



---

**FACULDADE DE TECNOLOGIA DE AMERICANA**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Karina Bordon Migotte Ioti

**PROJETOS BEM SUCEDIDOS, CLIENTES SATISFEITOS**

Americana, SP

2016



---

**FACULDADE DE TECNOLOGIA DE AMERICANA**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Karina Bordon Migotte Ioti

**PROJETOS BEM SUCEDIDOS, CLIENTES SATISFEITOS**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. MSc. Wagner Siqueira Cavalcante.

Área de concentração: Engenharia de Software.

**Americana, S. P.**

**2016**

I53p	<p>IOTI, Karina Bordon Migotte Projetos bem sucedidos, clientes satisfeitos. / Karina Bordon Migotte Ioti. – Americana: 2016. 33f.</p> <p>Monografia (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza. Orientador: Prof. Me. Wagner Siqueira Cavalcante</p> <p>1. Engenharia de software I. Cavalcante, Wagner Siqueira II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.</p> <p>CDU: 681.3.05</p>
------	--

Karina Bordon Migotte Ioti

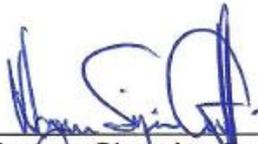
## PROJETOS BEM SUCEDIDOS, CLIENTES SATISFEITOS

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana, sob a orientação do Prof. MSc Wagner Siqueira Cavalcante.

Área de concentração: Engenharia de Software

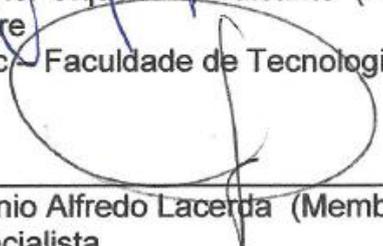
Americana, 21 de junho de 2016.

### Banca Examinadora:



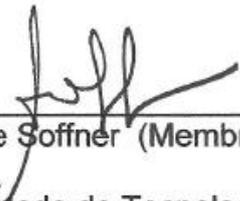
---

Wagner Siqueira Cavalcante (Presidente)  
Mestre  
Fatec – Faculdade de Tecnologia de Americana



---

Antônio Alfredo Lacerda (Membro)  
Especialista  
Fatec – Faculdade de Tecnologia de Americana



---

Renato Kraide Soffner (Membro)  
Doutor  
Fatec – Faculdade de Tecnologia de Americana

## **AGRADECIMENTOS**

Em primeiro lugar agradeço a Deus pelo privilégio e oportunidade desta conquista, bem como por todo auxílio e capacitação que tem me dado. Agradeço ao corpo docente da Fatec Americana, pela dedicação e ajuda, em especial ao professor Wagner Siqueira Cavalcante, por me orientar neste trabalho e por enriquecê-lo com seus valiosos conselhos.

Ao meu marido, Everson, que sempre me apoiou e está ao meu lado me motivando, a quem dedico todo meu amor e gratidão. Agradeço também a todos os colegas de curso, especialmente à amiga Fátima Raupp, pelo apoio e incentivo.

## DEDICATÓRIA

Dedico este trabalho a Deus, “de Quem, por Quem e para Quem são todas as coisas”, e ao meu marido Everson, cujo apoio e incentivo me foram fundamentais.

*“Pois qual de vós, pretendendo construir uma torre, não se assenta primeiro para calcular a despesa e verificar se tem os meios para a concluir? Para não suceder que, tendo lançado os alicerces e não a podendo acabar, todos os que a virem zombem dele, dizendo: Este homem começou a construir e não pôde acabar.” (Lucas 14: 28-30)*

## RESUMO

O presente trabalho tem por objetivo apontar o que é necessário para que um produto software tenha um projeto bem sucedido e o cliente sintá-se satisfeito em sua aquisição. Inicia falando a respeito da importância de conhecer o cliente e suas necessidades, que serão transformados nos requisitos, e mostra as principais classificações que os requisitos podem ter. Em seguida, discorre acerca de gerenciamento de projetos, seguindo conceitos baseados no guia *PMBOK®*, demonstrando as principais áreas de conhecimento que um gerente de projetos precisa se atentar. Para tanto, foram realizadas pesquisas em livros e publicações conceituados no ramo da Tecnologia da Informação. Ao final, um estudo de caso verídico demonstra a realidade prática encontrada comumente no mercado, respaldada por números de um relatório publicado por uma instituição de renome.

**Palavras Chave:** Projeto; Requisitos; Gerenciamento.

## ABSTRACT

*The present study aims to point out what is required for a software product has a successful project and the customer feel satisfied in your purchase. Starts talking about the importance of knowing the customer and their needs, which will be transformed into requirements, and shows the major classifications that the requirements may have. Next discusses about project management, following concepts based on the PMBOK® Guide, showing the main areas of expertise that a project manager needs to look into. Thus, it was done research in books and respected publications in the field of Information Technology. Finally, a study of real-life case demonstrates the practical reality commonly found in the market, supported by figures from a report published by a renowned institution.*

