PROCESSANDO REQUISITOS: O USO DE PROCESSOS NA ESPECIFICAÇÃO DE REQUISITOS

Alessandra Cristina Andrade
IESB – Instituto de Educação Superior de Brasília

RESUMO

Este trabalho apresenta as situações enfrentadas pelos analistas de requisitos e usuários de sistemas de software, bem como alguns métodos como o RUP que vem sendo comumente utilizados para realizar o levantamento das necessidades do cliente. Apresenta também os problemas encontrados durante este processo, de ambos os pontos de vista, além de uma breve descrição dos métodos utilizados e os problemas pontuais que cada passo deste processo traz.
Como proposta em um segundo momento apresenta uma diretriz para unificar o que há de vantajoso na disciplina de requisitos do RUP e os benefícios do Mapeamento de Processos. Uma junção de metodologias, retirando passos e documentos que tem atrapalhado e gerado problemas utilizados pelo RUP e acrescentando artefatos provenientes de outras metodologias como o BPMN, a exemplo do mapeamento de processos. O foco é auxiliar o entendimento dos envolvidos no desenvolvimento do software, apresentando uma diretriz que viabilize o levantamento das necessidades do usuário com o menor impacto e retrabalho tanto para o cliente quanto para o profissional da área de desenvolvimento de software.

Palavras-chave: Mapeamento de processo. Levantamento de Requisitos, RUP, BPMN.

ABSTRACT

This paper presents the situations experienced by the requirements analysts and users of software systems, as well as some methods such as RUP, which has been used to obtain the customer's needs. It also presents the problems of these processes, from both points of view, plus a brief description of the methods used and the specific problems that each step of this process brings.
As proposed in a second stage presents a guideline to unify the best in the discipline of the RUP requirements and benefits of Process Mapping. A joint methodologies, removing steps and documents that have generated problems used by RUP and adding artifacts from other methodologies such as BPMN, like the mapping process. The focus is to assist the understanding stakeholder in software development, with a guideline which facilitates the collection of user needs with minimal impact and rework for both the customer and for the professional in the area of software development.

Keywords: *Mapping Process, RUP, BPMN.*

1 INTRODUÇÃO

Entre os vários motivos que poderiam levar a escolha do tema para este artigo, optou-se por escrever sobre algo que fosse objeto de vivência do dia a dia, mas que também fosse um problema vislumbrado academicamente e já discutido por alguns autores, com algumas soluções propostas para que pudessem ser experimentadas e testadas.

Dentre as literaturas consultadas para subsidiar este estudo foi possível verificar em PRESSMAN (2011, p. 126) o seguinte texto:

> "É o seu pior pesadelo. Um cliente entra no seu escritório, se senta, te olha diretamente nos olhos e diz: "Eu sei que você imagina que entendeu aquilo que eu lhe disse, mas o que você não compreende é que aquilo que eu lhe disse era exatamente aquilo que eu quis dizer". Invariavelmente, isso acontece no final do projeto, após compromissos de prazos de entrega terem sido estabelecidos, reputações estarem na mira e muito dinheiro estar em jogo."

Problemas com o levantamento de requisitos são vivenciados a todo o momento, seja ainda na documentação que por vezes se faz extensa, ou por outras se faz insuficiente, seja por não ser possível extrair as informações necessárias do cliente.

Ainda por PRESSMAN (2011, p. 127) é possível ler sobre três tipos de problemas na fase inicial do levantamento:

> "**Problemas de escopo.** Os limites do sistema são definidos de forma precária ou os clientes/usuários especificam detalhes técnicos desnecessários que podem confundir, em vez de esclarecer, os objetivos globais do sistema.
> **Problemas de entendimento.** Os clientes/usuários não estão completamente certos do que é preciso, têm um entendimento inadequado das capacidades e limitações de seus ambientes computacionais, não possuem um entendimento completo do domínio do problema, tem problemas para transmitir suas necessidades ao engenheiro de sistemas, omitem informações que acreditam ser "óbvias", especificam requisitos que conflitam com as necessidades de outros clientes/usuários ou especificam requisitos que são ambíguos ou impossíveis de ser testados.
> **Problemas de volatilidade.** Os requisitos mudam com o tempo. Para ajudar a superar esses problemas, devemos abordar o levantamento de requisitos de forma organizada."

Para resolver estes problemas várias soluções já foram apresentadas, como: diagramas propostos pela UML, artefatos da disciplina de requisitos do RUP, prototipação para ajudar a visualizar as funcionalidades do sistema. Atualmente se tem usado uma técnica que ume objeto de estudo deste artigo, o Mapeamento de Processo, uma metodologia adotada para auxiliar a tentativa de minimizar os impactos causados por um levantamento de requisitos ineficaz ou insuficiente.

