**ENGENHARIA DE SOFTWARE – MELHORES PRATICAS EM LEVANTAMENTO DE REQUISITOS**

**FABIANO COUTINHO MENESES**

FICA – Faculdades Integradas Camões  
Alameda Dr. Muricy, 707 - Centro, Curitiba - PR, 80020-040

Curitiba - Paraná - Brasil

ADS - Análise e Desenvolvimento de Sistemas

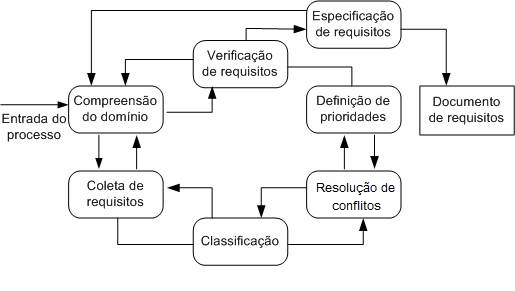
[fabianocm1995@hotmail.com](mailto:fabianocm1995@hotmail.com)

***Abstract.*** *Document that has as main objective to verify how best practices that an Analyst should take and to present a little about an initial part of a project, the Survey of Requirements. We'll be looking at how to start the process of identifying the customer problem and picking up the top topics of your authoring software or application.*

***Resumo.*** *Documento que tem como principal objetivo verificar as melhores praticas que um Analista deve tomar e apresentar um pouco sobre a parte inicial de um projeto, o Levantamento de Requisitos. Estaremos analisando como se inicia o processo de identificar o problema do cliente e levantar os principais tópicos que seu software ou aplicativo irá possuir.*

1. **INTRODUÇÃO:**

O Levantamento de requisitos é um processo que envolve todas as atividades que contribuem para a produção de um documento de [requisitos](https://pt.wikipedia.org/wiki/An%C3%A1lise_de_requerimento_de_software) e sua manutenção ao longo do tempo. É um processo iterativo, com uma contínua validação de uma atividade para outra.



O processo de levantamento de requisitos é composto por quatro principais atividades de alto nível:

* **Identificação ou levantamento**;
* **Análise e negociação**;
* **Especificação e documentação**;
* [**Validação**](https://pt.wikipedia.org/wiki/Valida%C3%A7%C3%A3o_de_requisitos).

Cada atividade é principal para um levantamento de dados correto e todas juntas são de suma importância para um projeto.

1. **REVISÕES:**

Esse documento tem como principal objetivo apresentar informações sobre a parte inicial de criação de um sistema, que irá reunir informações de diferentes diretórios bibliográficos onde constam no final do artigo.

Criado por Fabiano Coutinho Meneses – Analise e Desenvolvimento de Sistemas – Data: 13/08/2017

1. **MELHORES PRATICAS:**

A atividade de levantamento de requisitos é uma das mais importantes práticas da engenharia de software. Com ela, busca-se o entendimento das necessidades do usuário e dos requisitos de negócio, de forma a endereçá-los posteriormente através de uma solução tecnológica.

Considerando-se a complexidade na execução das atividades de levantamento de requisitos e a dependência do relacionamento entre os envolvidos, algumas boas práticas devem ser adotadas pelos analistas de forma a facilitar o processo:

* **Preparação:** Prepare-se previamente e de forma adequada para as atividades planejadas, utilizando entrevistas, questionários, brainstorms e workshops.
* **Stakeholders:** Mapeie quem serão os participantes do processo, quais os seus papéis no projeto e na organização e quais são os seus níveis de conhecimento e influência.
* **Postura:** Busque sempre a efetividade nas comunicações, assim como procure demonstrar ponderação durante as situações de conflito.
* **Entendimento:** Procure focar no entendimento do problema e evitar conclusões precipitadas.
* **Experiências Passadas:** Utilize de forma positiva as experiências vividas anteriormente para ajudar a melhor compreender o problema.
* **Documentação:** Descreva o problema de forma clara e objetiva. Em caso de dúvidas, consulte o cliente e evite inferências. Procure usar exemplos citados pelos stakeholders.
* **Validação:** Faça com que os stakeholders validem a documentação, verificando o entendimento do problema e as melhorias desejadas e eventualmente façam solicitações de mudanças.