**Keywords:** *Project; Requirements; Management.*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>11</b>
<b>2</b>	<b>COMPREENDENDO AS NECESSIDADES DO CLIENTE - A ANÁLISE DE REQUISITOS.....</b>	<b>14</b>
<b>3</b>	<b>GERENCIAMENTO DE PROJETOS.....</b>	<b>19</b>
3.1	GERENCIAMENTO DA INTEGRAÇÃO.....	21
3.2	GERENCIAMENTO DO ESCOPO.....	21
3.3	GERENCIAMENTO DE TEMPO.....	22
3.4	GERENCIAMENTO DE CUSTOS .....	22
3.5	GERENCIAMENTO DA QUALIDADE.....	23
3.6	GERENCIAMENTO DE RECURSOS HUMANOS.....	23
3.7	GERENCIAMENTO DAS COMUNICAÇÕES .....	24
3.8	GERENCIAMENTO DE RISCOS.....	25
3.9	GERENCIAMENTO DE AQUISIÇÕES .....	26
3.10	GERENCIAMENTO DAS PARTES INTERESSADAS - STAKEHOLDERS .....	27
<b>4</b>	<b>ESTUDO DE CASO.....</b>	<b>28</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>31</b>
<b>6</b>	<b>BIBLIOGRAFIA .....</b>	<b>33</b>

**LISTA DE FIGURAS E DE TABELAS**

<b>Figura 1: Tipos de requisitos não funcionais.....</b>	<b>17</b>
<b>Figura 2: Ilustração de problemas comuns de comunicação em projetos.....</b>	<b>25</b>

## 1 INTRODUÇÃO

O sucesso de um software vai além de uma programação bem feita. Embora fatores como a escolha da tecnologia, linguagem e codificação sejam fundamentais, um projeto bem feito, documentado e bem gerenciado é o caminho mais seguro a ser trilhado, para se chegar ao objetivo final: um produto de qualidade e clientes satisfeitos.

Juran (1992, p. 07) define que:

Satisfação do cliente é um resultado alcançado quando as características do produto correspondem às necessidades do cliente. Ela é, em geral, sinônimo da satisfação com o produto.

Sommerville (2007, p. 5) afirma que um bom software é aquele que fornece a funcionalidade e o desempenho exigidos pelo usuário, sendo de fácil manutenção, confiável e com boa usabilidade. Sendo assim, o ponto de partida é o analista entender o que o cliente quer (requisitos) e projetar como isso será feito (projeto e gerenciamento).

Em um processo de software, há quatro atividades fundamentais comuns a toda produção, que são:

- Especificação de software, em que clientes e engenheiros definem o produto a ser produzido e suas restrições;
- Desenvolvimento de software, que corresponde a como o software será projetado e programado;
- Validação de software, necessária para verificar se o software atenderá o que o cliente deseja;
- Evolução de software, que visa permitir que o software possa ser modificado para se adaptar às mudanças dos requisitos e de mercado. (Sommerville, 2007).

Definir requisitos funcionais, não funcionais, e até mesmo o que o software/sistema não deve apresentar, estabelece o conjunto de objetivos gerais a

serem alcançados de acordo com as expectativas do cliente. Com os requisitos devidamente validados, tem-se o ponto de partida do projeto.

Um gerenciamento eficaz permite que o projeto seja executado considerando cada aspecto que pode determinar seu sucesso ou fracasso. Assim, gerenciar Integração, Escopo, Tempo, Custos, Qualidade, Recursos Humanos, Comunicações, Riscos, Aquisições e Partes Interessadas constitui a realização do projeto com maiores chances de sucesso.

O presente trabalho teve como **objetivo geral** apontar os principais fatores de gerenciamento que levam ao sucesso de um projeto de software; como **objetivos específicos** dissertar acerca de cada processo de gerenciamento, de modo a ter uma compreensão prática da importância de cada processo e de sua aplicabilidade.

A **justificativa** para este trabalho deu-se pela necessidade do conhecimento de ferramentas que possibilitem a conclusão positiva de projetos de software, tanto do ponto de vista do cliente quanto do Analista de Sistemas e Engenheiro de Software.

A **pergunta** que se procurou responder foi: “Quais práticas devem ser observadas para se obter sucesso no desenvolvimento de um projeto de software?”.

As **hipóteses** foram:

- a) A correta identificação das necessidades do cliente - análise de requisitos – faz com que o produto software seja efetivamente útil e atenda ao cliente;
- b) O bom gerenciamento de cada aspecto pertinente ao projeto (custos, tempo, escopo, entre outros) viabiliza a conclusão do projeto;
- c) Embora a definição dos requisitos tenha sido correta e o projeto bem gerenciado, o produto final não atingiu as expectativas, seja por não ser mais necessário ou por qualquer outro motivo.

A **metodologia** utilizada, quanto à natureza, foi a **pesquisa aplicada**, que, para Lakatos e Marconi (1990, p.19) caracteriza-se pelo interesse prático de que seus resultados sejam utilizados ou aplicados na solução de problemas reais.

O **método científico de pesquisa** utilizado foi o **dedutivo**, que, de acordo com Andrade (1999, p. 23), é o método que mostra uma cadeia de raciocínios em conexão, do particular para o geral, explicando fenômenos particulares relacionando com princípios gerais.

O trabalho foi estruturado em cinco capítulos, sendo que o primeiro introduz o tema, falando da metodologia de pesquisa; o segundo capítulo trata das necessidades do cliente, explicando o conceito de requisitos; o terceiro capítulo discorre acerca da gerência de projetos, explicando as principais áreas de atuação de um gerente, e o quarto capítulo apresenta um estudo de caso que ilustra os temas abordados.

Com base nas informações adquiridas a partir dos estudos realizados nos capítulos anteriores, o quinto capítulo se reserva às considerações finais.