A proposta deste estudo é unificar o que há de vantajoso na disciplina de requisitos do RUP e os benefícios do Mapeamento de Processos, apresentando uma diretriz que viabilize o levantamento das necessidades do usuário com o menor impacto e retrabalho tanto para o cliente quanto para o profissional da área de desenvolvimento de software.

Para obter os resultados esperados, o artigo está estruturado da seguinte forma: a seção 2 apresenta um breve conceito de Levantamento de Requisitos e algumas estatísticas de problemas no desenvolvimento de software relacionado ao levantamento das necessidades e problemas do usuário. Na seção 3 uma apresentação simplificada sobre mapeamento de processos e alguns pontos chaves fornecendo embasamento para a seção 4, onde o melhor das propostas apresentadas pelo RUP e pelo Mapeamento de processos será trabalhado para uma adequação que atenda melhor e minimize os problemas apresentados na seção 2. A seção 5

finalmente apresenta a conclusão do artigo, identificando as restrições encontradas no desenvolvimento deste artigo e os possíveis trabalhos futuros acerca do tema.

2 LEVANTAMENTO DE REQUISITOS

Ao pensar em desenvolvimento de software é necessário pensar em levantamento de requisitos, pensar em como serão extraídos dos clientes os problemas enfrentados, as necessidades atuais que ele possui para que se possa transformar tudo isto no sistema esperado.

Para resolver este problema o RUP oferece uma disciplina que percorre praticamente todas as suas fases em maiores ou menores proporções, como já é de conhecimento, a disciplina de Requisitos. De acordo com o que é definido pelo RUP: um requisito é uma condição ou um recurso com o qual um sistema deve estar em conformidade.

A disciplina de requisitos pode ser vista como uma fronteira entre as necessidades dos usuários e o sistema a ser construído. Como se consegue perceber em SHUJA e KREBS (2007, p. 93) esta disciplina consiste em atividades que garantam a engenharia de requisitos e uma gestão eficaz e descreve como criar uma visão e traduzi-lo em um modelo de casos de uso.

É nela, que as necessidades dos *stakeholders* (assim nomeados pelo RUP) são mapeadas, documentadas e gerenciadas, através dos artefatos e *templates* propostos que podem ser customizados de acordo com a complexidade do projeto.

A finalidade desta disciplina é:

- Manter um alinhamento entre os *stakeholders* sobre o que o sistema deve fazer;
- Fornecer aos desenvolvedores uma compreensão dos requisitos do sistema;
- Definir o escopo do sistema;
- Fornecer uma base para estimar o custo e o prazo de desenvolvimento do sistema;
- Definir a interface de usuário para o sistema.

Para atender a estas finalidades esta disciplina propões alguns produtos de trabalho como resultado a ser gerado, são exemplos destes produtos de trabalho que serão analisados no contexto deste artigo. Em SHUJA e KREBS (2007, p. 101 e 102) é possível entender o conceito destes artefatos:

- Documento de visão do projeto: prevê os requisitos fundamentais, que desempenham um papel central para os requisitos técnicos mais detalhados.

- Documento de Requisitos de Software: a especificação de requisitos de software organiza todos os requisitos que cercam o projeto. Captura os requisitos de software para o sistema completo ou uma parte do referido sistema.

- Especificação Suplementar: este documento trata os tipos de requisitos que não se inserem no âmbito dos requisitos funcionais. Sendo, estes não são facilmente capturados nos casos de uso. Tais comportamentos do sistema incluem requisitos legais e regulamentares, atributos de qualidade, e outros não requisitos funcionais.

- Modelo de Casos de Uso: O principal objetivo da modelagem de caso de uso é fazer com que o núcleo do que um sistema deve fazer para atender às necessidades das partes interessadas. Para isso, primeiro se deve concentrar em que (o ator) irá usar, ou se será usado por ele, e, em seguida, que a funcionalidade do sistema deve fornecer (dar valor para o usuário). Em relação a isso, este artefato captura o modelo das funções do sistema e seu ambiente e serve como um contrato entre o cliente e os desenvolvedores.

- Especificação de Casos de Uso: Um caso de uso define uma sequencia de ações que serão executadas por um sistema. Este é um conceito importante, as especificações de alguns diálogos entre um ator e o sistema devem produzir valor ao cliente. Os casos de uso devem habilitar o sistema para entregar toda a funcionalidade esperada.

