosse proposto. Com um bom modelo de processos gerindo os projetos de desenvolvimento os problemas tendem a diminuir e a satisfação com o produto final aumenta.

**Palavras-chave:** gestão de projetos, desenvolvimento de software, tecnologia da informação e comunicação, controle de qualidade.



## 1. INTRODUÇÃO

Uma das mais importantes áreas das empresas modernas é a área da Tecnologia da Informação (TI), responsável pela implantação e gestão de todas as Tecnologias de Informação e Comunicação (TIC) usadas por elas. Assim, podemos observar que praticamente todo o planejamento estratégico das empresas é sustentado pela área de TI, bem como a gestão das ferramentas que apoiam suas operações diárias e rotineiras.

Neste contexto, uma boa escolha de quais soluções de software usar é de suma importância para o sucesso das empresas, sejam elas das esferas públicas ou privadas. No caso das empresas públicas esse é um ponto ainda mais importante, pois se analisarmos a administração pública pela ótica social, ela é concebida como uma organização que fornece, direta ou indiretamente, serviços aos cidadãos (LAURINDO, 2008). Assim, a utilização das TIC é vista como fomento para um fornecimento mais eficaz e eficiente desses serviços.

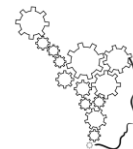
Uma boa solução de TIC possibilita, ao gestor, o conhecimento dos contextos e variáveis que envolvem todos os processos de tomada de decisões, e, assim, melhorar a qualidade delas e sua probabilidade de sucesso. Em suma, favorece uma melhor gestão do funcionamento da empresa e permite um melhor planejamento de suas atividades futuras.

Para atender essas características, muitas empresas optam por utilizar soluções de TIC desenvolvidas por terceiros, independentemente do tipo de licença de uso que apresentam (proprietária ou aberta). Contudo, existem as que optam pelo desenvolvimento de suas próprias soluções de TIC, de forma a obter soluções mais personalizadas e que atendam, de uma forma mais precisa, suas particularidades individuais.

No caso daquelas que optam por desenvolver suas próprias soluções, uma maior atenção deve ser dada ao processo de produção e manutenção de tais soluções, pois, elas devem refletir a real necessidade do negócio e, devem estar preparadas a permitir sua evolução conforme as necessidades do negócio mudem ou evoluam.

A mudança é inevitável em todos os projetos de software, independente do seu tamanho. Os requisitos dos sistemas mudam de forma simultânea às mudanças sofridas pelo negócio, pois as necessidades e requisitos organizacionais se alteram durante sua vida útil (SOMMERVILLE, 2011), além disso, eventuais defeitos precisam ser reparados.

Estes softwares são usados como parte das operações rotineiras da empresa, ou seja, nas atividades diárias e, portanto, são responsáveis pela operacionalização de seus processos.



Assim, eles apoiam ou executam as operações e atividades da empresa, de forma a propiciar um fluxo eletrônico, automático e preciso, das informações entre as suas diversas funções e processos. Nesse contexto, um processo pode ser definido como um conjunto de atividades que se desdobram em um conjunto de operações (CARVALHO, 2015).

É importante apontar que, apesar da solução de software desenvolvida ser usada nas operações diárias da empresa, seu desenvolvimento não deve ser considerado uma atividade rotineira. Assim, cada novo desenvolvimento de uma solução de software, ou uma alteração em uma já existente, deve ser considerado um projeto.

Segundo Carvalho (2015), um projeto pode ser entendido como um processo único, que é formado por atividades e operações coordenadas e controladas, devendo possuir um início e término definido e que visa alcançar um único objetivo.

As principais características que diferenciam um processo de um projeto são a temporalidade, a individualidade do produto desenvolvido pelo projeto, a complexidade e a incerteza (VARGAS, 2014).

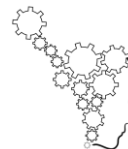
A temporalidade indica que todo projeto apresenta um início e um fim definidos, ou seja, ao contrário dos processos das empresas, que são executados rotineiramente, ele é um evento de duração finita e com um objetivo bem definido.

Esse objetivo bem definido indica a característica de individualidade. Ou seja, um projeto deve apresentar um resultado novo. Assim, cada projeto gera um produto novo e único. No caso do desenvolvimento de softwares, esse produto pode ser um novo software ou a adição, modificação ou remoção de alguma funcionalidade não existente.