## 2 COMPREENDENDO AS NECESSIDADES DO CLIENTE - A ANÁLISE DE REQUISITOS

Segundo Pressman (1995, p. 231):

Uma compreensão completa dos requisitos de software é fundamental para um bem-sucedido desenvolvimento de software. Não importa quão bem projetado ou quão bem elaborado seja, um programa mal analisado e especificado desapontará o usuário e trará aborrecimentos ao desenvolvedor.

Quando um software é desenvolvido, é preciso considerar diversas necessidades a serem atendidas, e as restrições operacionais. Dessa forma, os requisitos de um sistema abrangem não apenas o que o usuário final espera utilizar, mas satisfazer as várias partes envolvidas no projeto: usuário final, financiador, desenvolvedor, restrições de hardware e software, entre outros.

O processo de descoberta das necessidades envolve diálogo e interação com o cliente e seu ambiente de trabalho, buscando descobrir, refinar, modelar e especificar requisitos. É uma tarefa fundamental e complexa, pois na tentativa de se compreender o que o outro diz, podem haver interpretações incorretas.

Essa análise possibilita a especificação de função e desempenho, bem como interface e outros elementos do software, e também as restrições que serão apresentadas. Podem ser criadas representações, diagramas ou protótipos que ajudem a avaliar a compreensão do desempenho do software, servindo inclusive para avaliação final do produto desenvolvido.

É importante considerar potenciais problemas que se originam dessa interação com o cliente, tais como: omissão ou interpretação errônea de informações, conflito entre exigências feitas por diferentes pessoas, necessidades que podem modificar-se com o tempo. Quão maiores forem as necessidades ou problemas, mais complexa é a tarefa da análise, pois uma nova informação, função ou restrição poderá influenciar inúmeros outros elementos.

Para minimizar o efeito destes problemas, podem-se observar algumas técnicas aplicadas em áreas chave:

- Técnicas de comunicação: é preciso considerar fatores-chave no estabelecimento da comunicação entre pessoas, que vão desde a empatia entre as partes, até fazer as perguntas certas no tempo certo. Assim, nos primeiros contatos, é necessário buscar conhecer o cliente, sua forma de pensar, de conversar, suas necessidades, fazendo perguntas com objetivo mais amplo, estabelecendo bases de relacionamento e confiança, que ajudarão em todo o processo de trabalho relativo ao software. Mais avante, outros elementos a serem analisados são o local onde ocorrem os encontros, tempo, participantes, os chamados “mecanismos de definição” (planilha, projetor, cavalete, cartaz ou qualquer meio que facilite a exposição de ideias e a compreensão), tudo visando facilitar a comunicação e clareza de entendimento entre os envolvidos.
- Princípios de análise: a informação recebida precisa ser compreendida e dominada, a fim de que sua utilização seja eficaz; assim, a criação de modelos que a descrevam em funções e comportamentos no software ajudarão no entendimento do que a aplicação fará. Pode-se evoluir essa ideia até a prototipação propriamente dita (se viável) para um maior entendimento, tanto do cliente como do desenvolvedor.
- Especificação e revisão: por definição, especificação é uma descrição daquilo que é desejado. Tais exigências podem ser conferidas de formas diferentes, seja por detalhamento escrito ou por um protótipo executável. Uma vez feita a especificação, é importante que desenvolvedor e cliente a revejam de forma a garantir que (1) o escopo do projeto foi corretamente delineado; (2) as funções, o desempenho e as interfaces foram adequadamente definidas; (3) a análise do ambiente e dos riscos de desenvolvimento justificam sua execução; (4) o desenvolvedor e o cliente possuem a mesma percepção dos objetivos do sistema/software.

Tais observações fazem-se necessárias para atender o fator fundamental na criação de um software/sistema – a satisfação de necessidades. Essas necessidades – os chamados requisitos – podem ser classificadas como funcionais ou não funcionais.

De acordo com Sommerville (2007, p. 80):

Requisitos funcionais são as declarações de serviços que o sistema deve fornecer; como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também estabelecer explicitamente o que o sistema não deve fazer.

Pode-se ilustrar com a ideia de um sistema de controle de contas a pagar, com uma tela de cadastro de contas, onde o usuário preencheria campos específicos com dados de títulos a vencer, e um botão “salvar” que gravaria o registro em uma tabela do banco de dados acoplado à aplicação. Não haveria, nesta tela, a possibilidade de exclusão do título já gravado, sendo essa uma opção disponibilizada apenas ao administrador do sistema, que o faria por outros meios. Neste caso, esta funcionalidade do sistema atenderia ao requisito funcional “cadastrar contas a pagar”.

Requisitos não funcionais são aqueles que não dizem respeito diretamente às funções específicas do sistema; podem ser rotulados como propriedades ou características que o sistema deve apresentar, ou mesmo restrições. Confiabilidade, tempo de resposta, espaço em disco, facilidade de uso são alguns exemplos.

Muitos requisitos não funcionais dizem respeito ao sistema como um todo, sendo frequentemente mais importantes que os requisitos funcionais individuais, uma vez que, se a falha em cumprir um requisito funcional pode degradar o sistema, não cumprindo um determinado requisito não funcional, pode-se tornar todo o sistema inútil (Sommerville, p. 82). No sistema de contas a pagar, citado como exemplo anteriormente, mesmo que alguma função não fosse implementada, poderia ser utilizado, mas se não apresentar requisitos de segurança e privacidade, por exemplo, seria inviável sua utilização.

Os requisitos não funcionais nem sempre estão relacionados ao software a ser desenvolvido, podendo restringir processos de desenvolvimento, requisitos legais ou éticos, entre outros, surgindo conforme a necessidade de usuários, em razão de restrições de orçamento, de políticas organizacionais, pela necessidade de interoperabilidade com outros sistemas de software ou hardware, entre diversos fatores.