Ao fim do processo será possível demonstrar de maneira documental o entendimento do problema, as necessidades do cliente e as oportunidades de melhorias. Isso delimitará o escopo do projeto e deverá nortear o desenho da solução, assim como o planejamento do projeto.

* 1. **ESTUDOS DE VIABILIDADE:**

Antes de se iniciar uma análise mais detalhada dos requisitos de um projeto, deve ser feito um estudo de viabilidade. Como o próprio nome sugere, a intenção desse estudo é avaliar sob o ponto de vista tecnológico e organizacional se o projeto é viável.

Uma forma de avaliar a viabilidade de um projeto é obter, através das "Partes interessadas" ou “Stakeholders” do projeto (em reuniões, entrevistas ou até mesmo questionários), a resposta às seguintes questões:

* O sistema solicitado contribui para os objetivos da organização?
* Dadas as restrições tecnológicas, organizacionais e temporais associadas ao projeto, será que o sistema poderá ser implementado?
* Caso haja necessidade de integração entre diferentes sistemas, será que esta opção esta disponível?

Esse estudo deverá culminar com a produção de um relatório e deverá determinar a continuação do desenvolvimento do projeto, tornando mais claras as restrições e definindo mesmo alguns requisitos de alto nível.

O estudo de viabilidade é uma atividade eminentemente de negócio, e que pode eventualmente envolver alguém da área de tecnologia. No entanto quem tem a competência de responder se o projeto vale ou não o investimento é o responsável pelo negócio. O estudo de viabilidade gera informações muito úteis ao início do trabalho da engenharia de requisitos.

* 1. **IDENTIFICAÇÃO:**

Pronto, foi verificado que o projeto é viável, o passo seguinte é a identificação dos requisitos.

Um bom levantamento de requisitos começa sempre pela seleção das melhores fontes de informação que serão usadas para montar a matriz de requisitos, que será matéria-prima para definir o escopo do projeto.

* 1. **ATIVIDADES ENVOLVIDAS:**

Algumas das atividades envolvidas nesta fase incluem:

* ***Compreensão do domínio:*** É importante para o analista compreender o domínio no qual a organização e o projeto se inserem.
* ***Identificar as partes interessadas:*** Estes já deverão ter sido identificados nos estudos de viabilidade.
* ***Captura:*** Consiste na obtenção dos requisitos diretamente com o cliente***(funcionais e não-funcionais)*** pretendidos para o sistema será desenvolvido.
* ***Identificação e análise de problemas:*** Os problemas devem ser identificados e devem ser propostas soluções em conjunto com as partes interessadas.
  1. **DIFICULDADES:**

As vezes podem existir algumas dificuldades típicas que lhe estão associadas:

* O cliente pode não saber exatamente o que deseja para o sistema, ou sabê-lo, mas não conseguir articulá-lo *(o que é bastante comum).*
* Os requisitos identificados podem não ser realistas *(ou algo que o cliente irá mesmo precisar)*.
* Cada parte interessada pode expressar os mesmos requisitos de formas diferentes, sendo necessário *(através de um bom conhecimento do domínio)* identificar estas situações.
  1. **PRINCIPAIS TÉCNICAS PARA LEVANTAMENTO DE REQUISITOS:**

Existem várias técnicas de identificação de requisitos, e que são adequadas a diferentes situações, vamos ver como funciona cada uma delas:

* 1. **ENTREVISTAS E QUESTIONARIOS:**

Pode considerar a técnica mais simples de utilizar. Ainda que seja bastante eficaz numa fase inicial de obtenção de dados (e mesmo de esclarecimento de algumas dúvidas), está condicionada a alguns fatores:

* Influência do entrevistador nas respostas do cliente: convém que o entrevistador dê margem ao entrevistado para expor as suas ideias sem as enviesar logo à partida.
* Relação pessoal entre os intervenientes na entrevista.
* Predisposição do entrevistado: caso, por exemplo, o papel do entrevistado venha a ser afetado pela introdução de um sistema na organização, este pode propositadamente dificultar o acesso à informação.
* Capacidade de seguir um "plano" para a entrevista.
  1. **WORKSHOPS:**

Técnica usada através de uma reunião estruturada, da qual devem fazer parte um grupo de analistas e um grupo representando o cliente, para então obter um conjunto de requisitos bem definidos. Ao contrário das reuniões, promove-se a interação entre todos os elementos presentes no workshop fomentando momentos de descontração como forma de dinamizar o trabalho em equipe, existindo um facilitador neutro cujo papel é conduzir o workshop e promover a discussão.

