

Thúlio dos Santos Lins de Araújo
Josemary Marcionila Freire Rodrigues
de Carvalho Rocha
Hercilio de Medeiros Sousa
Cícero de Sousa Lacerda

GERENCIAMENTO DE PROJETOS

para pequenas e médias
empresas de T.I.



iesp
faculdades

Thúlio dos Santos Lins de Araújo
Josemary Marcionila Freire Rodrigues de Carvalho Rocha
Hercílio de Medeiros Sousa
Cícero de Sousa Lacerda

GERÊNCIA DE PROJETOS PARA PEQUENAS E MÉDIAS EMPRESAS DE T.I.

Editora do IESP
Cabedelo
2019



INSTITUTO DE EDUCAÇÃO SUPERIOR DA PARAÍBA

DIRETORA GERAL

Érika Marques

DIRETORA ACADÊMICA

Iany Barros

DIRETOR
ADMINISTRATIVO

Richard Euler Dantas de Sousa

EDITORES

Cícero de Sousa Lacerda
Hercílio de Medeiros Sousa
Jeane Odete Freire Cavalcanti
Josemary M. Freire Rodrigues de Carvalho Rocha

CORPO EDITORIAL

Hercílio de Medeiros Sousa - Computação
José Carlos Ferreira da Luz - Direito
Luciane de Albuquerque Sá de Souza - Administração
Maria da Penha de Lima Coutinho - Psicologia
Rafaela Barbosa Dantas - Fisioterapia
Rogério Márcio Luckwu dos Santos - Ed. Física
Thiago Bizerra Fideles - Engenharia de Materiais
Thiago de Andrade Marinho - Mídias Digitais
Thyago Henriques de Freire - Ciências Contábeis

**Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Padre Joaquim Colaço Dourado (IESP)**

G367 Gerenciamento de projetos para pequenas e médias empresas de TI
[recurso eletrônico] / Organizadores, Thúlio dos Santos Lins de
Araújo ... [et al.]. - Cabedelo, PB : Editora IESP, 2019.
51 p.

Formato: E-book

Modo de Acesso: World Wide Web

ISBN 978-85-5597-114-3

1. Gerencia de projeto. 2. Pequenas e médias empresas -
Processos. 3. Tecnologia da informação. 4. Gerência de projeto. I.
Araújo, Thúlio dos Santos Lins de. ... [et al.]. IV. Título.

CDU: 005.8

Bibliotecária: Angélica Maria Lopes Silva – CRB-15/023

**GERÊNCIA DE PROJETOS
PARA PEQUENAS E MÉDIAS
EMPRESAS DE T.I.**

SUMÁRIO

APRESENTAÇÃO.....	6
1 INTRODUÇÃO.....	7
1.1. Objetivos.....	7
1.2. Justificativa	8
2. FUNDAMENTAÇÃO TEÓRICA	9
2.1. Gerência de Projetos	9
2.2. Caracterizando um Projeto	10
2.3. O processo de software e a gerência de projetos.....	11
3. Micro e Pequenas Empresas de Software	12
3.1 O mercado	12
3.2 Organização	13
3.3. Colaboradores	13
3.4 Problemas das Empresas na gestão de TI	14
4. Processos e Gerência de Software	15
4.1. Definição de Processo.....	15
4.2 Definição de Processo de Software	15
5. MODELOS DE PROCESSOS	16
5.1. Modelo em cascata.....	16
5.2. Modelo iterativo e incremental	17
5.3. Diferenciando os modelos em cascata, iterativo e incremental.....	17
5.4. Modelo em espiral.....	19
5.5. Modelo Ágil.....	20
6. O PROCESSO PROPOSTO	21
6.1. Níveis de detalhamento do processo.....	21
6.2. Nível do Ciclo de Vida	23
6.3. Verificação da qualidade	27
7 A GERÊNCIA DO PROJETO	29
7.1. Os papéis, responsabilidades e artefatos.....	29
7.2. Clientes.....	29
7.3. Analista de Requisitos	30
7.4. Gerente de Projetos.....	31
7.5. Arquiteto de Software	32
7.6. Desenvolvedor	33

7.7 Analista de Testes	33
7.8 Analista de Suporte	34
7.9 Analista de Qualidade.....	34
7.10 Confrotamento de papéis	35
7.11 Distribuindo papéis em uma microempresa	35
7.12 Nível das atividades.....	36
8 METODOLOGIA	47
9 CONSIDERAÇÕES FINAIS	48
Referencias Bibliográficas	49

APRESENTAÇÃO

Este trabalho apresenta uma proposta de um processo de desenvolvimento de software para micro e pequenas empresas de tecnologia da informação. A partir dele, também foram realizadas pesquisas de campo e bibliográficas, observando características do mercado-alvo. Veio a surpresa! Não só eles, mas uma grande fatia do mercado brasileiro de software é composta por micro e pequenas empresas que precisam adicionar qualidade ao seu processo de produção de software. Por isso, esta proposta se torna também um estudo que pode ser amplamente utilizado para reduzir custos e melhorar a distribuição dos recursos de uma micro e pequena empresa de TI.

Com o objetivo de embasar teoricamente os leitores, foram discutidos conceitos de alguns dos principais autores (Pressman, Sommerville), processos, metodologias (sequencial, interativo e incremental, espiral, ágil) e assuntos na atualidade, de forma que o leitor terá uma maior intimidade com os termos e técnicas que serão usados no decorrer do texto.

Ao final o leitor terá adquirido uma carga de conhecimentos que lhe proporcionará entender os objetivos e os métodos utilizados no processo de forma integral. Além disso, terá em mãos um processo de desenvolvimento de software que mostrará formas de como aproveitar melhor seus recursos, proporcionando um aumento na qualidade dos serviços prestados e na produtividade da empresa como um todo.

1 INTRODUÇÃO

Diante de um cenário de constantes mudanças, evoluções tecnológicas e transformações políticas e econômicas, as organizações veem cada vez mais a necessidade de melhorar seu relacionamento entre custo e benefício. Hoje em dia, o ambiente de negócios está cada vez mais competitivo e exigem das empresas uma atuação profissional e excelência operacional. Tais expectativas se tornam ainda mais desafiadoras quando se trata de pequenas e médias empresas, com limitação de recursos, sejam eles humanos, financeiros ou de infraestrutura, e uma competitividade de mercado que muitas vezes nivelam por baixo a qualidade do produto dessas empresas.

A necessidade da gerência de projetos vem atingindo empresas de TI em todos os níveis, porém muitas delas não estão preparadas para aplicá-la ou não conhecem bem a metodologia e as melhores práticas. Com isso, a maioria delas utiliza metodologias complicadas e burocráticas, dificultando assim o trabalho de seus funcionários. Muitas das empresas têm suas funções a serem seguidas, porém não existe uma forma de executá-las, onde implica desde comunicação de informações do trabalho a ser executado até o cronograma a ser cumprido. Por esses motivos questiona-se qual a relevância dos fundamentos da gestão de projetos em pequenas e médias empresas.

O propósito deste trabalho é mostrar boas práticas e metodologias a serem seguidas para implantar a gerência de projetos em empresas de pequeno e médio porte. Será feito um estudo objetivo das principais metodologias de gerenciamento de projeto, voltando-os para a aplicação na área de Tecnologia da Informação (TI). O intuito deste estudo é saber como se deve escolher e utilizar as melhores metodologias de gestão de projeto, que são mais conceituadas no mercado, de acordo com as necessidades de cada empresa e seu projeto. Com organização, planejamento, metas e prazos bem definidos, a empresa já tem um bom começo para obter sucesso em seu projeto.

1.1. OBJETIVOS

1.1.1. Objetivo geral

Analisar a relevância e os fundamentos da Gerência de Projetos em pequenas e médias empresas, com o objetivo de difundir os papéis do gestor e a importância do mesmo para as organizações.

1.1.2.Objetivos específicos

- Explanar as principais definições existentes da Gerência de Projetos;
- Descrever o papel do Gestor de TI na organização;
- Analisar a atual realidade da Gerência de Projetos nas pequenas e médias empresas e suas dificuldades de implantá-la;
- Apresentar um modelo de Gestão eficaz para empresas de pequeno e médio porte;

1.2. JUSTIFICATIVA

O assunto tratado nesta monografia é de fundamental importância para identificar as dificuldades da implantação da Gerência de Projetos em pequenas e médias empresas. Muitas delas, por seguirem modelos de metodologias burocráticas e práticas menos utilizadas na atualidade, acabam encontrando tais dificuldades e assim comprometem o desenvolvimento do projeto ou tornam desgastante para o pessoal envolvido.

Com um gerenciamento de projeto adequado, a organização será capaz de atender as exigências do cliente moderno com maior qualidade e preço competitivo, além de se destacar perante o mercado.

Várias medidas podem ser tomadas na tentativa de alcançar este objetivo, uma delas está na prática de gerenciar projetos de uma forma profissional e planejada. O gerenciamento de projetos compreende a concepção de metas e objetivos do projeto, a elaboração de um plano, a execução do plano e a revisão e controle do projeto. Por fim, o gerenciamento de projetos oferece uma grande variedade de princípios, procedimentos, habilidades, ferramentas e técnicas que são necessárias para que se possa atingir os objetivos planejados.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. GERÊNCIA DE PROJETOS

Antes de abordarmos o assunto Gerência de Projetos, devemos entender primeiramente o que é um Projeto. Um projeto ele é único, pois ele não se trata de uma operação de rotina. O projeto é temporário no sentido de que tem um início e fim definidos, além de escopo e recursos para execução do mesmo com a finalidade de produzir um produto ou serviço.

Segundo Valeriano (1998), “o projeto é entendido como um conjunto de ações, executadas de forma coordenada por uma organização transitória, ao qual são alocados os insumos necessários para, em um dado prazo, alcançar um objetivo determinado”.

Um projeto ele é dividido em etapas ou fases distintas, também chamadas de ciclo de vida do projeto. Iniciação – Planejamento – Execução – Monitoramento e Controle – Encerramento, são fases que o PMI (Project Management Institute – Instituto de Gerenciamento de Projetos), define como ciclo de vida de projeto. Segundo a instituição, todos os projetos podem ser executados com base nessas cinco fases. A Figura 1 abaixo ilustra o ciclo de vida do projeto.



Figura 1: Ilustração do Ciclo de Vida do Projeto
Fonte: RBplan – <http://www.rbplan.com.br>

A definição do ciclo de vida pode ajudar o gerente de projetos a esclarecer se deve tratar o estudo de viabilidade do projeto como a primeira fase do projeto ou como um projeto autônomo separado. Cada fase do projeto é marcada pela entrega de um ou mais produtos (deliverables), como estudos de viabilidades ou protótipos funcionais. No início de cada fase define-se o trabalho a ser feito e o recurso envolvido na sua execução. O fim da fase é marcada por uma revisão dos produtos e do desempenho do projeto até o momento.

2.2. CARACTERIZANDO UM PROJETO

Os projetos são normalmente autorizados como resultado de uma ou mais considerações estratégicas. Estas podem ser uma demanda de mercado, necessidade organizacional, solicitação de um cliente, avanço tecnológico ou requisito legal.