A figura 1 apresenta uma classificação de diferentes tipos de requisitos não funcionais que podem surgir.

**Figura 1 – Tipos de requisitos não funcionais**



**Fonte: Sommerville (2007)**

Estes diferentes tipos de requisitos listados na figura 1 podem ser classificados de acordo com sua procedência:

- Requisitos de produtos: que especificam o comportamento do produto – confiabilidade, portabilidade, eficiência, etc;
- Requisitos organizacionais: procedem de políticas e procedimentos nas organizações, tanto do desenvolvedor como do cliente.
- Requisitos externos: são procedentes de fatores externos ao sistema e seu desenvolvimento, como requisitos de interoperabilidade com outros sistemas, requisitos legais, que devem ser seguidos para garantir que o sistema opera de acordo com a lei, entre outros.

O processo de coleta de requisitos é fundamental, pois nessa etapa têm-se a noção do que é necessário ser feito. Embora seja um processo trabalhoso e até difícil, deve ser feito com a máxima precisão, para não gerar retrabalho, frustração e prejuízos. Com os requisitos estabelecidos, é gerado o documento de requisitos de software, que é a declaração oficial do que é exigido dos desenvolvedores de sistema.

Com as necessidades sendo entendidas por meio da análise de requisitos e feito o planejamento do projeto, segue-se para o seu desenvolvimento. O próximo capítulo apresenta as principais áreas de gerenciamento de projeto que são fundamentais para garantir o bom êxito em sua execução.

### 3 GERENCIAMENTO DE PROJETOS

Sommerville (2007, p. 61) esclarece que:

O gerenciamento de projetos de software é uma parte essencial da engenharia de software. Um bom gerenciamento não pode garantir o sucesso de um projeto. No entanto, um mau gerenciamento geralmente resulta em falha do projeto: o software é entregue com atraso, custa mais do que foi originalmente estimado e falha ao atender seus requisitos.

Alguns fatores diferenciam o produto software dos demais produtos, e isso pode tornar seu gerenciamento mais difícil, pois:

- O produto é intangível: um produto visível, como um navio, permite visualizar um atraso em sua construção; um software não possui essa facilidade, de modo que o gerente do projeto depende da informação de outras pessoas para acompanhar seu progresso;
- Não há processo de software padrão: embora nos últimos anos a compreensão dos processos de software tenha desenvolvido significativamente, ainda não há como prever se determinado processo é o ideal ou causará algum dano ao desenvolvimento;
- Grandes projetos de software são frequentemente únicos, o que torna difícil prever problemas mesmo com vasta experiência anterior, somando ainda a rápida mudança tecnológica nos computadores e nas comunicações.

Por essas particularidades, não é de surpreender que o projeto sofra atrasos e supere o orçamento previsto. Um gerenciamento eficaz visa minimizar essas dificuldades, observando-se processos-chaves, que são:

- Gerenciamento da Integração;
- Gerenciamento do Escopo;
- Gerenciamento de Tempo;
- Gerenciamento de Custos;
- Gerenciamento da Qualidade;
- Gerenciamento de Recursos Humanos;

- Gerenciamento das Comunicações;
- Gerenciamento de Riscos;
- Gerenciamento de Aquisições;
- Gerenciamento das Partes Interessadas.

Tais processos-chave foram renomeados como Áreas de Conhecimento em Gerenciamento de Projetos, de acordo com o Conjunto de Conhecimentos em Gerenciamento de Projetos (*Project Management Body of Knowledge – Guia PMBOK®*), elaborado pela entidade americana *Project Management Institute (PMI)*. Em 1981 o PMI deu início, de maneira formal, a catalogar todo conhecimento relevante à área de gerenciamento de projetos de TI, dando origem ao documento PMBOK®. Ao longo dos anos, este documento vem sendo reescrito conforme a dinâmica crescente de novas informações nesta área.

### **3.1 GERENCIAMENTO DA INTEGRAÇÃO**

O gerenciamento da integração do projeto inclui processos e atividades necessárias para identificar, definir, combinar, unificar e coordenar os vários processos e atividades das áreas de conhecimento. O gerente de processos age como integrador de processos e de pessoas, definindo onde e quando concentrar recursos, tratando problemas antes de se tornarem críticos e coordenando o trabalho visando o sucesso do projeto. Algumas atividades são:

- Desenvolver o termo de abertura do projeto: autorizar um projeto ou fase e documentar requisitos iniciais.
- Desenvolver o plano de gerenciamento do projeto: ações necessárias para definir, preparar, integrar e coordenar planos auxiliares.
- Orientar e gerenciar o trabalho do projeto: realizar trabalho definido no plano de projeto para atingir os objetivos do projeto.
- Monitorar e controlar o trabalho do projeto: acompanhar e revisar o progresso para atender aos objetivos de desempenho definidos no plano do projeto.
- Realizar o controle integrado de mudanças: revisar as solicitações de mudança, aprovação e gerenciamento de mudanças.
- Encerrar o projeto ou fase: finalizar todas as atividades de gerenciamento do projeto para terminar o projeto ou a fase.

### **3.2 GERENCIAMENTO DO ESCOPO**

O gerenciamento de escopo deve assegurar que todo o trabalho necessário esteja incluído, para que o projeto seja concluído com sucesso. O escopo inclui atributos e as funções que caracterizam o produto software bem como as tarefas que precisam ser desempenhadas para a execução da entrega. Durante a definição do escopo, é boa a prática de se definir o que está incluso e o que não está.