Voltando para o contexto da disciplina de requisitos, e das longas discussões sobre o fracasso dos projetos de software é possível verificar no relatório do *The Standish Group* (2009), como andam as entregas de sistemas:

Tabela 1 - Percentual de resultado dos projetos de Software

% de Projetos	Resultado do projeto
24%	Fracasso
44%	Sucesso parcial
32%	Sucesso

Indo um pouco mais além também é possível encontrar as principais causas de falhas em projetos de software, que segundo o próprio *The Standish Group* são:

Tabela 2 - Percentual das causas de fracasso em projetos de Software

% de causa	Causa de fracasso
13,1%	Requisitos Incompletos
12,4%	Falta de envolvimento do usuário
10,6%	Falta de recursos
9,9%	Expectativas não realistas
9,3%	Falta de apoio executivo
8,7%	Mudanças de requisitos
8,1%	Falta de planejamento
7,5%	Não precisa mais daquilo
6,2%	Falta de gestão da TI
4,3%	Analfabetismo tecnológico
9,9%	Outros

Dos itens e percentuais apresentados como causas do fracasso nos projetos de software pelos menos 40% estão diretamente relacionadas com o levantamento de requisitos, e outros 15% indiretamente relacionadas.

Sendo assim é possível concluir a importância de um levantamento e especificação de requisitos realizados de forma correta para o sucesso do projeto de software. Contudo ainda há muito que se discutir sobre as melhores formas de se extrair as necessidades dos usuários, ou ainda qual o melhor caminho a seguir para fazer parte e quem sabe aumentar os 32% de casos de sucesso em entregas de projetos de software.

3 MAPEAMENTO DE PROCESSOS

O BPM (*Business Process Management*) traz uma proposta nova para tratamento do gerenciamento de processos de negócio, e vem oferecer nove áreas de conhecimento distintas que tem o propósito de subsidiar e fornecer insumos para esta gerência.

Como objeto de estudo para este artigo é utilizada apenas uma das nove áreas de conhecimento apresentadas pelo BPM, a Modelagem de Processo de Negócio.
Segundo pode ser verificado em CAPOTE (2011, p. 115):

> "A modelagem de processos de negócio nada mais é que um conjunto de atividades necessárias para a criação de representações de processos existentes, ou que ainda estão em planejamento ou sendo projetados. Uma característica bastante marcante sobre a modelagem de processos de negócio, conforme a própria definição do tipo do processo denota, é que este tipo de modelagem deve contemplar e cobrir os processos de forma completa, ou, de ponta-a-ponta."

Em um modelo de processo são encontradas: informações do negócio, informações operacionais, informações específicas do processo e informações técnicas. Um modelo de processo completo pode atender diversos propósitos, conter diagramas, e outras informações sobre ambiente, relacionamento entre elementos, objetos, etc.

O propósito da modelagem de processo é criar uma representação dos processos descrevendo-os de forma que seja suficiente para a análise e que subsidie um melhor entendimento do negócio do cliente, apoiar o treinamento de recursos, possibilitar a avaliação de mudanças e melhorias nos processos já existentes, servir como base de comunicação e descrever requisitos para novas operações.

Como já citado um modelo de processo pode conter diagramas, um dos diagramas, o de processos, que auxilia no entendimento do processo de negócio do cliente, em CAPOTE (2011, p. 117), lê-se sobre o diagrama de processo:

> "É o primeiro passo para a representação e compreensão mais completa dos processos. Usualmente é composto apenas de fluxos simples de atividades, e por muitas vezes, representa apenas o caminho feliz, ou dia feliz dos processos – quando representamos somente as situações de sucesso na realização do processo – sem considerar exceções e falhas não tão evidentes."

Outro produto que pode ser encontrado e que também é objeto deste estudo é o mapa de processos, que pode ser considerado um diagrama de processo, acrescentado de outras informações, como atores, eventos, regras, resultados, entre outros detalhes necessários para a representação do mesmo.

O mapa de processo procura fornecer uma composição mais abrangente, uma forma de visualizar as informações que compõe todo o processo de forma mais detalhada. Por meio deste mapa é possível, por exemplo, acompanhar quem é o executor de cada tarefa, o passo a passo a serem seguidos, os eventos vinculados, as regras e até mesmo os resultados esperados. Um exemplo simples de mapa de processo pode ser visto na figura 1.