Um artigo publicado na ABEPRO (Associação Brasileira de Engenharia de Produção), fala que os projetos de TI são diferentes e potencialmente mais difíceis que outros projetos de engenharia, uma vez que são caracterizados por sua alta complexidade e por altas chances de falhas além de prazos muito curtos. Para Gervazoni (2006), as Diretrizes para Qualidade de Gerenciamento de Projetos), 2003), fala do projeto onde ele deve ser único e pode ser constituído de um grupo de atividades coordenadas e controladas com data para início e término, que é a chave para se determinar se realmente estamos em um projeto. Se você estiver empenhando forças para realizar ou desenvolver um produto ou serviço e não possui data de início e fim, é provável que você não esteja em um projeto, sendo assim seu gerenciamento pelo PMBOK fica comprometido.

Partindo do esclarecimento do que é um projeto, será explanado agora a definição de Gerência de Projetos. Segundo o Project Management Institute (PMI), gestão de projetos é a aplicação de “conhecimento, habilidades, ferramentas e técnicas aplicadas às atividades do projeto para atender ou exceder as expectativas dos patrocinadores ou acionistas”. Para Phillips (2003), gerência de projeto é a capacidade de administrar uma série de tarefas cronológicas que resultam em uma meta desejada.

A gerência de projetos de TI é a capacidade de equilibrar dedicação e implementação de tecnologia, ao mesmo tempo liderando e inspirando os membros da equipe. É claro que a meta da gerência de projetos não é a tecnologia em si, mas um movimento no sentido da assistência ao cliente, da melhora na qualidade do produto e do aumento nos lucros. (PHILLIPS, 2003).

2.3. O PROCESSO DE SOFTWARE E A GERÊNCIA DE PROJETOS

Sobre os processos de desenvolvimento de software, existem inúmeras tentativas de definição, onde cada autor enfatiza uma característica diferenciada, porém todos eles caminham em direção a um ponto comum. Logo, alguns deles foram selecionados e serão mencionados procurando atingir os aspectos mais relevantes ao tema abordado neste trabalho.

Inicialmente, PRESSMAN (2006, p. 16) expõe seus conhecimentos sobre a Engenharia de Software organizando-os em camadas, onde todas são estruturadas com foco na qualidade do produto de software. Nesse modelo o alicerce é a camada correspondente ao *processo* de software, funcionando como um “adesivo que mantém unidas as camadas de tecnologia e permite o desenvolvimento racional e oportuno de softwares de computador”. As outras duas camadas são: *métodos*, que compreendem as técnicas de “como fazer” para construir software e; as *ferramentas* que oferecem apoio automatizado ou semi-automatizado para o *processo* e para os *métodos*. Adicionalmente, SOMMERVILLE (2001, p. 43), afirma que a melhoria e padronização no processo de software pode ser implementada em várias etapas, trazendo coesão aos objetivos das atividades de uma organização. Essa prática traz melhoria na comunicação entre os integrantes da equipe, redução no tempo de treinamento de um novo integrante da equipe e aumenta a economia na automação do processo.

Segundo MACHADO e WEBER (2001) a globalização da economia vem influenciando as empresas produtoras e prestadoras de serviços de software a alcançar o patamar de qualidade e produtividade internacional para enfrentarem a competitividade cada vez maior. A norma internacional NBR ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de Software (ISO/IEC 12207, 2008) é usada como referência em muitos países, inclusive no Brasil, para alcançar esse diferencial competitivo.

3. MICRO E PEQUENAS EMPRESAS DE SOFTWARE

3.1 O MERCADO

Como visto anteriormente, o mercado brasileiro de software é muito vasto e está em constante desenvolvimento. Segundo reportagem exibida no site do SEBRAE (BONNER, 2009), foi o setor brasileiro que mais cresceu durante a última crise que gerou instabilidade e dor de cabeça para muitos países. Apesar de o Brasil ser um país com elevada carga tributária, o governo tem lançado incentivos fiscais como redução de IPI sobre *hardware*, regulado pela Lei 8.248/91 e alterações posteriores, principalmente as leis 10176/2001 e 11.077/04. UENO (2007, p. 10) mostra que, em relação ao software, as propostas governamentais de redução de impostos ainda precisam ser bem melhoradas para se tornarem atrativas para o setor.

Em se tratando de processo, a Sociedade Brasileira para a Promoção da Exportação de Software (SOFTEX), apoiada por várias instituições, entre elas o Ministério da Ciência e da Tecnologia (MCT), criaram em 2003 um modelo de referência para Melhoria do Processo de Software Brasileiro (MPS.BR), adaptado para a realidade e a cultura local e dentro dos padrões internacionais como ISO/IEC 12.207, ISO/IEC 15.504 e CMMI-DEV. Trazendo mais credibilidade e segurança para os produtos de software desenvolvidos no Brasil.

PERFIL DAS EMPRESAS

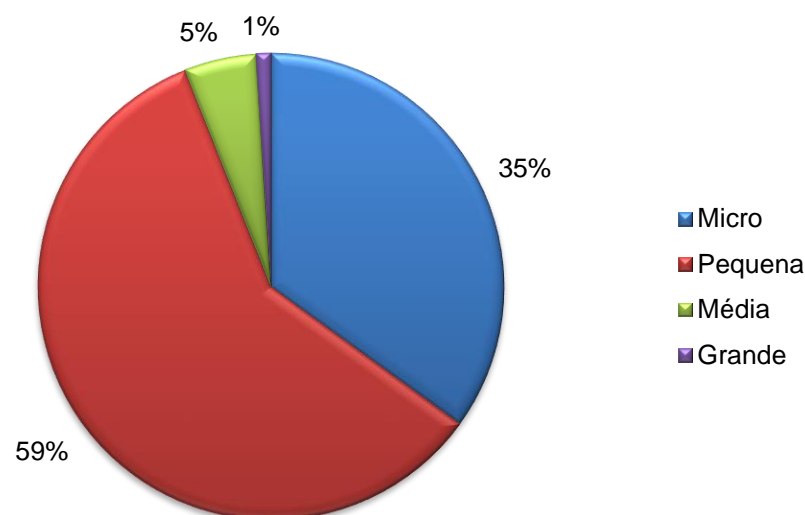


GRÁFICO 1 – DISTRIBUIÇÃO DO MERCADO BRASILEIRO DE SOFTWARE DE ACORDO COM A CLASSIFICAÇÃO DAS EMPRESAS

Segundo a pesquisa da Abes (FILHO, HOCHSTETLER, *et al.*, 2006, p. 21), existiam cerca de 7,7 mil empresas atuando no desenvolvimento, produção e distribuição de software e prestação de serviços. Deste total, 54% são de empresas de distribuição, 22% de serviços relacionados e 24% atuam na produção de software. Além disso, como mostra o Gráfico 2.1, as empresas que empregam menos de 50 funcionários (micro e pequenas) equivalem a 94% do mercado.

3.2 ORGANIZAÇÃO

A organização das micro e pequenas empresas de software do país é também algo muito importante e deve ser levado em consideração sob vários aspectos. O principal deles é a organização dos recursos humanos e, dependendo de como acontece na prática, pode trazer sérios problemas que influenciam diretamente nas atividades. Esta seção fará uma explanação sobre como esses aspectos acontecem no dia-a-dia.

3.3. COLABORADORES

Segundo o SEBRAE (2005), desde 1999 o critério adotado para conceituar micro e pequena empresa de comércio ou serviço é a receita bruta anual, ou o número de funcionários, cujos valores foram atualizados pelo Decreto nº 5.028/2004, de 31 de março de 2004, segundo descrito na Tabela 2.1. Ela não é um padrão nacional, serve apenas como referência¹ para as economias locais (Unidades da Federação).

Classificação	Renda bruta anual	Número de funcionários
Microempresa	Menor que 433.755,14	Menor ou igual a 9
Pequena empresa	Entre 433.755,14 e 2.133.222,00	Entre 10 e 49

TABELA 1 - CLASSIFICAÇÃO DAS MICRO E PEQUENAS EMPRESAS DE SOFTWARE BRASILEIRAS SEGUNDO O SEBRAE

¹ Cada um dos estados brasileiros reajusta a sua tabela de acordo com as características do mercado local.

Observando o mercado local, é muito difícil para uma microempresa (tendo um quadro reduzido de pessoal) adotar a um processo de desenvolvimento de software, sem que uma pessoa não acumule papéis. Isso é comum no mercado. A maioria das empresas prefere dar mais papéis para os colaboradores do que contratar novos funcionários. Isso é uma característica observada e tratada pelo Processo proposto. Ele propõe a distribuição de papéis entre um número mínimo de pessoas, julgado através do interesse (artefatos e responsabilidades) do papel nas atividades do processo.

3.4 PROBLEMAS DAS EMPRESAS NA GESTÃO DE TI

O principal problema enfrentado pela empresa (ainda não aberta oficialmente) que desenvolve o Sistemas é a organização das atividades para o desenvolvimento e um bom produto de software. Assim como nas empresas brasileiras de software, ela possui conhecimento técnico sobre várias tecnologias. Dessa forma, a equipe de desenvolvimento corre o risco de se confundir na hora de desenvolver as atividades de produção do software, porque os integrantes dela não sabem em que momento deve-se especificar qual parte do software produzido e que atividade deve ser mais intensa para garantir que o problema do cliente será resolvido de fato.

Assim, observa-se que o principal problema não está no conhecimento ou aprendizado da equipe, e sim, na sistematização das atividades que levam a construção de um produto de software viável para as duas pontas (empresa e cliente).

4. PROCESSOS E GERÊNCIA DE SOFTWARE

4.1. DEFINIÇÃO DE PROCESSO

Por ser um termo com ampla utilização nas diversas áreas do conhecimento, faz-se necessário um direcionamento do seu significado ao tema em questão. A primeira definição é abrangente o bastante para explicar o conceito. Logo, um processo pode ser definido como uma série de ações na qual uma ou mais entradas são utilizadas para produzir uma ou mais saídas (AMBER, 1998). Característica bastante visível nas atividades do Processo proposto.

4.2 DEFINIÇÃO DE PROCESSO DE SOFTWARE

Assim, pensando em Engenharia de Software, define-se como um conjunto de passos parcialmente ordenados visando atingir uma meta que, neste caso, é entregar um produto de software com qualidade, dentro dos prazos e custos estabelecidos. Diversos autores renomados tentam definir processo de software enfatizando os aspectos mais importantes. Sendo assim, SOMMERVILLE (2001, p. 44) define como sendo um conjunto de atividades direcionadas à construção de um produto de software. Esta definição pode ser ainda mais abrangente aplicando as seguintes palavras de PRESSMAN:

Um processo de software é base para o controle gerencial de projetos de software e estabelecimento do contexto no qual os métodos técnicos são aplicados. Os produtos de trabalho (modelos, documentos, dados, relatórios, formulários, etc.) são produzidos, os marcos são estabelecidos, a qualidade é assegurada e as modificações são adequadamente geridas (2006, p. 17).

Observa-se facilmente que todos eles possuem um mesmo referencial: atingir o objetivo de entregar um produto de software com qualidade e atendendo as expectativas do cliente. Quando o objetivo é alcançado, todos os envolvidos no processo estarão satisfeitos, tanto a equipe, por desenvolver um bom trabalho, quando o cliente, por ter um sistema confiável e financeiramente acessível.