* 1. **CENÁRIOS:**

Uma forma de levar as pessoas a imaginarem o comportamento de um sistema é o uso de cenários. Através de exemplos práticos descritivos do comportamento de um sistema, os seus usuários podem comentar acerca do seu comportamento e da interação que esperam ter com ele. Trata-se de uma abordagem informal, prática e aplicável a qualquer tipo de sistema.

* 1. **PROTOTIPAGEM:**

O uso de ***Prototipagem*** é feito em diversas fases do processo de levantamento de requisitos. Trata-se de uma versão inicial do sistema, baseada em requisitos ainda pouco definidos, mas que pode ajudar a encontrar desde cedo falhas que através da comunicação verbal não são tão facilmente identificáveis. Neste tipo de abordagem apenas são desenvolvidas algumas funcionalidades sendo normalmente desenvolvidas primeiras àquelas que são mais fáceis de compreender por parte do utilizador e que lhe podem trazer maior valor acrescentado (principalmente na prototipagem evolutiva, isto é, aquela que mais tarde é evoluída para a fase de desenvolvimento). O uso de protótipos deve ser considerado apenas mediante uma análise custo-benefício, já que os custos de desenvolvimento de um protótipo podem facilmente crescer, sendo particularmente úteis em situações em que a interface com os usuários é, para eles, um aspecto crítico.

* 1. **BRAINSTORMING:**

*É*uma técnica para geração de ideias. Consiste em uma ou várias reuniões que permitem que as pessoas sugiram e explorem novos caminhos.

As principais etapas necessárias para conduzir uma sessão de ***Brainstorming***são:

* Seleção dos participantes.
* Explicar a técnica e as regras a serem seguidas.
* Produzir uma boa quantidade de ideias.
  1. **JAD (*Joint Application Design*):**

Técnica utilizada para promover cooperação, entendimento e trabalho em grupo entre os usuários desenvolvedores.

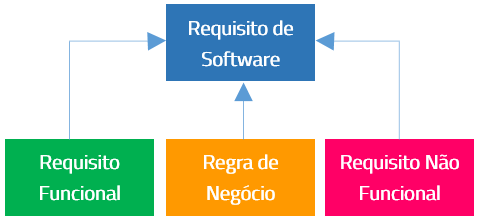
O ***JAD*** facilita a criação de uma visão compartilhada do que o produto de software deve ser. Através da sua utilização os desenvolvedores ajudam os usuários a formular problemas e explorar soluções. Dessa forma, os usuários ganham um sentimento de envolvimento.

A técnica JAD tem quatro princípios básicos:

* Dinâmica de grupo.
* Uso de técnicas visuais.
* Manutenção do processo organizado e racional.
* Utilização de documentação padrão.
  1. **REQUISITOS DE UM SOFTWARE:**

Antigamente dizia-se que requisitos eram sinônimos de funções, ou seja, tudo que o software deveria fazer funcionalmente. No entanto, atualmente assumiu-se que requisitos de software é muito mais do que apenas funções. Requisitos são, além de funções, objetivos, propriedades, restrições que o sistema deve possuir para satisfazer contratos, padrões ou especificações de acordo com os usuários.

De forma mais geral um requisito é uma condição necessária para satisfazer um objetivo.



Podemos classificar os requisitos em 2 (dois) tópicos, Requisitos Funcionais e os Não Funcionais, estaremos aprendendo um pouco sobre cada um.