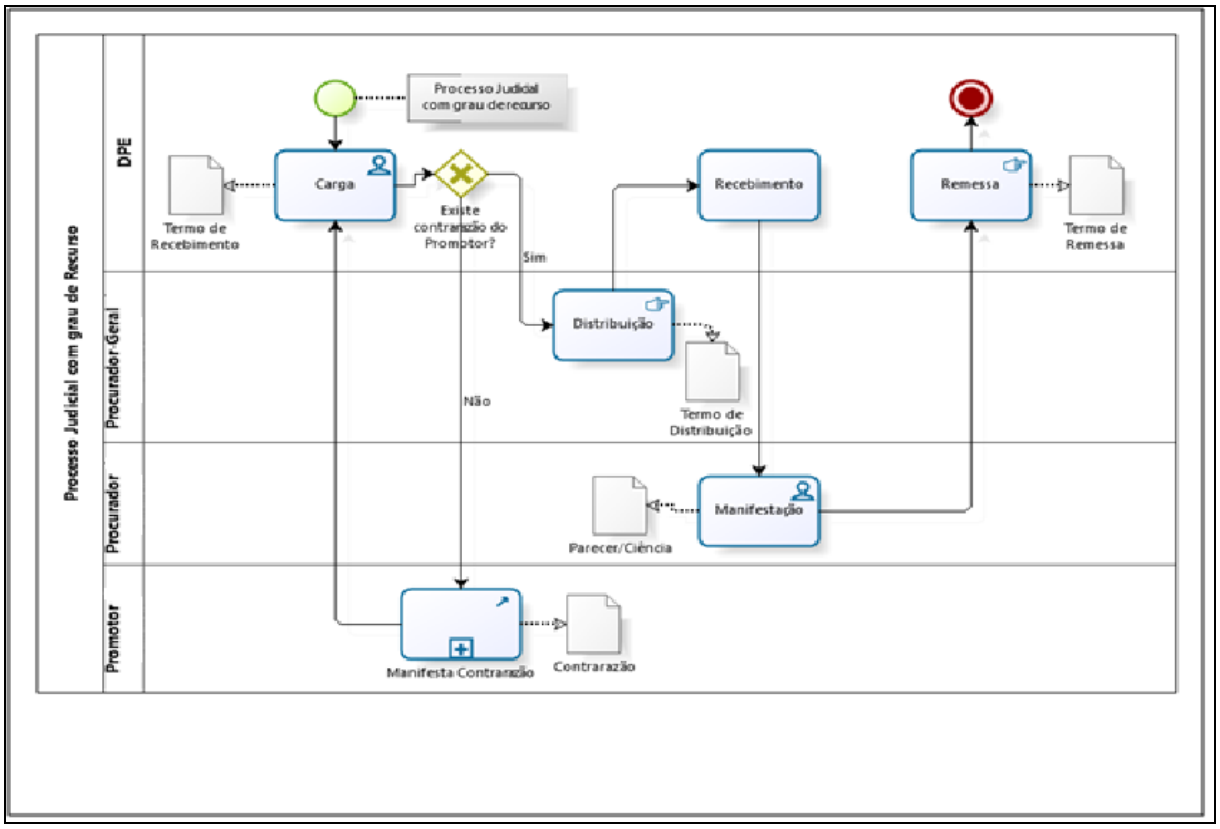

Figura 1 – Exemplo de Mapa de Processo

Mesmo não sendo de forma textual, é possível acompanhar o passo a passo pelo mapa de processo, como por exemplo, se faz um caso de uso, que neste caso, já é uma representação textual de um passo a passo.

Considerando a disciplina de requisitos tratada pelo RUP e descrita no Capítulo 2 bem como os artefatos propostos por ela, e a área de conhecimento de Modelagem de Processos proposta pela BPM bem como os produtos citados por esta área de conhecimento, é possível passar para a próxima etapa deste estudo onde será feita a análise das melhores práticas das duas propostas, e dos problemas encontrados pelas mesmas para enfim propor uma melhor forma de trabalho tanto para o cliente quanto para o profissional de informática que trabalha com a elaboração dos artefatos e processos propostos.

4 UNIFICANDO O MELHOR DOS DOIS MUNDOS

Enfim é chegado o ponto chave deste estudo, avaliar os artefatos propostos pela disciplina de Requisitos do RUP em suas diferentes fases.

Neste caso foram selecionadas as fases de concepção e elaboração do RUP, para verificar quais os benefícios e problemas apresentados pelas mesmas, e encontrados os problemas quais as soluções que podem ser adotadas tendo como base a área de conhecimento de Modelagem de Processos oferecida pelo BPM.

O primeiro artefato a ser utilizado é o Documento de Visão do Projeto, pois é este artefato que irá dar subsídio para todos os outros que serão criados posteriormente. Neste artefato é definido: quem são os *stakeholders*, quais são os problemas enfrentados por eles,

quais as necessidades que eles possuem e descendo à sua forma mais refinada é possível encontrar quais são as características que o produto de software deve ter.