Algumas das atividades ligadas ao gerenciamento de escopo são:

- Planejar o gerenciamento do escopo: estabelecer e documentar como será feita a definição, validação e controle do escopo;
- Coletar os requisitos: definir e documentar as necessidades das partes interessadas;
- Definir o escopo: é feita uma descrição detalhada do projeto e do produto;
- Criar a Estrutura Analítica do Projeto (EAP): subdividir os produtos e o trabalho em pacotes mais gerenciáveis;
- Validar o escopo: formalizar a aceitação dos produtos do projeto;
- Controlar o escopo: monitorar e gerenciar eventuais alterações.

### **3.3 GERENCIAMENTO DE TEMPO**

É o processo necessário para assegurar que o projeto seja finalizado no prazo estabelecido, sendo crucial para que o trabalho seja concluído com sucesso. Para tanto, executa as tarefas:

- Planejar o gerenciamento do cronograma: estabelecer políticas e procedimentos para planejar, desenvolver, controlar e gerenciar o cronograma;
- Definir as atividades: identificar as atividades específicas que devem ser realizadas para que haja as entregas do projeto;
- Sequenciar as atividades: identificar e documentar as relações de dependência das atividades;
- Estimar os recursos das atividades: estimar quais tipos e quantidade de recursos necessários para cada atividade;
- Estimar as durações das atividades, bem como quantos períodos de execução serão necessários;
- Desenvolver o cronograma: analisar sequência, duração, recursos e restrições para elaborar o cronograma;
- Controlar o cronograma: acompanhar e gerenciar eventuais mudanças.

### **3.4 GERENCIAMENTO DE CUSTOS**

Para que o projeto seja concluído dentro do orçamento aprovado, é necessário observar os seguintes processos:

- Planejar o gerenciamento dos custos: estabelecer as políticas, procedimentos e documentação necessária para planejar, gerenciar, executar e controlar os custos do projeto;
- Estimar os custos: analisar os custos dos recursos necessários para a execução das atividades do projeto;
- Determinar o orçamento: agregar os custos estimados das atividades para estabelecer um orçamento;
- Controlar os custos: monitorar o andamento do projeto, acompanhando as despesas e fazendo ajustes necessários.

### **3.5 GERENCIAMENTO DA QUALIDADE**

O gerenciamento da qualidade tem por objetivo assegurar que o produto final satisfaça as necessidades do cliente, envolvendo todas as atividades do projeto em todo seu ciclo de vida, com atividades de melhoria contínua nos processos.

Processos do gerenciamento da qualidade do projeto:

- Planejar o gerenciamento da qualidade: Por meio da identificação dos requisitos e/ou padrões da qualidade do projeto e do produto, documentar de que modo o projeto demonstrará conformidade;
- Realizar a garantia da qualidade: certificar que os padrões da qualidade e definições operacionais apropriadas estão sendo utilizados através de auditoria dos requisitos da qualidade e dos resultados das medições do controle da qualidade;
- Controlar a qualidade: Monitorar e registrar os resultados da execução das atividades da qualidade para avaliar o desempenho e recomendar mudanças necessárias.

### 3.6 GERENCIAMENTO DE RECURSOS HUMANOS

Esta área do conhecimento abrange os processos que organizam e gerenciam a equipe do projeto, que conta com pessoas com funções e distintas responsabilidades atribuídas, focalizando o término do projeto.

Para gerenciar os recursos humanos, o gerente de projetos deve:

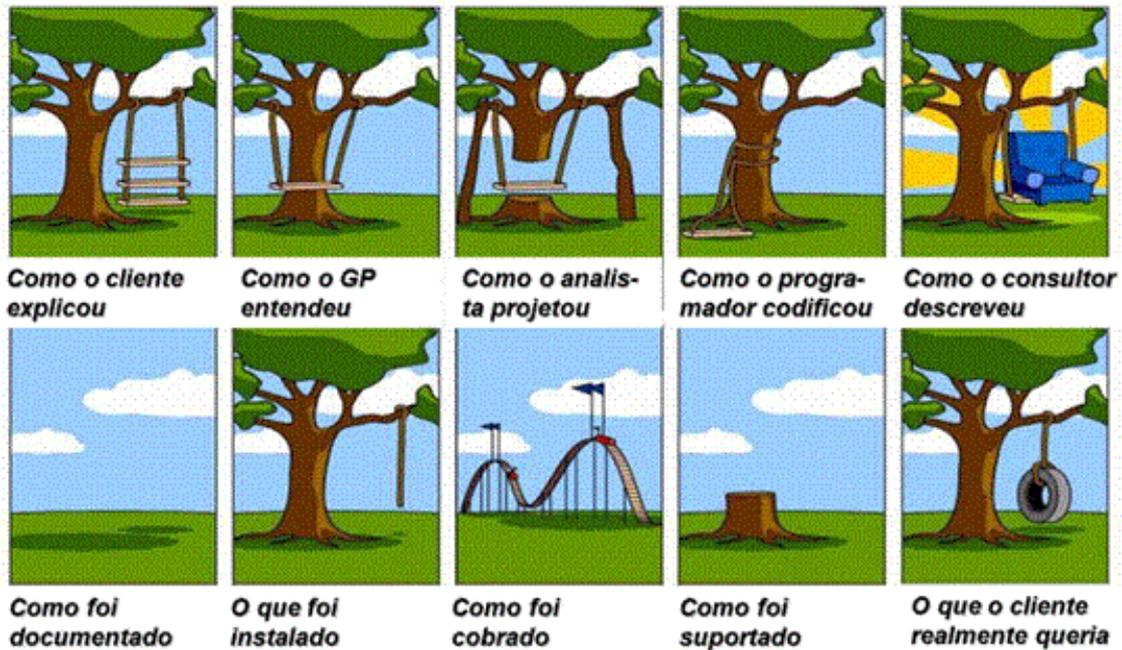
- Planejar o gerenciamento dos recursos humanos, identificando e documentando as funções, responsabilidades, competências necessárias e relações hierárquicas;
- Mobilizar a equipe, obtendo os recursos humanos necessários para o projeto.
- Desenvolver a equipe do projeto, aperfeiçoando as competências e interação dos membros da equipe para aprimorar o desempenho;
- Gerenciar a equipe do projeto, acompanhando seu desempenho, fornecendo *feedback*, resolvendo problemas e coordenando mudanças.