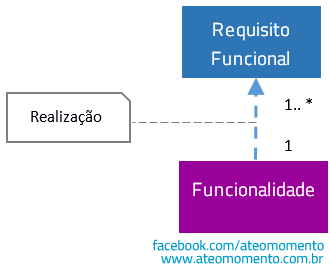
* 1. **REQUISITOS FUNCIONAIS:**

Quando falamos de um Requisito Funcional estamos nos referindo à requisição de uma função que um software deverá atender/realizar. Ou seja, exigência, solicitação, desejo, necessidade, que um software deverá materializar.

É necessário entender que**uma funcionalidade não necessariamente realizará apenas um Requisito Funcional.**

Uma funcionalidade pode realizar**vários**Requisitos Funcionais – significa que em uma funcionalidade **um ou mais** Requisitos Funcionais podem ser **atendidos,** não necessariamente apenas um.

Se pensarmos em [multiplicidade](http://www.ibm.com/support/knowledgecenter/SS5JSH_9.5.0/com.ibm.xtools.petal.ui.doc/topics/rkeydifmultiplicity.html?lang=pt-br), uma funcionalidade pode realizar um ou muitos Requisitos Funcionais.



* 1. **REQUISITOS NÃO FUNCIONAIS:**

Existe uma definição propagada na literatura de Engenharia de Software que afirma que um Requisito Funcional define **o que** o sistema fará, e o Requisito Não-Funcional define **como** o sistema fará.

Alguns afirmam que um Requisito Não-Funcional especifica como um Requisito Funcional será implementado, também não é uma boa definição, pois uma funcionalidade teoricamente pode ser implementada sem nenhum Requisito Não-Funcional no projeto e isso não gerar ônus nenhum.

Acho que nenhuma dessas definições é suficiente, não dá para entender muito bem, tem outra que diz que requisitos não funcionais são **atributos de qualidade para o sistema.**

**Alguns exemplos de requisitos não funcionais:**

* Linguagem que foi utilizada para criação do software;
* Especificações de Hardwares aonde será instalado o software;
* Plataforma que o software será destinado;
* Entre outros...

**10.0 CONCLUSÃO:**

Não importa qual é o processo, projeto, trabalho ou atividade, sempre quando precisamos dar início temos a obrigação de identificar começo meio e fim sem pular nenhuma etapa, pois todas são importantes para o seu sucesso, parece obvio com certeza, mas muitos “pulam” algumas delas, como apresentamos logo acima o início para se montar um projeto de software será com o Levantamento de Requisitos, agora um questionamento, depois de ler todo esse trabalho com o tema citado acima, você acha que essa etapa é realmente importante? Com certeza sim.

Dentro do processo de Levantamento de Requisitos identificamos várias fases, cada uma com sua importância para um bom desenvolvimento, também conhecemos algumas ferramentas diferentes que podemos estar utilizando para obtenção de dados de acordo com a situação.

Definimos Levantamento de Requisitos como uma das partes mais importantes na criação de um software pois é dessa área que irá sair a ideia de como o sistema irá funcionar, o que ele vai possuir, como será desenvolvido, em qual plataformas ou hardwares será compatível, entre outras informações.

**BIBLIOGRAFIA**

1º Engenharia de Software – Técnicas para Levantamento de Requisito. Disponivel em <<http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/9151>> Acesso dia: 08/08/2017

2º Engenharia de requisitos. Disponivel em <http:// pt.wikipedia.org/wiki/Engenharia de requisitos> Acesso dia: 08/08/2017

3º Introdução a Requisitos de Software. Disponivel em <<http://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>> Acesso dia: 10/08/2017

4º Junior Peixoto de Azevedo. Disponivel em <<http://www.redalyc.org/pdf/3967/396742032003.pdf>> Acesso dia: 05/09/2017

5º TCLBrasil, Boas práticas para Elicitação de Requisitos. Disponivel em <<https://www.ibm.com/developerworks/community/blogs/tlcbr/entry/boas_praticas_para_a_elicitacao_de_requisitos?lang=en>> Acesso dia: 21/08/2017

6º Wilson de Padúa Paula Filho. Engenharia de Software – Fundamentos, Métodos e Padrões. Disponível em <<http://aulasprof.6te.net/Arquivos_Aulas/07-Proces_Desen_Soft/Livro_Eng_Soft_Fund_Met_Padroes.pdf>> Acesso dia: 01/09/2017