A partir das definições do Documento de Visão do Projeto, ele é dividido em dois outros documentos: o Documento de Requisitos de Software onde serão concentrados e organizados todos os requisitos funcionais do sistema e o Documento de Especificação Suplementar onde serão concentrados todos os requisitos não funcionais do sistema, e então é possível manter a rastreabilidade do sistema.

Ter todos estes documentos de forma separada parece ótimo e organizado. Além de, em um cenário perfeito, possibilita um rastreamento dos requisitos desde a raiz (o início, o requisito original) até o requisito final, possibilitando acompanhar todas as mudanças sofridas, inclusões que não estavam previstas no escopo inicial, exclusões, etc., porém para o usuário final não é assim tão simples.

É comum para um analista de requisitos, ouvir inúmeras queixas dos usuários sobre a quantidade de documentos que precisa ler e assinar, além de variados questionamentos sobre o fato de não entenderem o conteúdo do documento que está sendo assinado. Alguns usuários chegam a questionar se isto não seria função da "área de TI".

Outra situação comum no mercado de informática são softwares/funcionalidades que são desenvolvidas, porém não são utilizadas pelos usuários que as solicitaram, porque na verdade não era aquilo que era necessário, não era esta a necessidade ou o problema a ser resolvido. Em inúmeros casos só se verifica o erro na identificação do requisito depois que o sistema é construído e entregue ao usuário.

Se por um lado as queixas do cliente são estas, por outro lado, os analistas de requisitos se queixam de uma enorme dificuldade de extrair do usuário o problema ou a situação atual que vivida pelos gestores, pelos responsáveis pelo sistema. Como pode ser visto em WAGNER (2012):

> "Muitas vezes nos vemos de frente com um cliente que não sabe exatamente do que precisa, ou que não consegue transmitir esta informação de maneira clara. Resultado: Você gasta horas e horas em uma reunião de levantamento de requisitos (ou briefing para os publicitários) e não consegue extrair informações relevantes para o desenvolvimento do projeto."

Por vezes o usuário acaba por detalhar demasiadamente algumas situações que confundem o analista e não acrescentam ao projeto, situações que não são relevantes para o entendimento do negócio, e acaba por esquecer-se de informações importantes, ou até mesmo essenciais, por julgar que estão implícitas, ou que são óbvias ao contexto do sistema. Assim apresentado por SILVA (2012): "[...] mas nem sempre fica claro para todos os envolvidos do projeto sobre qual é a fronteira da aplicação".

Em alguns casos o cliente (usuário) vive um caos total, sem nenhum tipo de processo ou padronização da forma de trabalho e acredita que o sistema irá solucionar todos os problemas, porém não acha ser necessário organizar o processo antes de implantar o sistema. Acreditam que o sistema trará a solução "milagrosa" que tanto se esperava, que irá solucionar tudo e trará a organização para tudo o que precisa ser organizado.

Com tantos problemas não seria necessário falar das fábricas que por vezes tem retrabalho, e até mesmo prejuízo, ao refazer funcionalidades que foram levantadas, mas não atenderam às necessidades do cliente. Como pode ser visto em VERÍSSIMO (2007):

> "Um estudo baseado em 6700 sistemas desenvolvidos em 1997 (1) demonstrou que os custos resultantes da má realização da etapa de levantamento de requisitos, podem levar os sistemas custar duzentas vezes mais que o necessário."

Um dos produtos citados no capítulo 3 que a área de conhecimento de Mapeamento de Processo propõe é bem simples e até parecido com um conhecido diagrama da UML, o de atividades, porém seguindo uma linha de pensamento nova.

O diagrama de processo traz a proposta de mapear o que já existe, para só então mapear o que se quer alcançar, ou seja, o que pode ser melhorado.

A proposta é começar entendendo a área de negócio, sem ao menos pensar no sistema, como se ele ainda não existisse, fazer um fluxo macro de como a informação trafega dentro do setor que se deseja informatizar, de como as pessoas trabalham, deixar que o próprio usuário entenda como ele trabalha, que ele mesmo encaixe suas atividades em um fluxo de informação, para que só então se comece a falar dos problemas que enfrentados.

Com esta ajuda, do Diagrama de Processos, é possível gerar um documento de visão bem mais consistente, de forma que se tenha sempre para onde olhar, nos momentos em que o foco for perdido ou o escopo sair do contexto.

Outra proposta para esta fase é uma adaptação: retirar o Documento de Requisitos de Software, que para o usuário por vezes se torna repetitivo, por citar partes que já foram vistas no Documento de Visão do Projeto.

Com o tempo que seria tomado por este documento se pode apenas focar em outros documentos que precisam ser feitos, para que possam ser trabalhados mais atentamente, com maior prazo.