5. MODELOS DE PROCESSOS

O estudo de modelos de processos de software é interessante para observar como a organização das atividades faz a diferença no momento de se produzir um software. Por isso, os tópicos seguintes abordam os modelos cascata, iterativo, incremental, espiral e traz exemplos de metodologias ágeis de desenvolvimento, tentando explicar como eles abordam e organizam suas propostas de processos de software e apresentando seus pontos positivos e negativos.

5.1. MODELO EM CASCATA

É o paradigma mais antigo da engenharia de software, sendo, algumas vezes, chamado de ciclo de vida clássico (PRESSMAN, 2006, p. 39). É um modelo de desenvolvimento de software sequencial no qual o desenvolvimento é visto como um fluir constante para frente (como uma cascata) através das fases de análise de definição dos requisitos, design do software, implementação e teste de unidade, integração e teste do sistema, operação e manutenção, conforme demonstrado na Figura a seguir:

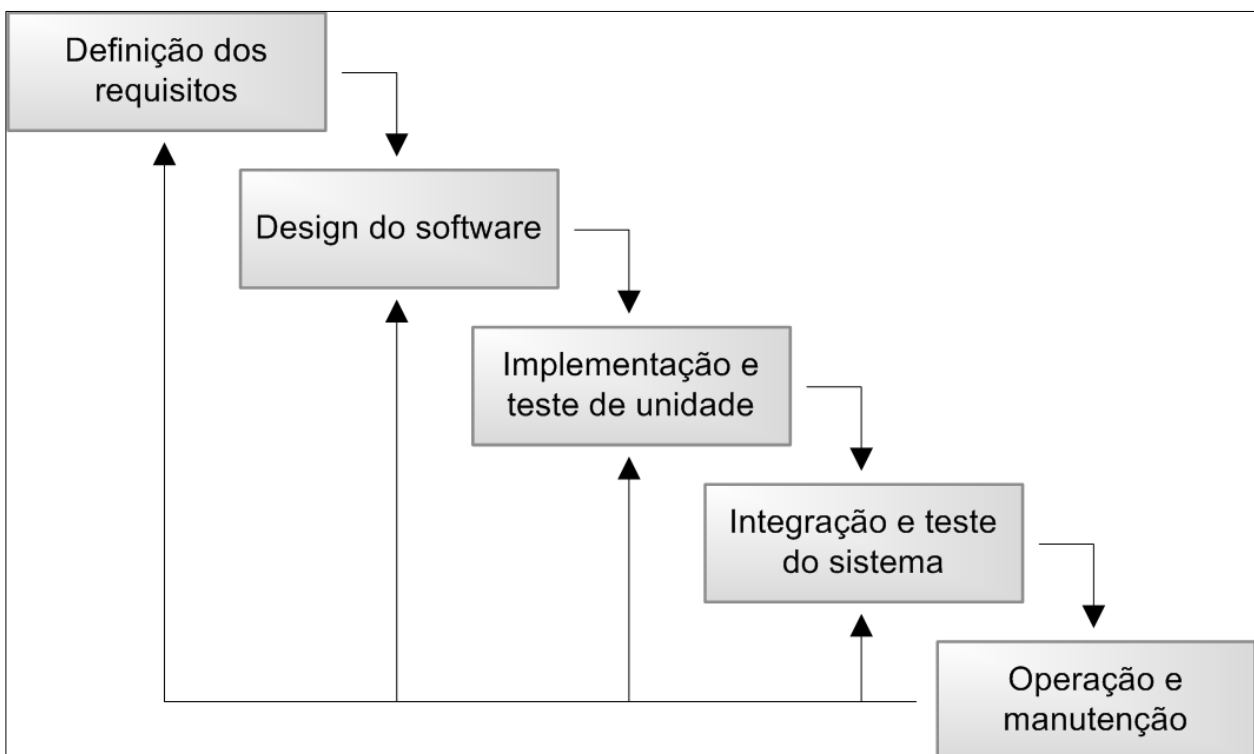


FIGURA 2 - CICLO DE VIDA DO MODELO SEQUENCIAL (CASCATA)

A origem do termo *cascata* é frequentemente citado como sendo um artigo publicado em 1970 por W. W. Royce.

Ironicamente, Royce defendia uma abordagem iterativa para o desenvolvimento de software e nem mesmo usou o termo *cascata*. Ele, originalmente, descreve o que é hoje conhecido como o modelo em *cascata* como um exemplo de um método que ele argumentava ser um risco e um convite para falhas (DUTRA, 2008).

5.2. MODELO ITERATIVO E INCREMENTAL

Desenvolvimento incremental é uma estratégia de planejamento em que várias partes do sistema são desenvolvidas em versões, como na Figura 3.2, ou em paralelo, sendo integradas quando completas. Não implica, requer ou pressupõe desenvolvimento iterativo ou em cascata – ambos são estratégias de retrabalho.

A alternativa ao desenvolvimento incremental é desenvolver todo o sistema com uma integração única. Desenvolvimento iterativo é uma estratégia de planejamento em que o tempo de revisão e melhorias de partes do sistema é pré-definido. Isto não pressupõe desenvolvimento incremental, mas funciona muito bem com ele. Uma diferença típica é que a saída de um incremento não é necessariamente assunto de um refinamento futuro, e seu teste ou retorno do usuário não é utilizado como entrada para planos de revisão ou especificações para incrementos sucessivos. Ao contrário, a saída de uma iteração é examinada para modificação, e especialmente para revisão dos objetivos das iterações sucessivas.

5.3. DIFERENCIANDO OS MODELOS EM CASCATA, ITERATIVO E INCREMENTAL

Para firmar os conceitos dos modelos apresentados acima, foi esquematizada uma pequena comparação entre elas na Figura 3.2. Têm-se como exemplo, três partes que juntas compõem o software desejado. Assim, no modelo sequencial ou cascata, todos os software é trabalhado e desenvolvido de uma só vez.

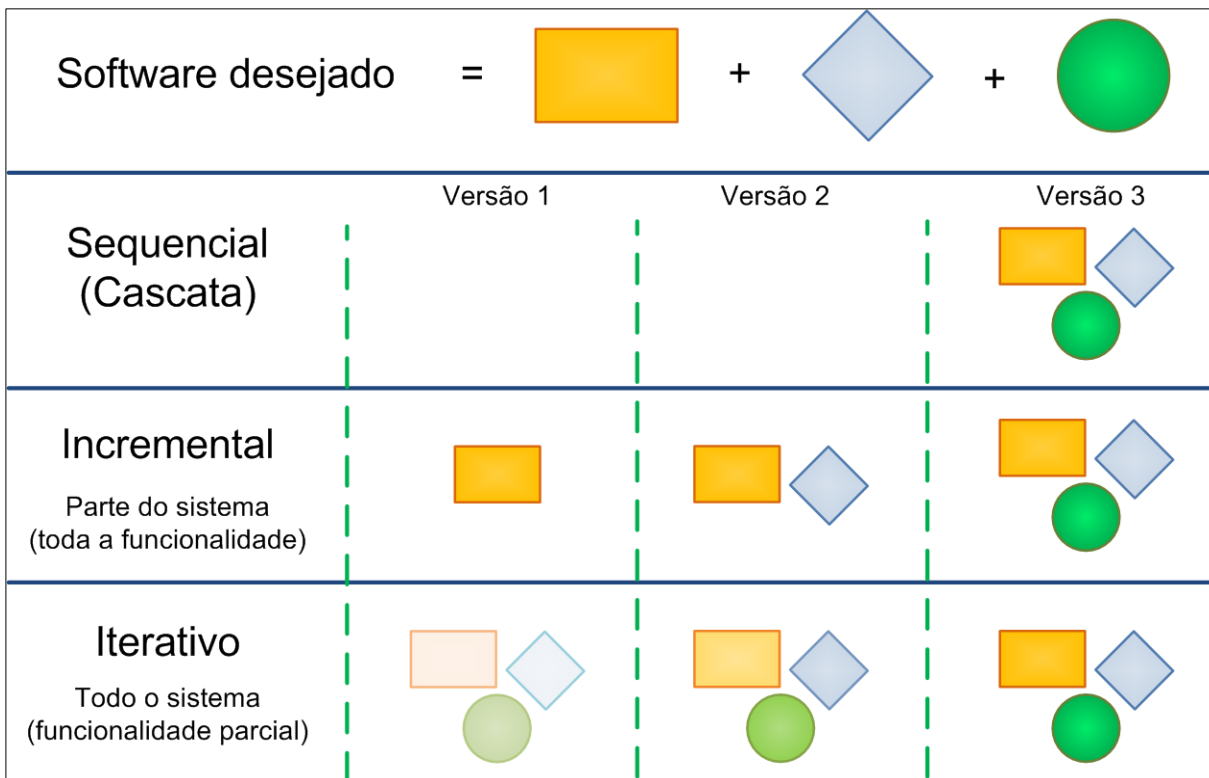


FIGURA 0 - COMPARAÇÃO ENTRE OS MODELOS EM CASCATA, INCREMENTAL E ITERATIVO

Observa-se que o modelo iterativo lança versões de funcionalidades parcialmente implementadas. Na abordagem Incremental, são desenvolvidas as funcionalidades por completo, uma por uma, e lançadas incrementalmente em versões até que a última apresente a solução desejada pelo cliente.

O desenvolvimento sequencial apresenta um aspecto que não é muito interessante para o mercado de software atual. Como se pode ver, o software é especificado, desenvolvido, testado e integrado de forma completa e implantado de uma só vez. Assim, os desenvolvedores não conseguem acompanhar as necessidades do cliente com as regras de negócio dele mudando rapidamente, devido ao dinamismo do mercado. Sendo assim, ao praticar o desenvolvimento de software com essa abordagem, é provável que o ele não consiga atender as necessidades do cliente desde a sua concepção. Porém, esta forma de desenvolver software facilita o aprendizado e entendimento, pelo fato de que as atividades estão organizadas sequencialmente.

A vantagem do modelo incremental, em relação ao sequencial, é que ele possibilita a inserção de funcionalidades de cada vez. Assim, o cliente começa a utilizar uma parte da solução de cada vez, solicitando mais funcionalidades ou módulos para desenvolvimento e integração. A desvantagem desse modelo é que, uma vez definida a funcionalidade e o desenvolvimento é iniciado, o cliente só poderá solicitar alterações após o desenvolvimento

dela. Podendo, em alguns casos, causar o desenvolvimento errado de uma funcionalidade, devido à uma má interpretação dos requisitos ou se o cliente não tiver se expressado da forma correta.

A iteratividade por si só já é interessante, ou seja, criar uma funcionalidade de forma iterada permite ao cliente expressar sua necessidade de forma fragmentada e analisada, observando os equívocos e mudanças ocorridas no ambiente do cliente. Isso torna o produto de software mais adequado às necessidades do cliente. Em contrapartida, essa iteração no desenvolvimento pode levar o cliente a atrapalhar o andamento do projeto, com a inserção desordenada de funcionalidades.