Um projeto não é uma atividade simples, ele é formado por várias outras atividades interdependentes e com propósitos definidos, seguindo uma clara, e lógica, sequência de eventos. Ele utiliza recursos, que devem ser previamente alocados a cada uma de suas atividades. Ele também depende de pessoas, que atuarão como partes interessadas nele, ou executando e controlando suas atividades.

A incerteza se dá devido aos muitos problemas que um projeto pode sofrer durante sua execução. Assim, esta característica é a que indica a necessidade de uma boa gestão do projeto, de forma a mitigar ou eliminar os pontos de problemas.

Neste contexto, os principais benefícios da utilização do gerenciamento de projeto, segundo Vargas (2014), são: evitar surpresas durante a execução; antecipar situações



desfavoráveis que poderão ser encontradas; adaptar os trabalhos ao cliente; disponibilizar orçamentos antes do início dos gastos; agilizar as decisões; aumentar o controle gerencial; facilitar e orientar as revisões da estrutura do projeto; otimizar alocação de pessoas, equipamentos e materiais; e, gerar documentação para projetos futuros.

Como todo projeto pode ser dividido num ciclo de atividades bem definidos, ele oferece um melhor controle de recursos, gastos e das metas que devem ser atingidas. Esse ciclo de vida permite a identificação de todas as etapas, a relação entre elas e quais entregáveis elas geram, de forma a garantir uma visão geral de todo o projeto. Este ciclo, de acordo com Vargas (2014), contempla as fases de: iniciação, planejamento, execução, monitoramento e controle, e, encerramento.

O mapeamento dos processos pode ser feito usando-se a Notação de Modelagem de Processos e Negócios (BPMN – Business Process Model and Notation). A função principal do BPMN é prover uma notação que é prontamente entendida por analistas de negócio e pelos desenvolvedores que implementarão os processos (OMG, 2011).

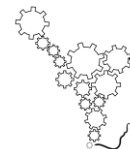
Este estudo analisou uma empresa pública, que desenvolve suas próprias soluções de software de forma a identificar como eram seus processos de desenvolvimentos. O modelo de desenvolvimento encontrado não mantinha uma clara definição formal de como o processo geral era feito. Não havia um detalhamento de início e fim do projeto, nem alocação de recursos e pessoas bem definidos. A documentação era implícita no software e dependente do analista que o desenvolveu.

## **2. METODOLOGIA**

Para esse trabalho foi feita uma análise da forma como são desenvolvidos os softwares usados nas operações diárias de uma empresa pública, de forma a verificar se não havia espaço para melhorias ou identificar a existência de problemas. Foram considerados o desenvolvimento de novos softwares e suas alterações futuras.

Em paralelo, foi feita uma pesquisa bibliográfica sobre gestão de processos e projetos de forma a identificar as melhores práticas que podem ser adotadas neste contexto.

Após ambas as análises, foi proposto um modelo de processos que permita o gerenciamento de projetos de desenvolvimento de novas soluções de software, e que, posteriormente, também permita o gerenciamento do desenvolvimento de novas funcionalidades e correções nas que já existam. Além disso, foi proposto um conjunto de



documentos para a documentação e controle de todo o desenvolvimento do projeto, e servir como referência para projetos futuros.

### 3. RESULTADOS

De forma a melhorar a atividade de desenvolvimento de software desta empresa, foi definido um modelo de processo para a gestão destes projetos, sejam eles para novos softwares ou para alterações em existentes.