Sendo assim será apresentado como resultado da fase de Concepção: um Diagrama de Processos que poderá ser evoluído posteriormente, um Documento de Visão do Projeto consistente e alinhado com o foco dado ao diagrama, e por fim a Especificação Suplementar com uma definição mais clara e trabalhada dos requisitos não funcionais que por vezes não são trabalhados de forma adequada.

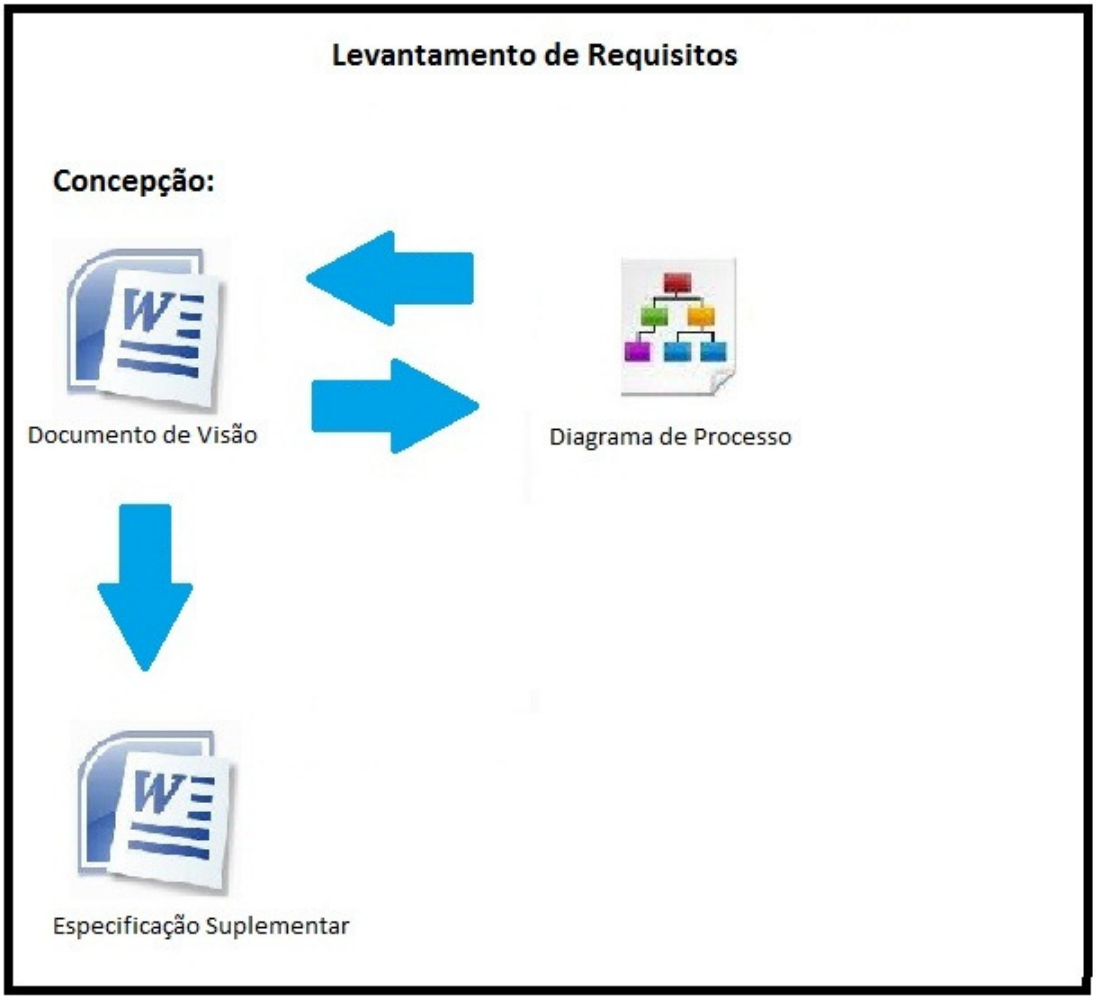

Figura 2 – Fase de concepção – Artefatos gerados

Seguindo então para a próxima e talvez mais complicada fase do RUP, a Elaboração, porém não esta não será tratada como um todo, não sertã tratado, por exemplo, a implementação do produto, apenas a especificação do mesmo.

Nesta fase é comum trabalhar com a Especificação de Casos de Uso, que se trata de um detalhamento dos casos de uso que são mapeados no Modelo de Casos de Uso, esta especificação acaba sempre acompanhada de um protótipo e outro artefato conhecido como Descrição de Interface ou Interface de Caso de Uso[1] e acaba sempre causando muitas "dores de cabeça" aos analistas de requisitos.