5.4. MODELO EM ESPIRAL

Este modelo tem o objetivo de prover um “metamodelo” que pode acomodar diversos processos específicos. Isto significa que podemos encaixar nele as principais características dos modelos vistos anteriormente, adaptando-os às necessidades específicas de desenvolvedores ou às particularidades do software a ser desenvolvido. Este modelo prevê prototipação, desenvolvimento evolutivo e cíclico, e as principais atividades do modelo cascata.

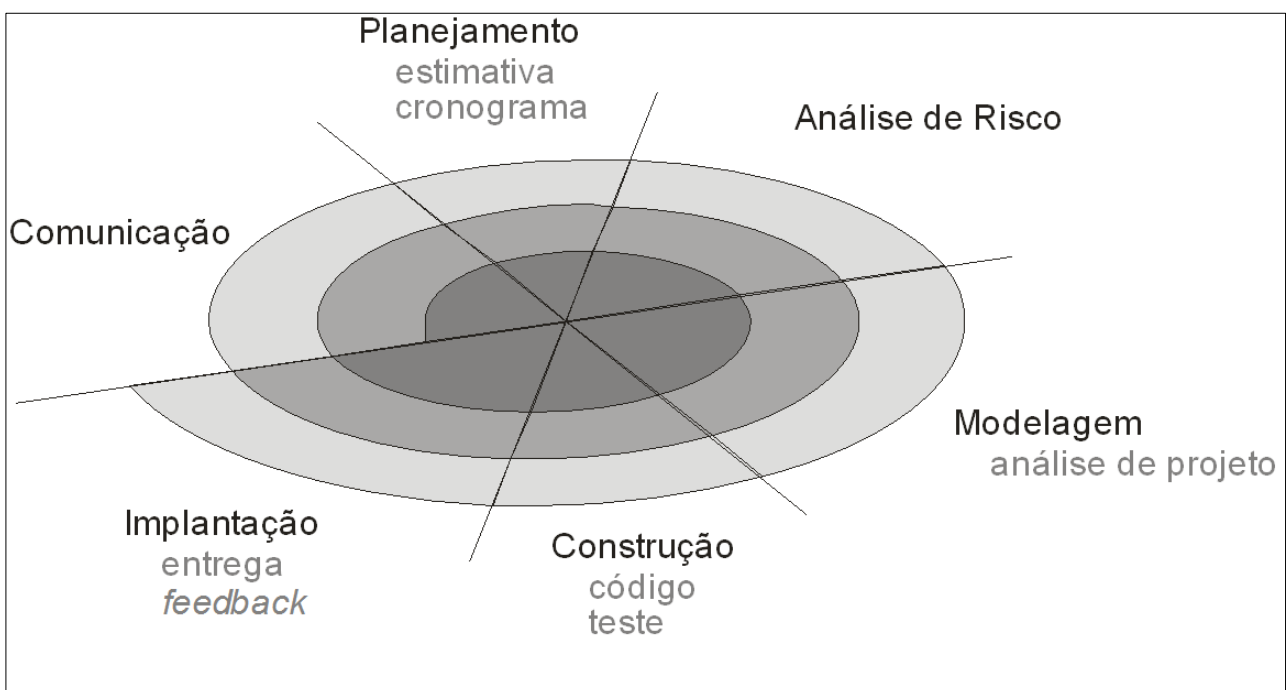


FIGURA 4- MODELO DE PROCESSO EM ESPIRAL (PRESSMAN, 2006, p. 45).

Sobre esse modelo SIMÃO & VARELA dizem:

O modelo em espiral foi proposto por Boehm em seu artigo de 1988 *A Spiral Model of Software Development and Enhancement*, como forma de integrar os diversos modelos existentes à época, eliminando suas dificuldades e explorando seus pontos fortes. Este modelo foi desenvolvido para abranger as melhores características tanto do ciclo de vida clássico como da prototipagem, acrescentando, ao mesmo tempo, um novo elemento - a análise de riscos - que falta a esses paradigmas (SIMÃO e VARELA, 2009, p. 10).

Um ponto a ser observado nesse modelo é a análise de risco que, segundo DUTRA (2008), exige muita experiência da equipe. Além disso, pode ser difícil convencer os clientes que a abordagem evolucionária é controlável. Ela exige competência considerável na avaliação de riscos e depende dessa competência para ter sucesso. Se um risco importante não for descoberto e gerenciado, fatalmente ocorrerão problemas.

5.5. MODELO ÁGIL

Os processos em desenvolvimento ágil de software parecem ser mais eficientes do que as metodologias antigas. Porém, mesmo utilizando menos tempo do programador no desenvolvimento de softwares funcionais de alta qualidade, tem a desvantagem de ter uma perspectiva de negócio que não provê uma capacidade de planejamento em longo prazo. Em essência, eles proveem mais funcionalidades por custo/benefício. Existem várias metodologias que podem ser consideradas como abordagens ágeis, entre elas: Scrum, Programação Extrema (XP), FDD, Crystal Clear, DSDM entre outras (Processo de Desenvolvimento de Software, 2009). As metodologias ágeis não serão definidas por estarem fora do contexto deste trabalho.

6. O PROCESSO PROPOSTO

Como é parte dos objetivos desse trabalho, esta seção pretende apresentar uma proposta de processo de desenvolvimento de softwares para utilização em micro e pequenas empresas tecnologia da informação. Antes que seus detalhes sejam citados, uma forma de visualização do processo é apresentada com o intuito de deixar o leitor bem posicionado em termos de qual característica e nível de detalhamento ele está observando, além de proporcionar um panorama completo da organização do Processo proposto. Outra estratégia adotada para melhorar o entendimento do conteúdo apresentado aqui foi à definição dos papéis antes que as atividades sejam apresentadas e detalhadas. Assim, o leitor compreenderá melhor o objetivo de cada uma delas e entenderá a escolha dos papéis dos artefatos necessários para a realização da atividade e dos artefatos gerados como saída.

6.1. NÍVEIS DE DETALHAMENTO DO PROCESSO

Um processo de desenvolvimento de software pode ser observado sobre diversos aspectos. Uma forma de abordá-lo (a fim de entender a sua organização) é buscando uma visão em níveis de detalhamento. Pensando nisso, esse tópico tenta explicar o Processo proposto em três níveis: Ciclo de vida, Atividades e Detalhamento, mostrados na Figura abaixo:

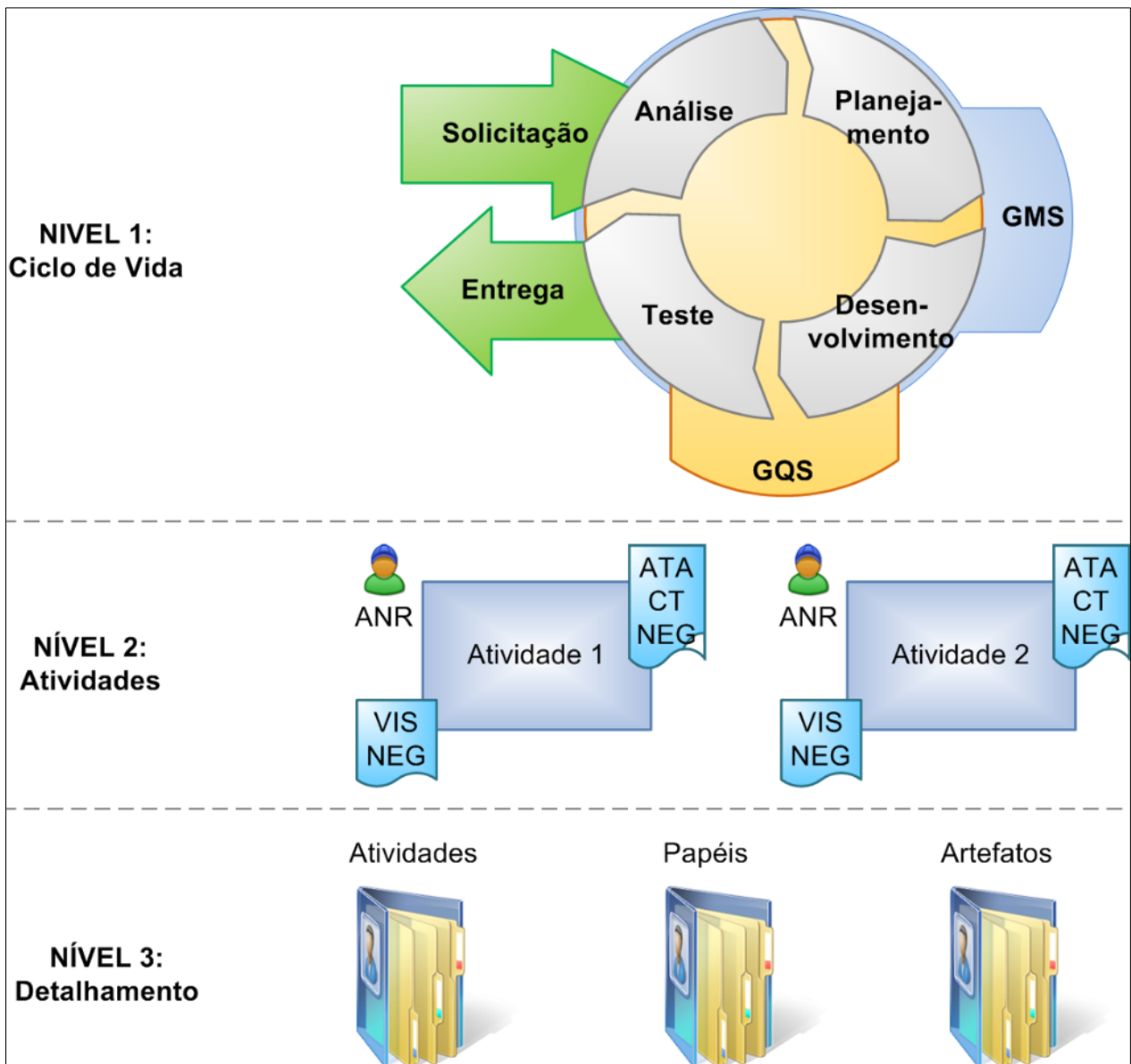


FIGURA 5 - VISUALIZAÇÃO DO PROCESSO PROPOSTO EM NÍVEIS

No nível mais alto de abstração dos detalhes está o Ciclo de Vida do processo, onde se especifica como estão organizadas as fases. Durante a definição deste nível, pensou-se como seriam melhor utilizados dos modelos de processos definidos pela Engenharia de Software. Note que foram aplicados aspectos do modelo iterativo e incremental ao Processo proposto, ou seja, no ciclo podem ser “lançados” pacotes de funcionalidades que após serem desenvolvidos e integrados representam uma solução.

O segundo nível possui a representação² das atividades compostas de: nome da atividade, participantes, artefatos de entrada e artefatos de saída. Outra característica importante expressa nesse nível é o relacionamento entre as atividades. Assim os integrantes

² Ciclo de vida e atividades, você encontrará o detalhamento deste trabalho.

da equipe sabem exatamente em que atividades eles possuem participação, quais artefatos necessários para iniciar a atividade e quais artefatos gerados como saída.