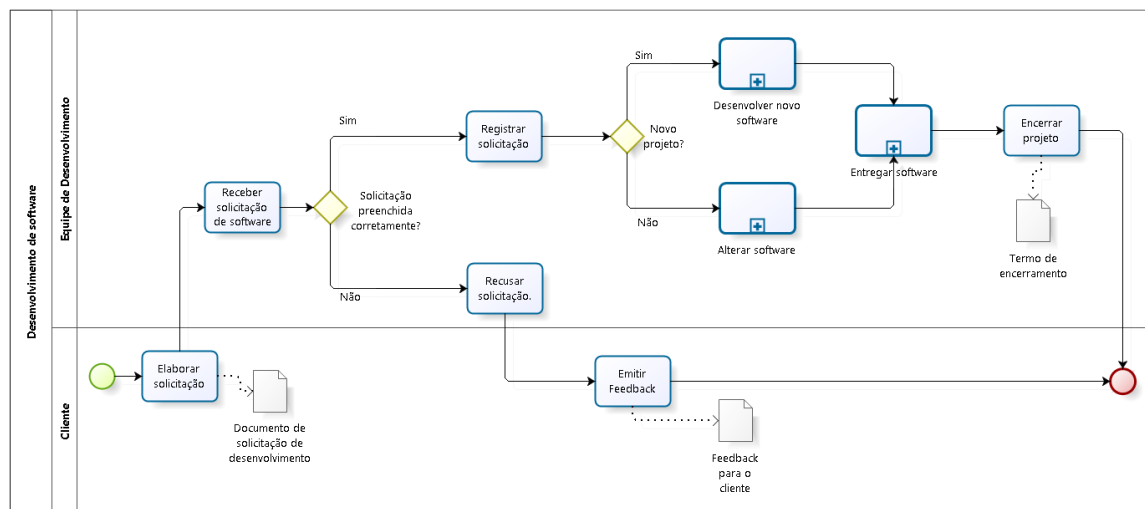
Um dos principais problemas detectados é que não havia uma solicitação formal de início de projeto por parte do cliente, não havia uma documentação padronizada da solicitação, nem das obrigações dos clientes e desenvolvedores. Isso dificultava o acompanhamento do projeto, e, por sua vez, a validação final por parte do cliente.

Neste contexto, o modelo proposto exige que o cliente faça uma solicitação formal do desenvolvimento de um novo software, ou de modificações em um existente, através de um documento padronizado. Este documento é, então, analisado pela equipe de desenvolvimento para identificar possíveis falhas na solicitação, e, somente se ela estiver correta, a fase de iniciação do projeto tem início. Em caso de falhas, a solicitação é rejeitada e um feedback do problema é devolvido ao cliente, encerrando o processo. Todo esse fluxo é demonstrado na Figura 1.

No caso de aceite, a solicitação é registrada e é dado procedimento na abertura do projeto. Doravante temos duas possibilidades de continuidade, abertura de projeto para: novo software (Figura 2) ou para alteração em um já existente (Figura 3). Qual seguir depende exclusivamente da solicitação do cliente.

O início do projeto se dá com a criação de um documento de abertura de projeto, também padronizado, autorizando formalmente sua abertura e execução. Ele contempla a análise de requisitos inicial, de acordo com as necessidades expostas pela solicitação original.

FIGURA 1 – PROCESSOS PARA UM PROJETO DE DESENVOLVIMENTO DE SOFTWARES

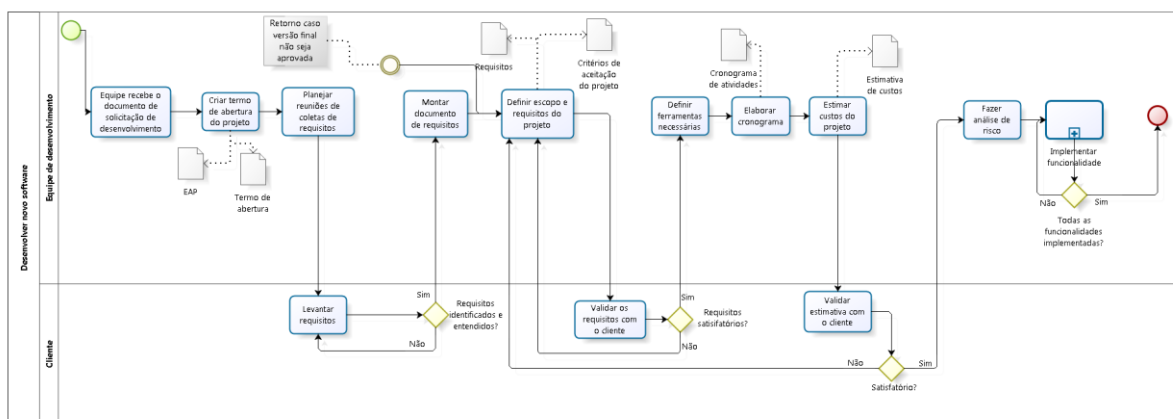


FONTE: O autor (2018)

Neste momento também é criado a Estrutura Analítica do Projeto (EAP). A EAP é a base principal do escopo do projeto, retratando sua divisão em partes (CARVALHO, 2015). Essa ferramenta, ainda segundo o autor, é uma das mais importantes do gerenciamento de projetos. Ela é frequentemente usada para elaborar, além de confirmar, o entendimento de todos os envolvidos sobre o escopo do projeto, servindo, também como estrutura de controle para, se necessário, efetuar mudanças no escopo. Neste contexto, a EAP é construída a partir de desdobramentos do escopo até o nível de atividades.