### 3.7 GERENCIAMENTO DAS COMUNICAÇÕES

Esta é uma das áreas mais importantes para o gerente de projeto, pois é o elo que liga pessoas, ideias e informações. Portanto, é necessário assegurar que as informações sejam geradas, coletadas, distribuídas, armazenadas, recuperadas e organizadas de maneira oportuna e apropriada. O gerenciamento das comunicações tem por principais objetivos: a) conectar as diversas partes interessadas para atender os objetivos do projeto, não obstante diferenças de interesses, cultura e outras; b) garantir que as partes estejam devidamente informadas e alinhadas; c) garantir a geração, disseminação, armazenamento, recuperação e descarte de informações do projeto.

A figura 2 ilustra os desafios que podem ocorrer frente à uma comunicação falha:

Figura 2 – Ilustração de problemas comuns de comunicação em projetos



Fonte: <http://escritoriodeprojetos.com.br/gerenciamento-das-comunicacoes-do-projeto.aspx>

(acesso em 05/2016)

Processos do gerenciamento das comunicações do projeto:

- Planejar o gerenciamento das comunicações: identificar as necessidades e relevância de informações e definir abordagem de comunicação;
- Gerenciar as comunicações: disponibilizar as informações necessárias às partes interessadas conforme planejado, o que envolve todo o ciclo de vida da informação descrito no plano de gerenciamento da comunicação, composto por gerar, coletar, distribuir, armazenar, recuperar e até descartar a informação.
- Controlar as comunicações: controlar e monitorar as comunicações de modo a garantir que as necessidades de informação das partes interessadas sejam atendidas.

### 3.8 GERENCIAMENTO DE RISCOS

Segundo o Guia PMBOK®, o gerenciamento dos riscos do projeto inclui os processos de planejamento, identificação, análise, planejamento de respostas,

monitoramento e controle de riscos de um projeto. Seu objetivo é maximizar a exposição aos eventos positivos e minimizar a exposição aos eventos negativos.

Processos do gerenciamento dos riscos do projeto:

- Planejar o gerenciamento dos riscos: estabelecer como administrar as atividades de gerenciamento de riscos para o projeto;
- Identificar os riscos: avaliar quais riscos podem comprometer o projeto e documentar suas características;
- Realizar a análise qualitativa dos riscos: verificar a exposição ao risco e priorizar aqueles que precisam de análise ou ação adicional;
- Realizar a análise quantitativa dos riscos: efetuar a análise numérica do efeito dos riscos identificados nos objetivos gerais do projeto;
- Planejar as respostas aos riscos: desenvolver opções e ações para aumentar as oportunidades e reduzir as ameaças aos objetivos do projeto.
- Controlar os riscos: Fazer a monitoragem e controle os riscos durante o ciclo de vida do projeto.

### **3.9 GERENCIAMENTO DE AQUISIÇÕES**

O gerenciamento de aquisições do projeto refere-se aos processos de compra ou aquisição de produtos, serviços diretos ou terceirizados. Tais processos incluem:

- Planejar o gerenciamento das aquisições: documentar quais compras serão realizadas, possíveis fornecedores e como será o processo;
- Conduzir as aquisições: analisar propostas, definir um fornecedor e redigir os contratos adequados e pertinentes;
- Controlar as aquisições: gerenciar as aquisições, monitorando se estão de acordo com o contrato pré-estabelecido e realizar eventuais mudanças necessárias;
- Encerrar as aquisições: concluir todas as aquisições do projeto.

### 3.10 GERENCIAMENTO DAS PARTES INTERESSADAS - STAKEHOLDERS

As partes interessadas – *stakeholders*, no inglês – são todas as pessoas e empresas envolvidas com o projeto, tanto clientes como desenvolvedores, e são responsáveis por atender o objetivo do projeto.

Para gerenciar as partes interessadas, é preciso:

- Identificar as partes interessadas, seus interesses e pontos de vista, envolvimento e impacto no sucesso do projeto;
- Planejar o gerenciamento das partes interessadas: desenvolver estratégias para minimizar possíveis resistências das partes interessadas e garantir que sejam participantes no projeto;
- Gerenciar o engajamento das partes interessadas: comunicar e interagir com as partes interessadas para atender suas necessidades e solucionar as questões quando ocorrem;
- Controlar o engajamento das partes interessadas: monitorar como as partes relacionam-se entre si, eliminando resistências e aumentando o suporte ao projeto.

Esses são os principais processos das áreas de conhecimento mencionadas no guia PMBOK® atualmente. No entanto, com a dinâmica da tecnologia e do mercado de tecnologia da informação, este documento é constantemente atualizado, com base nas experiências dos profissionais da área.