Por fim, no terceiro nível, estão todos dos detalhamentos necessários para a execução bem sucedida do processo. São eles:

- **Detalhamento das atividades** – para cada uma delas é definida uma visão geral, papéis responsáveis e participantes, artefatos de entrada e saída e uma visão detalhada, contendo passos principais e alternativos;
- **Definição dos papéis** – para cada um deles é realizada uma breve descrição, listadas as atribuições, artefatos que são de responsabilidade do papel, títulos exigidos, títulos desejados e as características pessoais esperadas da pessoa candidata ao papel;
- **Definição dos artefatos** – nesse nível, são disponibilizados os modelos dos artefatos necessários para controle e execução do processo. A estrutura organizacional deles pode variar de acordo com o objetivo.

6.2. NÍVEL DO CICLO DE VIDA

Para dar início a este tópico é importante ressaltar a diferença entre ciclo de vida de processo e ciclo de vida de software. São dois conceitos bem parecidos e que causam confusão em muitas pessoas. Assim, o ciclo de vida de um processo descreve as fases e atividades pelas quais um projeto de software passa até ser gerada uma nova funcionalidade ou solução, a partir de uma necessidade expressa em forma de solicitação. Diferente dessa definição, o ciclo de vida de um software inicia a partir da concepção do software, passando por um processo de desenvolvimento, seguindo pela sua utilização e evolução até que ele não atenda as necessidades do seu utilizador e caia em desuso.

Tomando como base esses conceitos, foi definido o Processo proposto dividido em fases do processo principal e processos de apoio.

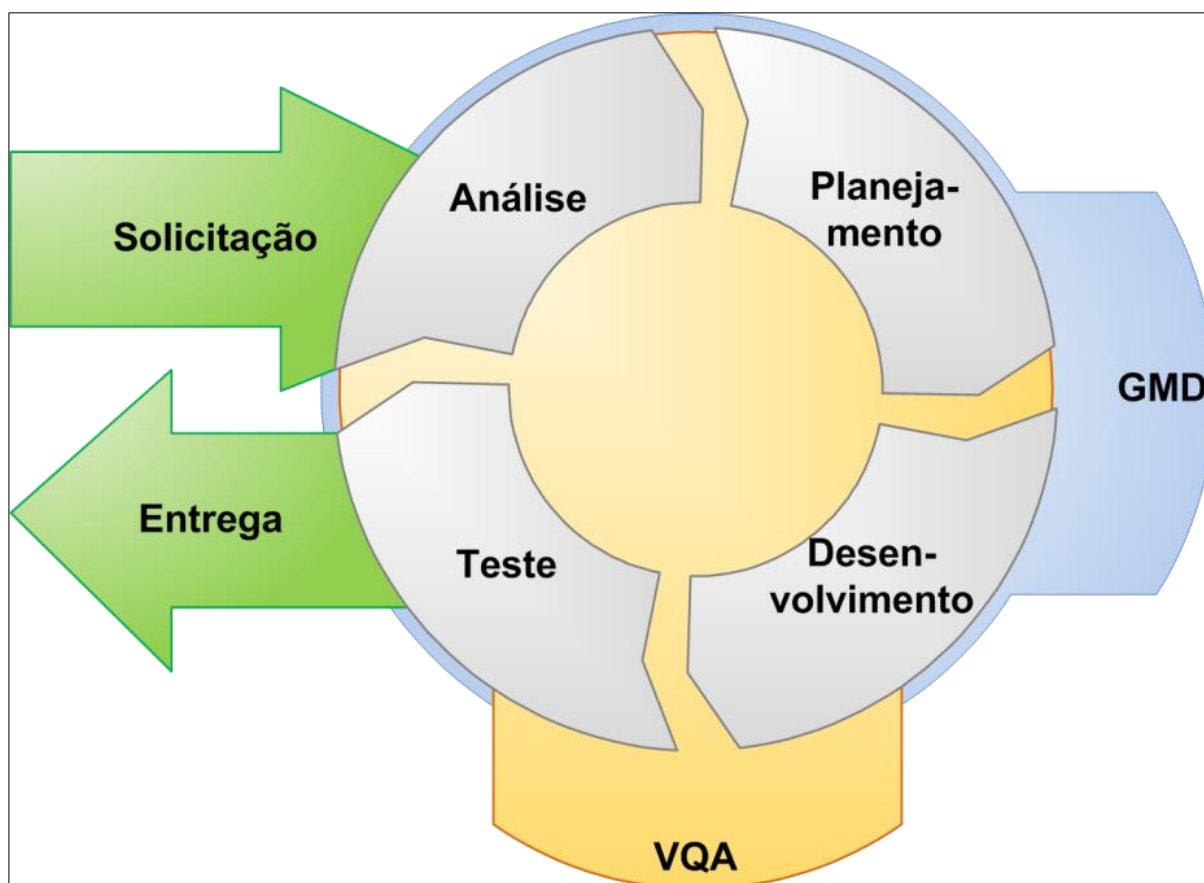


FIGURA 6 - CICLO DE VIDA DO PROCESSO PROPOSTO

Assim, como podemos observar na Figura acima, o Processo proposto foi definido e separado em fases que descrevem as entradas e saídas do ciclo de desenvolvimento (Solicitação e Entrega), o próprio ciclo (Análise, Planejamento, Desenvolvimento e Teste) e processos de apoio que são o acompanhamento para gerência da qualidade (VQA) e a gerência de mudanças (GMD). Cada uma delas abordando o desenvolvimento do software sobre uma visão diferente de acordo com os seus interesses. A seguir, cada uma das fases será explicada para que haja maior compreensão das suas abordagens e características.

Antes de iniciar as descrições das fases do ciclo de vida do Processo proposto, será realizada uma breve definição de do que significa o termo *fase* dentro do escopo deste trabalho. Assim, *fase* é a representação de um conjunto de atividades organizadas com um objetivo comum de desenvolver uma parte significativa de um projeto de software.

Geralmente elas são dependentes entre si em alguns aspectos. Os artefatos são um exemplo, ou seja, numa fase, geralmente são usados artefatos criados e desenvolvidos em outras fases. Segue algumas fases:

FASE DE SOLICITAÇÃO

Como observada na Figura 4.2, esta é a fase de entrada no processo. Ela descreve como o cliente pode expressar suas necessidades perante a equipe de desenvolvimento e como eles devem observar o problema em busca da melhor solução. Em linhas gerais, o objetivo dessa fase é identificar o problema, entender o ambiente do cliente e lançar uma proposta de solução. Então, a solução só será encaminhada para a próxima fase depois de autorizado pelo cliente e o mesmo assinar um contrato prévio.

FASE DE ANÁLISE

Aqui serão utilizados os meios mais apropriados da Engenharia de Software para especificar as necessidades do cliente em forma de requisitos e casos de uso, em um nível de detalhamento aceitável para a equipe de desenvolvimento e para o projeto. Para chegar a esse nível de aceitação são criados vários tipos de especificações (incluindo os diagramas e protótipos) que descrevem as necessidades do cliente e a solução proposta pela equipe de desenvolvimento. Ao final desta fase, estarão prontas todas as especificações dos requisitos, ou seja, o problema do cliente estará totalmente entendido e a solução modelada.

Uma característica muito importante dessa fase é a constante participação do cliente na maioria das atividades, na tentativa de garantir um melhor entendimento dos problemas e necessidades dele. Dessa forma, o cliente estará interagindo com a equipe de desenvolvimento, tirando dúvidas, descrevendo necessidades e ajudando a moldar a solução. Essa é uma estratégia que tenta garantir o aumento da interação e comunicação entre o cliente e a empresa, criando uma relação mais fortalecida e diminuindo a sensibilidade do cliente perante as renegociações de prazo e custo (havendo necessidade).

FASE DE PLANEJAMENTO

A fase de planejamento é uma fase de preparação e gerenciamento, onde são definidos os planos para execução das próximas fases. Nesta fase, é necessário avaliar o ambiente de desenvolvimento disponível para o desenvolvimento da solução. Dependendo das exigências do projeto e do cliente ou até mesmo dos requisitos, o ambiente deve ser ajustado para

acomodar satisfatoriamente a construção da solução. Durante a elaboração de um Plano de Projeto³, devem ser levados em consideração vários aspectos:

- Observar a *capacidade da equipe* em trabalhar com a tecnologia necessária para desenvolver a solução;
- Analisar a *necessidade de treinar a equipe* ou contratar pessoal já qualificado, para isso, fazendo um estudo e verificando a viabilidade de custo e prazo das opções disponíveis;
- Organizar o ambiente da empresa, *alocando recursos humanos e materiais* de forma que possibilite maior comodidade e agilidade da equipe de desenvolvimento.

É importante observar e ponderar outras características que podem surgir em cada ambiente onde o processo é empregado ou em cada tipo de solução desenvolvida. Além disso, é necessário identificar todos os riscos que podem vir a modificar o andamento do projeto em questão e tomar as medidas cabíveis para evitá-los ou gerenciá-los de maneira que o projeto sofra menos impacto.

FASE DE DESENVOLVIMENTO

No ciclo de vida do Processo proposto, está corresponde à terceira fase. Aqui serão definidas as especificações arquiteturais, banco de dados e codificação da solução. Um exemplo de especificação arquitetural é: atualmente a maioria dos projetos é definida seguindo o paradigma de Programação Orientada a Objetos (POO). Assim, sua arquitetura é constituída de classes e organizada em pacotes de funcionalidades. Além disso, são criados vários diagramas utilizando UML para mostrar o relacionamento entre as classes dentro dos pacotes.

Outra especificação importante nesta fase é a definição do esquema de banco de dados para a solução, sendo, para isso, desenvolvidos Diagramas Endidade-Relacionamento (DER), em seguida os Diagramas Entidades-Relacionais (ER) e, por fim, definição de um dicionário de dados. Estes podem fazer parte de um novo banco de dados ou serem integrados a outro.

A terceira e última atividade desta fase é a codificação. É nesse passo que são traduzidas todas as especificações arquiteturais e de banco de dados para uma linguagem de programação, resultando no código fonte ou *build* da solução.

FASE DE TESTE

³ Consulte as faases do projeto para obter mais detalhes.

Esta fase é responsável por definir como serão realizados todos os testes necessários para garantir o bom funcionamento e aceitação da nova solução de software, que poderá ser integrada a outra solução existente ou um novo software. Alguns dos testes propostos são descritos abaixo:

1. Teste de integridades dos dados;
2. Teste de interface com o usuário;
3. Teste de segurança;
4. Teste de funcionamento;
5. Teste de aceitação;
6. Teste de integração.

De forma geral, após a realização de todos os testes sobre as funcionalidades ou soluções, em caso de aprovação, elas devem seguir para a próxima fase. Caso sejam identificadas irregularidades, serão encaminhadas (juntamente com relatório de irregularidades) para a fase e atividade responsável por corrigir o erro detectado. Para complementar o entendimento sobre as atividades desta fase, o leitor pode consultar os Apêndice A – Ciclo de vida e mapa de atividades.

FASE DE ENTREGA

Esta fase, é a fase de saída da funcionalidade do ciclo de vida do processo. Nela são desenvolvidos os planejamentos para treinamento dos usuários e implantação da solução. Os cursos preparatórios para que os usuários aprendam a utilizar a solução antes da utilização serão desenvolvidos. Após o treinamento, a nova solução já poderá ser implantada, podendo ser necessário um acompanhamento da utilização durante alguns dias, caso o cliente solicite.