Uma vez entendidos todos os desdobramentos do projeto, inicia-se a fase de análise de requisitos do software, ou seja, a fase de planejamento. Os requisitos são levantados a partir de reuniões com o cliente, até que seja esgotada a identificação e entendimento de novos requisitos.

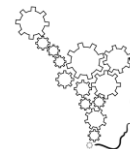
FIGURA 2 – DESENVOLVER NOVO SOFTWARE



FONTE: O autor (2018)



www.relainep.ufpr.br



Com os requisitos coletados é feito o confrontamento destes com o escopo do projeto e um documento final de requisitos é gerado. No desenvolvimento de um novo software essa transição entre etapas é direta, como vemos na Figura 2, já em um projeto de alteração de um software, antes do planejamento das reuniões e após o levantamento dos requisitos é feita uma análise do software existente, de forma a identificar todos os pontos de alteração, somente depois é feita a geração do documento de requisitos final, isso é observado na Figura 3.

Gerado este documento, os requisitos devem ser validados junto ao cliente para confirmar se eles realmente representam suas necessidades. É importante salientar que a validação dos requisitos pode exigir alterações no documento de requisitos, conforme inconsistências são identificadas. O ciclo de validação de requisitos termina quando ambas as partes chegam a um consenso sobre os requisitos do software.

Nesta fase também são definidos os critérios de aceitação do projeto por parte do cliente, ou seja, o que se espera ao final do projeto. Os critérios estão vinculados aos requisitos e, portanto são gerados durante suas validações.

Na sequência da fase de planejamento, temos a definição de quais ferramentas serão usadas no desenvolvimento. Novos projetos permitem uma maior liberdade de escolha, enquanto alterações podem exigir ferramentas já usadas quando da criação do software original.

Uma vez definidas as ferramentas, um cronograma de atividades é elaborado e registrado. Esse cronograma norteará toda a execução do projeto, permitindo seu acompanhamento e controle, além do gerenciamento de tempo.

Neste contexto, o gerenciamento do tempo de um projeto é composto por processos que permitam o cumprimento dos prazos de acordo com o planejado (CARVALHO, 2015), dentre eles o gerenciamento do cronograma, a definição da sequência de atividades a serem executadas, bem como a estimativa do seu tempo de duração. Segundo Vargas (2014), o principal objeto da gerência de tempo é garantir que o projeto seja concluído dentro do prazo determinado.

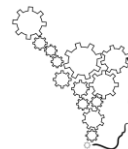
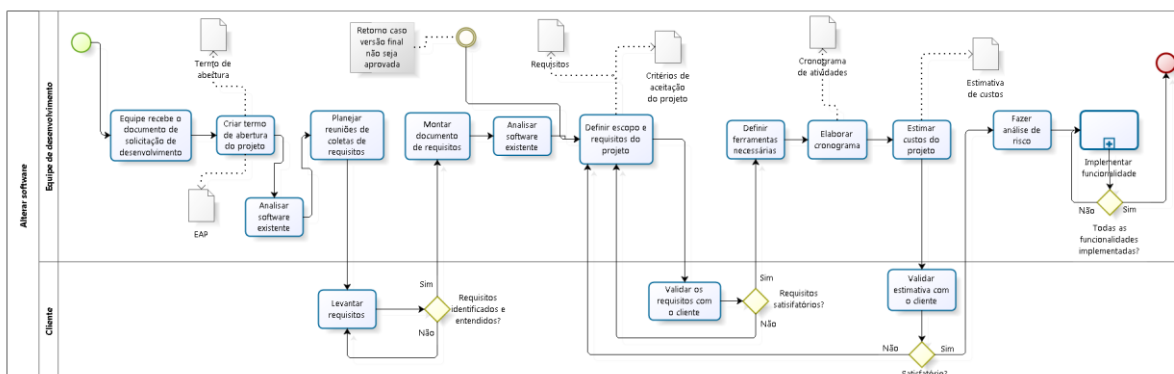


FIGURA 3 – ALTERAR SOFTWARE



FONTE: O autor (2018)

Powered by  
bizagi

O planejamento se encerra com a elaboração do documento com as estimativas de custos do projeto. Essa estimativa deve ser validada pelo cliente, e em caso de aceite a fase de execução do projeto é iniciada. Em caso de recusa pelo cliente, deve ser retornado a fase de definição dos requisitos, para que adequações sejam feitas de forma a alinhar os custos estimados com o desejado pelo cliente.