Dentre os problemas apresentados pela Especificação de Casos de Uso, o primeiro que pode ser citado está na forma de descrevê-lo.

Até hoje não existe um padrão, uma fórmula matemática exata para definir como um Caso de Uso deve ser escrito, o próprio RUP sugere um *template*, porém não sugere uma forma de escrita, existem vários padrões de boas práticas no mercado, as próprias fábricas de software acabam por criar os documentos de Diretrizes de Requisitos.

O fato é que de cliente para cliente, fábrica para fábrica, analista para analista, há inúmeras formas diferentes de descrever um caso de uso, o que gera também, inúmeras

[1] Artefato que descreve todos os campos apresentados na tela, contendo: descrição, tamanho, tipo, máscara, hint, obrigatoriedade, domínio, regra de apresentação, etc.

formas de interpretar um caso de uso, isso cria um grande problema quando há fábricas distintas descrevendo casos de uso e validando os artefatos entregues.

O consenso sobre o entendimento ou a forma de escrever é sempre um fator complicador, assim como é também para o usuário que precisar ler e entender o Caso de Uso para assinar. Muitas vezes este se trata de um usuário final, que não está envolvido com a área de informática, e não tem conhecimento técnico para entender os artefatos que são gerados.

A proposta para esta parte do processo de desenvolvimento de software elimina alguns artefatos e alguns pontos de conflito.

A primeira sugestão é retirar o documento de Modelo de Casos de Uso, visto que irá se trabalhar com Especificações de Casos de Uso, este artefato se faz desnecessário.

Partindo das propostas do BPM, a ideia é realizar um mapeamento de processo, partindo da evolução ou tendo como base o diagrama de processos construído na fase anterior.

Com a base já consolidada nesta etapa segue identificando no mapeamento de processo os atores envolvidos, regras negociais, mensagens, eventos que possam ocorrer e até mesmo descrever ou especificar as regras e mensagens separadamente e apenas referenciar no mapa de processos.

Nesta fase é importante também fazer uso dos protótipos que já é comprovado que auxiliam o entendimento do usuário, por apresentarem uma interface visual e interativa.

O Documento de Descrição de Interface ou Interface de Casos de Uso fica mantido visto que este auxilia o desenvolvimento e principalmente a criação do modelo de dados, sendo necessário *lincar* as mensagens aos botões para facilitar a identificação e o momento exato da apresentação das mesmas.

Fica assim representada a documentação inicial da fase de elaboração por: um Mapa de Processos ligado ao Documento de Regras de Negócio e Mensagens, a Diretriz de Usabilidade, e o Protótipo do projeto de forma objetiva e interativa.