Nessa fase, o acompanhamento é importante para que se tome conhecimento de como a solução está sendo encarada na prática. Além disso, é importante observar qual o comportamento dos usuários em relação à nova funcionalidade e se esta está desempenhando satisfatoriamente sua função.

6.3. VERIFICAÇÃO DA QUALIDADE

A fase de verificação da qualidade ou validação faz parte de um dos processos de apoio do Processo proposto, sendo realizado sempre nas transições entre as fases principais do ciclo de vida do processo. O objetivo dele é garantir a qualidade do produto de software que está sendo criado, sendo necessária, para isso, uma verificação na qualidade dos principais produtos de software criados. Além disso, a verificação também acompanha a qualidade do processo como um todo a partir de uma análise da quantidade de não-conformidades em comparação com o cumprimento dos prazos estabelecidos para o projeto. Logo após as devidas correções são apresentadas aos responsáveis e a atividade segue o curso normal.

Os papéis responsáveis pelas atividades durante esta fase são o Gerente de Projeto e Analista de Qualidade. O primeiro irá avaliar como está o andamento do projeto: se está em dia, se os riscos estão sendo controlados devidamente, etc. O segundo irá avaliar os artefatos gerados durante a fase, de forma a solicitar alterações nos mesmos, caso eles não estejam no padrão estabelecido pela empresa.

Em seguida haverá uma reunião entre os dois, onde o Analista de Qualidade passará ao Gerente a sua avaliação para que ele possa ter um maior controle do andamento do projeto, já que pode haver a necessidade de correção de alguns documentos.

7 A GERÊNCIA DO PROJETO

7.1. OS PAPÉIS, RESPONSABILIDADES E ARTEFATOS

Para que uma empresa possua uma organização de atividades eficiente, ela precisa definir “quem faz o que”. Por isso, antes da apresentação das atividades, todos os papéis serão definidos, expondo suas responsabilidades, atribuições e apontando suas características mais importantes.

É importante frisar nesse momento sobre a terminologia utilizada nas próximas seções para denominar os papéis e artefatos definidos aqui. Assim, quando forem citados nomes de papéis ou artefatos, eles terão a primeira letra escrita em maiúsculo. Por exemplo: a diferença entre “Cliente” e “cliente” é que o primeiro faz referência o papel definido no Processo proposto e o segundo representa uma pessoa qualquer que procura uma empresa em busca de soluções (produtos ou serviços).

7.2. CLIENTES

Este papel representa a pessoa que entende a regra de negócio da empresa a qual se destina a solução podendo ser representado por uma ou mais pessoas, dependendo do projeto e da necessidade. Nesse caso, ele pode ser dividido em dois perfis: Cliente Interno e Cliente Externo, onde cada uma das pessoas apresentará características e responsabilidades pertencentes a um ou outro perfil. Em todo caso, eles possuirão as mesmas atribuições.

O primeiro perfil analisado é o Cliente Interno representado por um colaborador da empresa de desenvolvimento. Geralmente este perfil pode possuir poucos conhecimentos técnicos em informática. Porém, possui boas habilidades para entender regras de processos de negócios de outras empresas e sabe persuadir o cliente em busca de descrições mais precisas das necessidades dele. Esse perfil se responsabiliza pelo desenvolvimento de alguns artefatos do projeto.

- O primeiro artefato é o Glossário⁴, que é um documento onde estão reunidos todos os termos incomuns, tanto para o cliente quanto para a equipe. O objetivo dele é definir um vocabulário compartilhado entre os *stakeholders* (envolvidos) do projeto, a fim de unificar o entendimento para que não haja ambiguidades. No Processo proposto ele é usado para especificar, de forma unificada, os termos comuns ao Cliente e a equipe.

Pode ser dividido em: Glossário de Negócios (termos comuns aos negócios do Cliente) e Glossário de Projeto (termos comuns à equipe do projeto).

- O outro artefato é denominado como Regras de Negócio. Nele o *Cliente Interno* especifica detalhadamente todas as regras de negócio, e que fazem parte do domínio do problema. Para a especificação poderiam ser usados cenários⁵ de uso. Este é um documento muito importante para a compreensão do problema e, principalmente, para o controle das mudanças, já que, na maioria das vezes, o ambiente do cliente é dinâmico e suas regras de negócio mudam com facilidade.

O segundo perfil é o do *Cliente Externo* incorporado por uma pessoa externa à empresa de desenvolvimento. Geralmente representado pela pessoa que procura a solução ou alguém indicado por ele com capacidade para responder às questões problemáticas enfrentadas. Este perfil de cliente pode apresentar pouco ou quase nenhum conhecimento sobre computadores de um modo geral.

- Sobre artefatos, o *Cliente Interno* é responsável por todos os documentos fornecidos por ele para definição e entendimento do problema. Esses artefatos podem ser dos mais variados tipos: relatórios, planilhas, anotações, desenhos, documentos, etc.

Até o momento foram analisados esses dois perfis de Cliente que se encaixam no Processo proposto. Porém, nada impede que outros perfis sejam observados e considerados.

7.3. ANALISTA DE REQUISITOS

O papel de Analista de Requisitos representa a pessoa que consegue observar o ambiente do Cliente identificando os problemas e propondo soluções viáveis e interessantes para o ele. O Analista de Requisitos também deve possuir habilidades para persuadir o Cliente deixando-o a vontade para expor suas idéias e necessidades, independente do perfil de Cliente considerado.

Como observado, o Analista de Requisitos possui atribuições de comunicação com o Cliente e desempenha atividades importantes para a empresa. Por isso, ele é responsável pelo artefato Documento de Visão de Negócio. Neste documento estão contidas informações relevantes, como as vantagens que o cliente terá caso ele contrate a solução. Além disso, esse documento especifica em detalhes o ambiente do cliente e identifica quais os possíveis usuários e envolvidos no projeto, mostrando para cada um deles, as características,

⁵ Um cenário é uma narrativa, textual ou pictórica, de uma situação (de uso de uma aplicação), envolvendo usuários, processos e dados reais ou potenciais (REZENDE, 2003).

atribuições e necessidades. Adicionalmente, são elicitados todos os problemas enfrentados e as possíveis soluções para eles.

Outra atividade desempenhada pelo Analista de Requisitos é a identificação, definição e especificação de requisitos em casos de uso. Em consequência disso, ele se torna responsável pelos artefatos Documento de Elicitação de Requisitos e Documento de Especificação em Casos de Uso.

No Documento de Elicitação de Requisitos, ele organiza os requisitos, dando um número identificador e uma descrição para cada um dos requisitos funcionais (RF – representam as funcionalidades da solução), não funcionais (RNF – representam as características da solução) e cria um escopo negativo, deixando explícito o que não faz parte da solução.

Uma forma de ele organizar os requisitos e identificar o relacionamento entre eles é criando uma Matriz de Rastreabilidade. Este artefato pode relacionar tanto RF com RF, quanto RF com RNF, sendo importante para identificar o impacto nas alterações dos requisitos. Ele também pode relacionar várias outras características de um projeto através da rastreabilidade entre elas, ou seja, a interligação existente entre características e especificações da solução. Para realizar esse controle, podem ser usadas planilhas eletrônicas, onde na linha estão expostas uma das características, e na coluna a outra. Para ter uma visão mais concreta sobre como seria esse tipo de controle, consulte o Apêndice C.9, nele há um exemplo de uma Matriz de Rastreabilidade que pode ser tomada como base para a construção sua própria matriz.

7.4. GERENTE DE PROJETOS

Uma pessoa que detém as atribuições desse papel deve possuir espírito de liderança e trabalho em equipe. Um Gerente de Projetos deve ser capaz de liderar uma equipe de projetos de forma que ela venha ter o máximo de desempenho possível. Ele deve ser capaz de gerenciar de maneira efetiva os riscos de um projeto. Para isso, ele deve possuir uma lista de possíveis riscos e junto a eles, um plano de ação preventiva ou corretiva, caso um deles venha a se concretizar, chamado plano de contingência.

Esta lista de riscos pode ser organizada da forma mais conveniente possível, de acordo com as considerações do Gerente de Projetos. Um exemplo seria organizá-la em um único documento denominado Lista de Riscos contendo os riscos e suas principais características. O documento Gestão de Ações Corretivas ajuda no controle e nas medidas a serem tomadas nos

desvios que acontecem durante o projeto, além de possibilitar que tais desvios possam ser evitados nas iterações futuras.

Outro artefato que também é de responsabilidade deste papel são os Contratos. Assim, um Gerente de Projetos também interage com o Cliente, realizando negociações e firmando os contratos. O Processo proposto não define nenhum modelo de contrato. Portanto, devem ser definidos contratos de acordo com o serviço a ser prestado ao cliente.

Em se tratando de experiência, não é necessário que ele possua conhecimento técnico na tecnologia trabalhada, embora isto adicione certa importância. Ele deve ser líder, realizar planejamentos concretos, acompanhar o trabalho e o projeto.

7.5. ARQUITETO DE SOFTWARE

Ao papel Arquiteto de Software são atribuídas à liderança e a coordenação das atividades técnicas no decorrer do projeto. O Arquiteto de Software também estabelece a estrutura geral de cada visão arquitetural do software: a decomposição da visão, o agrupamento dos elementos e as interfaces entre esses principais agrupamentos. Portanto, comparado aos outros papéis, a visão do Arquiteto de Software é ampla, e não detalhada (Rational Software Corporation, 2002).

Um dos artefatos pelo qual o Arquiteto de Software é responsável é o Documento de Arquitetura do Software que oferece diversas visões arquiteturais de representação da solução. Nele são inseridos, por exemplo:

- **Visão lógica** – representação do ponto de vista da arquitetura do modelo de *design*, dividindo em subsistemas e pacotes. Dentro de cada pacote está a representação das classes significativas do ponto de vista da arquitetura.
- **Visão de implantação** – descreve uma ou mais configurações físicas do computador ou da rede, indicando como o software utilizará esta estrutura.
- **Visão de dados** – descreve a perspectiva de armazenamento dos dados persistentes, podendo ser inseridos os diagramas DER e ER.

No Processo proposto, o Arquiteto de Software também determina como será feita a integração de uma nova funcionalidade ao sistema, definindo qual a ordem em que devem ser desenvolvidos os *builds*⁶ e quais os subsistemas que eles fazem parte.

⁶ Versão codificada da solução e que ainda não foi submetido a nenhum teste.

7.6. DESENVOLVEDOR

Este papel representa a pessoa capaz de traduzir problemas do “mundo humano” para o “mundo computacional” através de linguagens de programação. O desenvolvedor deve ter conhecimentos sólidos sobre os princípios básicos da computação, como algoritmos e estruturas de dados, além de saber aplicá-los em uma ou mais linguagens de programação. O desenvolvedor deve saber trabalhar em equipe dividir o conhecimento com os demais colegas de trabalho.

Adicionalmente, ele participa do processo de mensuração dos requisitos e identificação dos possíveis riscos de implementação das funcionalidades e comenta o código fonte da solução, tornando-o de fácil compreensão.