A aplicação destes conceitos pode ser o fator determinante do sucesso do projeto, conforme demonstrará o estudo de caso no próximo capítulo.

## 4 ESTUDO DE CASO

Uma empresa no ramo de venda de materiais de construção, com atuação há mais de 30 anos no mercado, localizada na cidade de Americana-SP, possuía um sistema integrado de gerenciamento que, com o crescimento da loja, aumento dos itens de venda, aumento do fluxo de informação e inovação da tecnologia, tornou-se obsoleto. O dono da empresa, então, decidiu substituí-lo por um novo sistema, e, após algumas pesquisas, resolveu contratar uma empresa de soluções em software localizada na cidade de Araraquara-SP.

Esta desenvolvedora era uma empresa familiar, com cerca de 5 anos de experiência, composta pelo pai, que geria as contas e vendia os produtos, o filho, que desenvolvia, programava, gerenciava a equipe, e mais 2 programadores que auxiliavam os desenvolvimentos, porém tudo que era desenvolvido tinha que passar pelo filho, extremamente centralizador.

A escolha desta empresa deu-se, principalmente, pelo fator financeiro. O dono da loja de materiais era o financiador do produto, e não tinha nenhum conhecimento dos processos na loja; estava apenas interessado no resultado final.

Nas reuniões de negociação, foi apresentado um protótipo do sistema, que funcionava adequadamente. Não havia nenhum profissional de tecnologia da informação representando a loja, que pudesse questionar acerca do produto que estavam vendo.

Com a implantação deste sistema, revelou-se seu real estado: estava inacabado, com muitas funções não funcionando, diferente do que foi apresentado.

Por já ter sido pago, e com as promessas que tudo seria arrumado rapidamente, seguiu-se meses de frustração e discussões, que quase tiveram desfecho judicial e esgotamento físico do filho, programador. Após mais de 2 anos desta situação, conseguiu-se que o sistema pudesse funcionar como um todo, e, em cerca de 10 anos de sua implantação, ainda gera reclamações por parte dos usuários, que tiveram que se adaptar ao que tinham para trabalhar.

Nesse exemplo pode-se observar o descontentamento evidente do cliente e, com isso, pode-se dizer que esse foi um projeto mal sucedido. Embora o fato crucial seja que o cliente foi “enganado”, alguns pormenores deste caso demonstram como este projeto caminhou na contramão dos conceitos anteriormente estabelecidos:

- As partes interessadas: ambos os lados não tinham a estrutura necessária, sendo que, do lado do cliente, faltou a participação efetiva dos usuários do sistema, que questionariam suas funcionalidades, bem como um profissional da tecnologia de informação, que analisaria os aspectos técnicos relativos ao software, com possibilidade de detectar ser o produto apresentado apenas um protótipo, e outros aspectos importantes, como o despreparo da empresa contratada. Do outro lado, a empresa desenvolvedora não contava com uma equipe preparada e experiente, nem mesmo um gerente eficaz; resumidamente, era um pai que queria vender e lucrar e um filho sobrecarregado que não conseguia gerar o produto que atenderia os requisitos;
- A análise de requisitos: embora não tenha sido originalmente um produto desenvolvido “sob medida” para o cliente, por não estar completo no momento da implantação, houve a possibilidade de serem atendidos alguns requisitos solicitados pelo cliente, o que não se pode considerar um aspecto positivo, pois apenas algumas funcionalidades não tornariam o sistema consideravelmente ajustado às necessidades particulares, nem tampouco genérico do ponto de vista comercial à empresa desenvolvedora. Não houve uma elucidação ou validação dos requisitos;
- Não foi determinado um escopo para o projeto – cada nova solicitação do cliente ia sendo acrescentada à enorme lista de atividades a serem realizadas ou corrigidas;
- O custo dificilmente pode ter sido devidamente calculado, pois além das atividades extras, o fato da empresa situar-se em outra cidade cerca de 170 km de distância e a necessidade de reuniões presenciais, por conta destes desajustes, causaram despesas extras para ambas as partes, sem falar em outros prejuízos de negócios para ambos, seja por dificuldades operacionais vividos pelo cliente, ou o atendimento a potenciais mercados de venda do produto de software na região, para a desenvolvedora;

- A qualidade, sem dúvida, ficou comprometida, pois além do produto final ficar descaracterizado de sua ideia inicial, o tempo correndo contra causa uma pressão que sacrifica a qualidade em prol da produtividade;
- O fator 'riscos' foi ao extremo, ao ponto da empresa desenvolvedora quase sofrer um processo judicial por perdas e danos, tal o descontrole nas áreas observadas anteriormente;

Yourdon (1999, p. 07) aponta diversas razões para que um projeto falhe e se torne o que ele denomina “virtualmente impossível”: 1) política, quando a execução do projeto é meramente um jogo de poder entre executivos; 2) promessas ingênuas feitas pelo departamento de marketing ou gerentes de projetos inexperientes; 3) otimismo ingênuo da juventude, que acha que pode desenvolver um sistema em um fim de semana; 4) a inexperiência de empresas que querem empreender; 5) a mentalidade errônea que programadores nunca dormem; 5) intensa concorrência de mercado e novas tecnologias emergentes; 6) pressão causada por regulamentações governamentais inesperadas; 7) crises inesperadas, como uma empresa fornecedora que pede concordata ou a morte de pessoas essenciais na equipe.

Estes são apenas alguns pontos negativos apontados, no entanto, uma análise mais profunda certamente revelaria outros problemas e consequências em diversos níveis de complexidade.