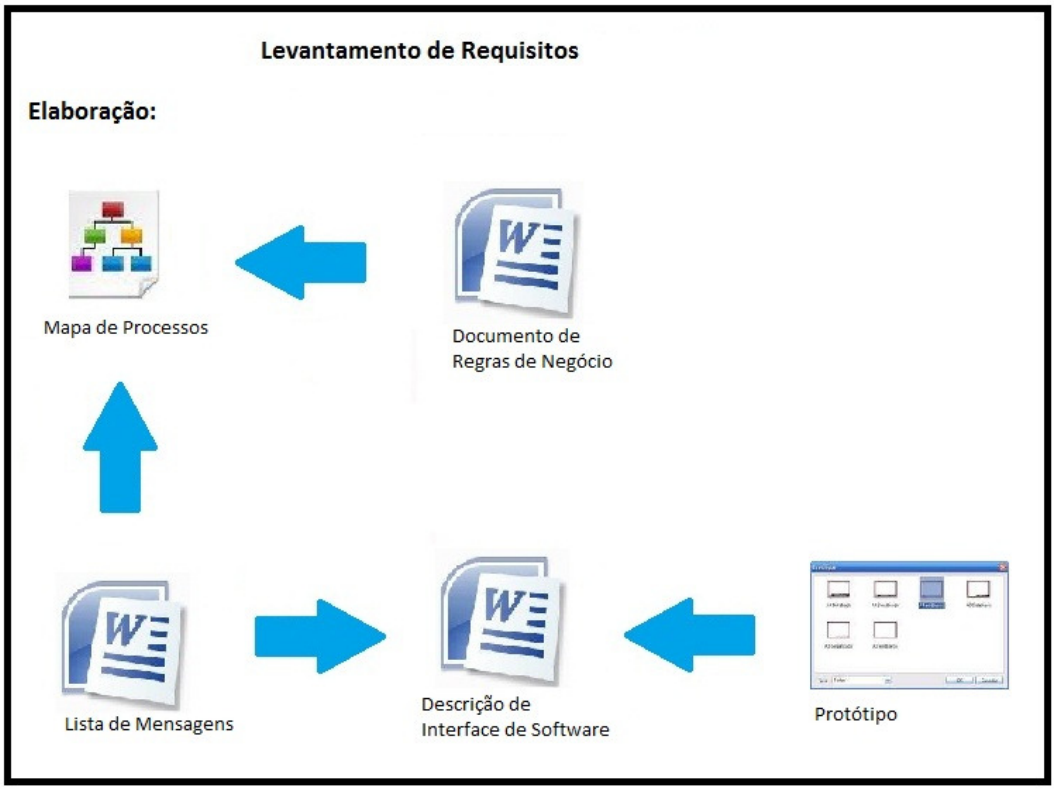

Figura 3 – Fase de elaboração – Artefatos gerados

Com essa diminuição na quantidade de artefatos, e deixando artefatos mais visuais e interativos, para que usuário final possa acompanhar e interagir, é possível minimizar os impactos gerados pelo desgaste com o excesso de documentações longas e que não agregam valor ao usuário.

5 CONCLUSÃO

Como pode ser observado ao longo deste estudo, foi separada uma pequena área de amostragem, o Levantamento de Requisitos, para que fosse possível observar os artefatos que são comumente utilizados e os processos que são seguidos. Após esta análise e sendo identificados os problemas deste universo que foi selecionado, foi então possível elaborar uma proposta de melhoria, baseada em uma junção de artefatos de dois processos distintos.

Uma restrição encontrada durante o trabalho foi não poder analisar todas as fases do desenvolvimento de software, passando pelo ciclo completo, verificando os problemas encontrados, por exemplo, na construção, nos testes, ou na transição de um sistema, para que também fosse possível propor para eles uma diretriz como sugestão de melhoria.

A conclusão que pode ser retirada é que sempre que os problemas aparecerem no processo de desenvolvimento de software e antes de se tornarem insolúveis ou desgastantes, é necessário parar, e verificar as lições aprendidas que estão em volta, na informática cada caso é um caso, e uma solução que foi ideal para um projeto não necessariamente e até dificilmente será ideal para o outro, porém juntando um pouco de cada, e inovando sempre que possível, se consegue chegar a uma solução que atenda de forma satisfatória aquele caso específico. O que não se pode é “engessar” o processo e querer fazer igual para todas as ocasiões, é sempre necessário refletir sobre o melhor caminho.

Para futuros estudos é recomendada a continuação da análise de melhorias para cada fase do RUP e cada artefato ou atividade que compõe o processo de desenvolvimento, tendo sempre em mente buscar em outros processos e metodologias uma forma de complementar ou substituir aquilo que não está dando certo.

Outra sugestão é a implantação da metodologia sugerida em um projeto real de pequeno porte para que possa ser testa a efetividade e os resultados apresentados para o mesmo de cada uma das perspectivas, cliente, analista, fábrica, dentre os outros profissionais envolvidos neste processo. Neste caso é interessante que cada uma destas experiências e retornos dos envolvidos seja documentada para continuação da analise e melhoria da metodologia.

REFERÊNCIAS

CAPOTE, Gart de B. Guia para Formação de Analistas de Processos – BPM. Rio de Janeiro: Gart Capote. 2011.

PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. Porto Alegre: AMGH, 2011.

SHUJA, Ahmad K.e KREBS, Jochen IBM rational unified process reference and certification guide. Boston: Pearson Education, Inc, 2007.

SILVA, Márcio A. **A importância do levantamento de requisitos no sucesso dos projetos de software**. Disponível em: < http://www.linhadecodigo.com.br/artigo/1685/a-importancia-do-levantamento-de-requisitos-no-sucesso-dos-projetos-de-software.aspx> Acessado em: 14 abr 2012

SOUZA, Washington. **Chaos report: Como esta a TI no mundo?** Disponível em: <http://www.blogcmmi.com.br/geral/chaos-report-como-esta-a-ti-no-mundo> Acessado em: 07 mar.2012

VERÍSSIMO, Ricardo: **Levantamento de Requisitos e Mapeamento de Processos**. Disponível em: <http://www.baguete.com.br/artigos/296/ricardo-verissimo/05/11/2007/levantamento-de-requisitos-e-mapeamento-de-processos> Acessado em: 14 abr 2012

WAGNER, Newton: **Levantamento de Requisitos: Entendendo o Cliente.** Disponível em: http://www.newtonwagner.net/entendendo-o-que-o-cliente-quer/comment-page-1/#comment-43477> Acessado em: 17/04/2012