Em se tratando de artefato, o Desenvolvedor fica responsável pelo Componente de Software que nada mais é neste contexto, do que a codificação do modelo arquitetural lógico da solução, definido anteriormente pelo Arquiteto de Software.

7.7 ANALISTA DE TESTES

O papel de Analista de Testes é representado por uma pessoa com habilidades para realizar diferentes tipos de testes sobre software. Durante suas atividades no ciclo de vida do processo, ele deverá atuar em duas fases distintas. A primeira atuação do Analista de Testes é na fase de análise, aonde serão realizados planejamentos para a realização dos testes nas fases seguintes. Durante o planejamento ele elabora um artefato denominado Plano de Testes que contém todas as diretrizes e resultados esperados para aqueles requisitos. O objetivo disso é garantir que as funcionalidades especificadas pelo Cliente não perderão a essência dos resultados esperados.

A segunda atuação deste papel é na fase de Testes. Nela o Analista de Testes acompanha a execução de todos os testes e compara os resultados com as diretrizes contidas no artefato Plano de Testes, definido anteriormente por ele. Ele também realiza as atividades de um testador executando todos os testes especificados. Como atribuições, este papel desempenha, por exemplo, testes de unidade, testes de integração, testes de segurança de dados, testes de aceitação e demais testes necessários para garantir um bom nível de qualidade para a solução.

7.8 ANALISTA DE SUPORTE

O papel do Analista de Suporte deve ser incorporado por uma pessoa capaz de realizar treinamentos para os usuários sobre sistemas computacionais. Para realizar suas atribuições de maneira eficiente, ele deve ser comunicativo e ter boa relação interpessoal, já que ele estará em contato tanto com os futuros usuários da solução e com o Cliente.

O Treinador, antes da realização dos treinamentos, desenvolve um Plano de Treinamento, contendo todas as especificações necessárias para a realização dos treinamentos dos usuários. Nele estão contidas informações como: período de execução das atividades, produtos a serem treinados, recursos a serem utilizados, etc.

Durante a definição do Processo proposto foi decidido que o Treinador também realizará a implantação da solução. Para isso, ele deve realizar um Plano de Implantação com características bem parecida com as de um Plano de Treinamento. Porém, possui uma maior ênfase nas especificações de configuração de hardware e software.

Ele também é responsável pela emissão do Comprovante de Implantação da funcionalidade ou solução. Este artefato não possui um modelo definido aqui, ficando sua criação a cargo da empresa de desenvolvimento que aderir ao processo.

7.9 ANALISTA DE QUALIDADE

Este papel é representado pela pessoa que realizará as auditorias sobre os artefatos produzidos durante o ciclo de vida do processo. Ele observa os artefatos, examinando-os segundo os padrões estabelecidos pela empresa de desenvolvimento.

Para melhor desenvolver suas atividades, o Analista de Qualidade faz uso da Lista de Verificação da Qualidade. Ela funciona como um *checklist* possibilitando um maior controle do que foi desenvolvido ou não. Os documentos já criados e verificados são classificados em:

- **Conforme** – quando um artefato está, por completo, aderente às políticas de qualidade estabelecidas pela empresa;
- **Não-conforme** – quando um artefato, ou parte dele, não obedece às políticas de qualidade da empresa;
- **Não se aplica** – quando não se pode classificar um artefato, parte dele ou outra característica, dentro dos padrões de qualidade estabelecidos;

- **Pendente** – quando uma não-conformidade já foi apontada em um artefato e, em uma segunda verificação, ela ainda permanece inalterada ou não apresenta qualidade satisfatória.

7.10 CONFROTAMENTO DE PAPÉIS

Como sendo um dos objetivos pretendidos para este trabalho, o Processo proposto também pode flexibilizar os seus papéis de modo a atender as necessidades do ambiente onde ele está sendo aplicado.

Sabe-se que muitas empresas de software, geralmente, trabalham com um quadro de funcionários inferior às suas necessidades, quando se trata de atividades. Ou seja, é observável que na maioria dos casos uma pessoa acumule vários papéis dentro de uma organização. Isso não é encarado aqui como um problema, mas sim como característica, devido ao dinamismo das empresas modernas que exigem dos seus colaboradores conhecimentos e habilidades um tanto “distribuídas”. Nesse caso, entenda como sendo uma pessoa que possui conhecimento em várias áreas e habilidade para lidar com diversas situações diferentes.

7.11 DISTRIBUINDO PAPÉIS EM UMA MICROEMPRESA

Para tornar mais claro o entendimento a cerca dessa característica, considere um exemplo de uma empresa contendo cinco pessoas. Os papéis definidos para o Processo proposto serão distribuídos para esse número de pessoas, considerando a grande quantidade de atividades que cada papel possui, não tornando um ou outro papel sobrecarregado ao ponto de deixar o processo ineficiente.

Nos projetos de software, cada papel tem seu objetivo e interesses definidos. Porém, alguns deles não podem ser exercidos simultaneamente pela mesma pessoa por correrem o risco de agredir a qualidade do produto e do processo. Em contrapartida, outros papéis podem ser agrupados de forma eficiente devido às características que cada um deles possui. A seguir continuaremos com um exemplo.

O primeiro agrupamento a ser considerado realiza a junção entre os papéis de Cliente (interno) e Analista de Requisitos, onde o segundo incorpora as atribuições do primeiro. O objetivo dessa junção está no fato de o Analista de Requisitos também possuir características

de comunicação.com o Cliente (externo). Logo, o Analista de Requisitos passa a ser responsável pelo artefato de Regras de Negócio e Glossário.

O segundo agrupamento une o Arquiteto de Software com o Desenvolvedor. Nesse caso, o primeiro papel passará a desempenhar todas as atribuições do segundo, ficando responsável pelo Componente de Software e realizando todo o processo de codificação.

O terceiro agrupa o Gerente de Projetos com o Analista de Qualidade, sendo motivado pelo fato de o primeiro papel acompanhar todas as atividades do processo. Assim, além dele gerenciar a equipe, também terá melhores condições de gerenciar a qualidade dos artefatos produzidos.

A quarta forma de agrupamento proposta neste trabalho une dois papéis: o Analista de Testes e o Analista de Suporte. Agora, o primeiro papel fará não só o planejamento, definição de diretrizes de testes e execução, como também passará a realizar o planejamento e execução dos treinamentos e implantação da solução.

Qualquer outra forma de combinação entre papéis deve ser avaliada. Por exemplo: os papéis de Desenvolvedor e Testador, não devem ser atribuídos a uma mesma pessoa devido ao risco de o Desenvolvedor criar testes ineficientes para os componentes de software que ele mesmo desenvolveu, entre outros casos.

7.12 NÍVEL DAS ATIVIDADES

As atividades dentro de um processo de desenvolvimento de software fazem parte do segundo nível de detalhamento de um processo. Elas são importantes para especificar em que momento será trabalhado qual aspecto da solução. Uma característica importante de uma atividade⁷ é o que ela expressa. Ela define **quem** participa da atividade, **quando** ela deve ser realizada, quais os **artefatos de entrada** e quais os **artefatos de saída**.

A Figura abaixo, mostra um exemplo de atividade. No canto superior esquerdo, encontra-se o(s) participante(s), que realiza, participa ou acompanha a atividade. No canto inferior esquerdo, estão as siglas dos artefatos de entrada, ou seja, os artefatos necessários para o bom desenvolvimento da atividade. No canto superior direito estão os artefatos de saída. Artefatos esses criados ou atualizados na atividade. Finalizando, no centro encontra-se a denominação dela.

⁷ A forma de representação mostrada acima é denominada PEPP (AGUIAR e ROULIER, 2004).

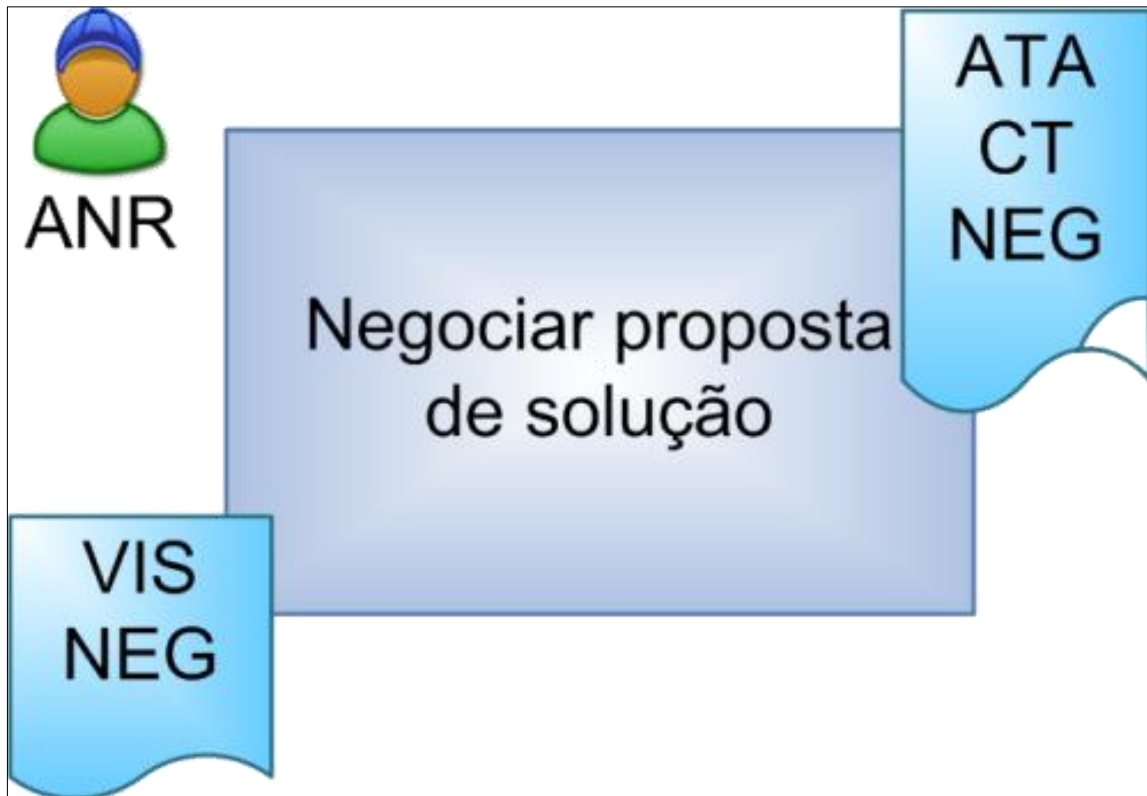


FIGURA 7 - EXEMPLO DE REPRESENTAÇÃO DE UMA ATIVIDADE

Atividades da fase de solicitação

A organização das atividades que compreendem a fase de solicitação, quais os seus objetivos e demais especificações necessárias estão descritas a baixo:

-Solicitar serviço

Esta atividade descreve como o cliente deve entrar em contato com a empresa de desenvolvimento de soluções de software para solicitar um novo serviço ou como ele deve proceder em caso de solicitação de mudança de funcionalidade. Aqui, o cliente irá contatar a empresa em busca de solucionar suas necessidades, vindo a iniciar uma Solicitação.