## 5 CONSIDERAÇÕES FINAIS

O avanço da tecnologia trouxe consigo a expansão da comunicação e do conhecimento, permitindo aos profissionais aprender com a experiência alheia. De fato, melhores práticas na engenharia e gerenciamento de projetos apresentadas neste trabalho originam-se das experiências vividas por profissionais da área. Embora cada projeto seja único, as dificuldades podem ser muito parecidas, assim como os interesses comerciais não fogem muito à regra.

Apesar disso, o relatório "*the chaos report*", um estudo feito com 365 empresas e um total de 8.380 projetos, publicado em 1994 pelo THE STANDISH GROUP, uma organização independente de tecnologia de informação com atuação internacional, mostra que apenas 16,2% dos projetos estudados podem ser considerados de sucesso, sendo que 52,7% foram concluídos, mas excederam em alguma área (custos, tempo) ou não tiveram todas as funcionalidades devidamente implementadas. Os 31,1% restantes são projetos fracassados, que nem sequer foram concluídos (THE STANDISH GROUP, 1994).

Observa-se, portanto, que, embora fundamental, apenas o conhecimento das melhores práticas não é garantia do sucesso. Outros fatores parecem pesar na hora de por em prática o conhecimento.

Analisando o estudo de caso, pode-se deduzir que, na ânsia de se colocar no mercado de trabalho, gerar capital, muitas empresas prometem além do que podem cumprir. A inexperiência conta também como divisor de águas, pois não há parâmetros suficientes para a base real do que será, na prática, o desenvolvimento de um determinado projeto.

O mercado capitalista atual deseja tudo "para ontem", e há a pressão imposta pela ideia do concorrente sempre à frente. Portanto, infere-se que o fator tempo é determinante: promessas ilusórias tendem a "fechar as portas" quando não cumpridas e, por outro lado, a necessidade de rapidez é real e presente em praticamente todos os setores econômicos.

Para se ponderar quais são as características dos projetos com sucesso, pode-se levar em consideração que, segundo o relatório do THE STANDISH

GROUP (1994), os cinco fatores mais encontrados em projetos bem sucedidos são, na ordem que se segue: envolvimento do usuário; apoio da alta gestão; declaração de requisitos claros; planejamento apropriado; expectativas realistas e, em contrapartida, observa-se que os projetos fracassados têm como características principais, nesta ordem: requisitos incompletos ou inconsistentes; falta de envolvimento do usuário; falta de recursos; expectativas irrealistas; falta de apoio executivo; mudanças nos requisitos e especificações; falta de planejamento; o projeto não era mais necessário; falta de gerenciamento de tecnologia da informação; analfabetismo técnico.

Assim, os principais fatores que normalmente definiriam o curso de um projeto seriam a participação do usuário e a comunicação eficiente entre os envolvidos, pois desta forma os requisitos se tornariam mais claros e o trabalho seria validado, o tempo seria maximizado e os custos controlados - apenas pela validação do trabalho, já que não haveria “trabalho fútil” ou mudanças constantes. Além disso, aumentariam as chances de que a expectativa do cliente, mais realista, seria satisfeita.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, M. M. de. **Introdução e metodologia do trabalho científico:** elaboração de trabalhos na graduação. 4ª ed. São Paulo: Atlas, 1999. p.23.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **NBR 10520:** informação e documentação: citações em documentos: apresentação. Rio de Janeiro: ABNT, 2002. 7p.

DINSMORE, P. C.; CABANIS-BREWIN, J. **AMA: manual de gerenciamento de projetos.** Tradução de Adriane Cavaliere. 2ª ed. Rio de Janeiro: Brasport, 2014. 512p.

FERENHOF, H. A. **Stakeholders: fator determinante para o sucesso.** In: 9o. Seminário Internacional de Gerenciamento de Projetos, 2009, São Paulo. Seminário Internacional de Gerenciamento de Projetos. São Paulo: PMI/SP, 2009. Disponível em: <http://www.tiespecialistas.com.br/2011/01/stakeholders-fator-determinante-para-o-sucesso-em-projetos>. Acesso em: 16/04/2016.

JURAN, J. M. **A qualidade desde o Projeto.** Tradução de Nivaldo Montingelli Jr. São Paulo: Cengage Learning, 2009. 551p.

LAKATOS, E. M.; MARCONI, M. A. **Técnicas de Pesquisa:** planejamento e execução de pesquisas, amostragem e técnicas de pesquisas, elaboração, análise e interpretação de dados. 2ª ed. São Paulo: Atlas, 1990. p. 19.

PRESSMAN, R. S. **Engenharia de Software.** Tradução de José Carlos Barbosa dos Santos. São Paulo: Pearson Makron Books, 1995. 1056p.

SOMMERVILLE, I. **Engenharia de Software.** Tradução de Selma Shin Shimizu Melnikoff, Reginaldo Arakaki, Edílson de Andrade Barbosa. São Paulo: Pearson Addison-Wesley, 2007. 552 p.

YOURDON, E. **Projetos Virtualmente Impossíveis: Guia completo do Desenvolvedor de Software para Sobreviver aos Projetos Virtualmente Impossíveis.** Tradução de Regina Cláudia Loverri. São Paulo: Makron Books, 1999. 202p.

THE STANDISH GROUP(1994). **The chaos report.** Disponível em: <[https://www.standishgroup.com/sample\\_research\\_files/chaos\\_report\\_1994.pdf](https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf)>. Acesso em: 20/05/2016.