É importante salientar que, quando for uma solução de mudanças uma análise mais detalhada pode ser necessária, pois, por exemplo, a solicitação pode indicar a necessidade de correção de um defeito que já foi previamente resolvido. Ou seja, nem toda solicitação de mudanças requer ação (SOMMERVILLE, 2011). Outro ponto importante é que os custos da alteração podem ser mais altos do que os benefícios adquiridos caso seja implantada.

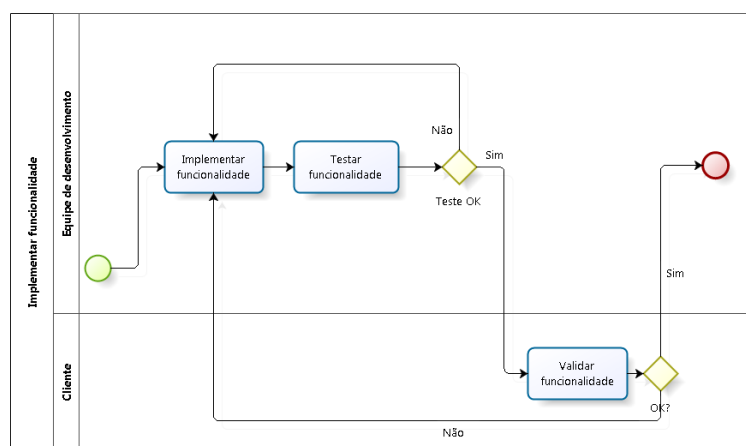
Antes do início da fase de execução é necessário efetuar uma análise de risco para o desenvolvimento do projeto. Essa etapa é importante para garantir que o desenvolvimento ocorra sem maiores problemas, principalmente quando é um projeto de alteração em um software existente, pois as alterações podem inviabilizar o funcionamento do software existente. Segundo Vargas (2014), o gerenciamento de riscos de um projeto é realizado com base nos riscos previamente identificados, por exemplo, através da EAP definida.

A execução propriamente dita é feita através do subprocesso implementar funcionalidade, ilustrado na Figura 4. Aqui, as funcionalidades são implementadas, uma a uma, verificadas e validadas com testes funcionais e junto ao cliente. Caso a validação de uma funcionalidade não ocorra, ela é reimplementada, reiniciando o ciclo. Uma funcionalidade é considerada implementada somente após seu teste de aceitação ser aprovado pelo cliente. É importante salientar que o documento de requisitos é de suma importância nessa fase, pois ele norteia a implementação das funcionalidades.



O teste de aceitação, vista na Figura 4, é uma das formas de se efetuar o monitoramento e controle do desenvolvimento do projeto. Além dela também são usados o documento de EAP e o cronograma proposto. Assim é possível acompanhar todo o desenvolvimento do projeto, ajudando a eliminar problemas que possam atrasá-lo ou atrapalhá-lo.

FIGURA 4 – IMPLEMENTAR FUNCIONALIDADE

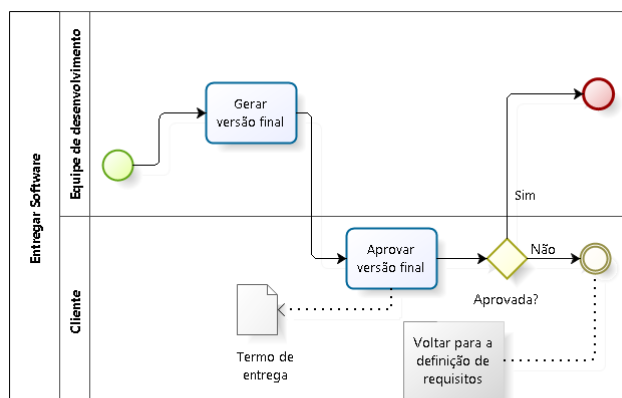


Powered by  
bizagi  
Modeler

FONTE: O autor (2018)

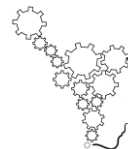
Ao término da fase de execução temos a entrega do software, ilustrada na Figura 5. Nesse subprocesso é gerada uma versão final do software que é, então, submetido para a aprovação do cliente. Com o aceite do cliente o desenvolvimento é considerado concluído e o projeto segue para a fase de encerramento. Nesse momento, o cliente registra um termo de entrega, documento padronizado, reconhecendo o final satisfatório do desenvolvimento do projeto como um todo.

FIGURA 5 – ENTREGAR SOFTWARE



Powered by  
bizagi  
Modeler

FONTE: O autor (2018)



Como vemos na Figura 5, caso a versão final não seja aceita pelo cliente, deve-se retornar a fase de definição do escopo e requisitos do projeto, seja no caso de novo software (Figura 2), ou alteração de software existente (Figura 3). Assim, os motivos da recusa devem ser analisados e os requisitos devem ser readequados para refletir o desejo do cliente, a partir deste ponto, o fluxo de processos segue normalmente com já explicado até que uma nova versão esteja pronta para validação do cliente.

O projeto se encerra com a emissão do termo de encerramento, como mostrado na Figura 1. Nessa etapa, os recursos que foram alocados para o projeto devem ser liberados, e as lições aprendidas devem ser registradas para uso em projetos futuros.

#### **4. CONCLUSÃO**

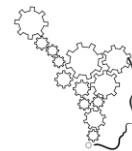
A atividade de desenvolvimento de softwares é uma que demanda bastante criatividade. Apesar dessa característica permitir a criação de novos e revolucionários produtos, ela pode impactar negativamente no processo de desenvolvimento, principalmente quando ele é feito em equipe.

A padronização das atividades e documentos em um projeto de desenvolvimento de software permite uma melhor gerência de todo o projeto e evita que ele atrase, falhe ou ultrapasse o orçamento existente. Contudo, isso, em momento algum, impacta negativamente na parte criativa do processo.

A documentação formal das várias etapas do projeto, como proposto, vem a auxiliar no sucesso do projeto, pois permite identificar, de forma antecipada, as etapas e as necessidades de cada uma delas, além dos problemas que podem surgir durante a execução do projeto.

De posse dessa documentação é possível que os envolvidos no projeto antevêm os problemas futuros, tomando precauções para que eles não afetem o projeto, ou, ao menos, diminuindo os riscos associados a eles. Permite que definam previamente todas as etapas e atividades do projeto, bem como seu cronograma, seus custos e quais os profissionais que precisam estar envolvidos nele.

Devido a isso, o controle da evolução do projeto é facilitado, bem como quais as responsabilidades de cada um dos envolvidos. Outro ponto a ser considerado é que, por envolver o cliente como uma personagem ativa durante a execução do projeto, as atividades, quando concluídas, já são validadas e confirmadas. Isso permite a identificação de



inconsistências pontuais, além de fornecer uma visão geral da evolução do projeto, ajudando a eliminar recusas ao término dele.

A documentação gerada permite a equipe de desenvolvimento a criação de uma biblioteca de projetos, com os acertos e erros cometidos nos já executados. Consultas a essa biblioteca ajuda no desenvolvimento de futuros projetos, já que problemas, dificuldades e acertos prévios ajudam a definir quais etapas, atividades, recursos, etc. são necessários.

Não é contemplado nesse trabalho a análise de softwares previamente desenvolvidos, e, portanto, com deficiências em suas documentações. O modelo proposto considera somente novos projetos e suas futuras alterações.

Como trabalhos futuros, podemos propor a validação do modelo proposto em ambiente de produção e, também, a implementação de um modelo que permita o gerenciamento de projetos de manutenção em sistemas de software legados que não possuem a documentação necessária para seu devido acompanhamento.

## REFERÊNCIAS

- CARVALHO, F. C. A. **Gestão de projetos**. São Paulo: Pearson Education do Brasil, 2015.
- LAURINDO, F. J. B. **Tecnologia da informação: planejamento e gestão de estratégias**. São Paulo: Atlas, 2008
- OMG. **Business Process Model and Notation (BPMN)**. Version 2.0, formal/2011-01-03, 2011. Disponível em <<http://www.omg.org/spec/BPMN/2.0/PDF>>. Acesso em 24 maio 2018.
- SOMMERVILLE, I. **Engenharia de Software**. 9ed. São Paulo: Pearson Prentice Hall, 2014.
- VARGAS, R. V. **Manual prático do plano de projeto: utilizando o PMBOK® Guide**. 5ed. Rio de Janeiro: Brasport, 2014.