As solicitações geradas nesta atividade são de responsabilidade do Cliente, onde o artefato Solicitação representa o principal meio de comunicação ativo do Cliente e a empresa. Outros meios de comunicação podem ser adotados e especificados em um outro artefato denominado Plano de Comunicação, geralmente usado em grandes projetos.

-Analisar necessidade

Esta atividade descreve como será realizada a atividade de análise de necessidades do Cliente que solicitou um serviço. Ela especifica também como observar o ambiente do Cliente em busca de conhecimentos mais detalhados sobre o problema em questão, e a forma como as pessoas que são afetadas por ele abordam a problemática e propõe soluções para os mesmos. Serão identificadas as necessidades expostas na Solicitação e então será reunida a maior quantidade de informações possíveis sobre aquele contexto. Para essa atividade serão necessários: o artefato de Visão de Negócio, Glossário e Regras de Negócio. Sendo estes utilizados pelos papéis: Cliente e Analista de Requisitos.

O Analista de Requisitos será responsável por analisar o cenário da solicitação, as necessidades do Cliente, receber e propor sugestões de solução para o problema das partes envolvidas. Já o Cliente será responsável por disponibilizar essas informações e propor soluções para o problema.

-Negociar proposta de solução

Esta atividade indica de que forma ocorrerá o processo de negociação da proposta de solução feita pelo Analista de Requisitos e Gerente de Projetos. Em visão geral, propõe-se uma reunião entre Cliente, Analista de Requisitos e Gerente de Projetos para que haja o esclarecimento da solução que será detalhada nas atividades consequentes. A proposta resulta na aceitação ou rejeição de um contrato firmado entre empresa e o Cliente para que sejam iniciadas as atividades de especificação da solução.

As propostas de solução serão discutidas observando a correspondência das necessidades no contexto especificado, sendo definida como solução a ser desenvolvida. Com base nesta definição, será elaborado o Contrato de Aceitação, no qual será de responsabilidade do Gerente de Projetos. Nele serão definidos em detalhes do que se trata a funcionalidade em termos de preços e prazos.

Serão participantes desse processo: o (i) Analista de Requisitos, descrevendo as soluções propostas; (ii) o Gerente de Projetos, realizando as negociações e preenchendo a Ata de Reunião e; (iii) o Cliente, julgando, segundo seu entendimento, as soluções que serão especificadas, concordando ou não com os contratos.

-Atividades da fase de análise

Este tópico exibe como serão realizadas as atividades de análise, identificação de requisitos e especificação dos mesmos. Para isso, serão expostas visões gerais, explicações e pontos importantes de cada uma das atividades.

-Elicitar requisitos

Esta atividade descreve quais os passos necessários para que haja uma elicitação de requisitos de maneira concreta e assertiva. Ela é responsável pela definição funcional e não-funcional dos requisitos de software, incluindo a especificação dos diversos diagramas necessários para uma definição concreta das funcionalidades e características da solução de software. Para dar início a esta atividade, deve-se ter em mãos documentos básicos que descreva o ambiente e o problema que o Cliente enfrenta, sendo resultantes, documentos técnicos de modelagem da solução e protótipos para avaliação.

Como responsável por essa atividade temos o Analista de Requisitos. Ele reúne todas as informações inerentes ao projeto e elabora os artefatos (Documento de Elicitação de Requisitos, Especificação de Casos de Uso e Protótipos de Interface com o Usuário). Esses documentos funcionarão como alicerce para o projeto, pois servirão como repositório principal de informações o no decorrer de sua execução do projeto. Com a revisão de requisitos, dá-se o suporte à atividade de elicitação de requisitos através de uma revisão no conteúdo produzido objetivando eliminar ambiguidades, requisitos duplicados, erros de entendimento e identificar o relacionamento entre os requisitos. Nesta atividade, o Cliente deve participar ativamente, identificando equívocos do Analista de Requisitos e sugerindo mudanças e definirão corretamente os requisitos. Os artefatos destinados à revisão são: Documento Elicitação de Requisitos e Documento de Especificação de Casos de Uso. Assim, o Analista de Requisito montará uma Matriz de Rastreabilidade tendo, para isso, o suporte do Cliente.

Essas atividades têm o objetivo principal de refinar as informações dos documentos, mantendo a fidelidade entre as informações nos documentos e o ambiente observado. Os documentos serão escritos de forma a simples e objetiva, onde todos os envolvidos no processo possam ter um entendimento fácil sobre os conceitos descritos no documento. Nesta fase os documentos de Elicitação de Requisitos e Especificação de Casos de Uso são atualizados conforme conveniente ao projeto.

-Definir tecnologia

Nesta atividade serão discutidas as questões referentes às tecnologias a serem empregadas no desenvolvimento do projeto. O analista de requisitos junto com o arquiteto de software irá definir como aproveitar melhor as tecnologias existentes observando o custo, acessibilidade e adaptabilidade à realidade do Cliente

Uma vez analisados todos os aspectos do contexto onde o problema está inserido, o Arquiteto de Software reunirá informações de tecnologias que podem ser utilizadas para a resolução daquele problema.

Junto com o Analista de Requisitos, ele irá definir quais são as tecnologias que compõem o arcabouço conceitual que servirá de base para a construção do sistema.

-Avaliar riscos e mensurar requisitos

Esta atividade explica como deve ser feita a atividade de avaliação de riscos de implementação e mensuração de requisitos. Aqui, o Analista de Requisitos deve identificar os riscos de implementação observando aspectos de tempo, possibilidade de desenvolvimento no cenário atual e viabilidade. Um ou mais desenvolvedores devem participar da mensuração, dando para todos os requisitos um número de pontos (horas de trabalho).

A ideia de mensurar os requisitos em pontos funciona da seguinte maneira: É apresentada a especificação de um requisito/caso de uso para um desenvolvedor, através de leitura ou apresentação gráfica. Em seguida, de acordo com o entendimento dele em relação ao desenvolvimento, serão atribuídas três notas: (i) a quantidade de pontos necessárias no pior caso de desenvolvimento do requisito, ou seja, quando o desenvolvedor não tem experiência e o problema pode ser mais complicado do que o que parece; (ii) a quantidade normal de pontos para ele desenvolver os requisitos e; (iii) no melhor caso, a quantidade de pontos necessárias para ele desenvolver os requisitos, tendo em vista que tudo ocorrerá sem interferências e ele já tenha experiência em uma solução idêntica ou bem parecida com a apresentada.

Vale ressaltar que esse tipo de mensuração é apenas uma sugestão. Na prática, podem ser adotadas várias outras formas de mensurar os requisitos.

- Definir diretrizes de teste

Aqui estão definidas as formas de realização dos testes sobre os requisitos da solução. Assim, o Analista de Teste estima os resultados gerados pelo sistema como solução satisfatória para os problemas do Cliente. Dessa forma, um Plano de Testes é definido para dar suporte à realização fiel dos testes nas fases posteriores do processo de desenvolvimento. Nessa atividade o Analista de Testes irá analisar o Documento de Elicitação de Requisitos e definirá as diretrizes dos testes que serão aplicadas. Uma vez definidas as entradas, saídas e as condições de execução de cada teste, o Analista de Testes irá elaborar o Plano de Testes, que formalizará todo o procedimento de teste que será realizado sobre o sistema.

- Negociar requisitos

Aqui estão definidas as atividades necessárias para a negociação de requisitos entre o Gerente de Projetos e o Cliente. Eles se reunirão para definir prioridades de requisitos, estabelecer prazos e estipular metas. O Analista de Requisitos participa, esclarecendo e apresentando os requisitos elicitados e definidos. Finalizando com um contrato firmado entre o Cliente e a empresa de desenvolvimento.

Atividades da fase de planejamento

Planejar desenvolvimento

Nesta atividade estão descritos os passos necessários para que o planejamento do projeto seja realizado de forma satisfatória. O Gerente de Projetos, ao iniciar esta atividade, já deve possuir conhecimento sobre os requisitos, sobre o ambiente onde a solução será desenvolvida, testada e utilizada, sobre os recursos necessários para a execução do projeto, orçamento, prazos, sobre como se pretende desenvolver os requisitos etc. Assim ele tem embasamento suficiente para criar um bom Plano de Projeto, que seja satisfatório em prazo e viável em termos de custo.

O responsável por essa atividade é o Gerente de Projetos, que irá desenvolver o Plano de Projeto de Software. Esse documento irá reunir informações referentes aos recursos físicos e humanos a serem utilizados no decorrer do desenvolvimento do projeto, bem como os prazos e metas a serem atingidas pela equipe.

Preparar ambiente de desenvolvimento

Esta atividade descreve como deve ser feita a preparação do ambiente de desenvolvimento de um software. A preparação, que será realizada pelo Gerente de Projetos, envolve tanto recursos humanos, como materiais. Os recursos humanos são: a alocação de funcionários para determinadas atividades e/ou a contratação de novos ou capacitação. Os recursos materiais são todos os equipamentos de hardware computacional e escritório, incluindo-se também as ferramentas de software, ou qualquer outro recurso necessário para o desenvolvimento adequado do processo.

Atividades da fase de desenvolvimento

Definir arquitetura

Atividade feita pelo Arquiteto de Software, sendo uma das mais importantes para o sucesso do projeto. Aqui serão definidas todas as especificações técnicas (diagramas de classe UML, por exemplo) da solução na tecnologia definida na fase de análise. Enfim, todo o arcabouço necessário para a codificação consistente da solução. O Arquiteto de Software irá desmembrar o documento de Arquitetura de Software e irá detalhá-lo de forma que possa ser executado pela equipe de desenvolvimento. Nesta fase será elaborado o plano de Integração do Build, que terão formatados todos os conceitos técnicos que irão estruturar o sistema.

Codificar

A atividade de codificação do software define os passos necessários para criar todo o código fonte de todos os métodos predefinidos na atividade de arquitetar os requisitos. Aqui, o desenvolvedor cria, revisa e comenta o código fonte de forma eficiente e facilite a leitura posterior. Segundo definido nas fases anteriores, serão construídos os algoritmos que constituirão o produto de software, que obedecerão ao padrão de construção definido no Modelo de Implementação do Software.

Atividades da fase de teste

Testar

A atividade de testar o software, embora simbolizada como única, envolve vários tipos de testes (descritos na seção Fluxo Principal). Os testes verificam vários aspectos do funcionamento do software. Os resultados destes testes são verificados e remetidos para as atividades mais convenientes para serem corrigidos quando resultam em erro.

- **Teste de integridades dos dados** – Este teste verifica, por exemplo, se a precisão dos cálculos está dentro do especificado ou se os dados são armazenados e exibidos ao usuário de forma correta e concisa.
- **Teste de interface com o usuário** – Verifica se todos os controles (botões, caixas de listagem, botões de checagem, etc.) estão funcionando perfeitamente. Também são verificadas as posições das caixas de mensagem exibidas e observação da formação dos controles durante o dimensionamento das janelas.
- **Teste de segurança** – Verifica a segurança das informações que trafegam entre o usuário e o sistema (WEB ou LAN) ou se dados estão sendo exibidos para papéis não autorizados
- **Teste de funcionamento** – Verifica através da simulação de um caso de uso real, se o funcionamento do sistema está dentro do esperado.
- **Teste de aceitação** – Este teste observa a aceitação do cliente em relação à(s) funcionalidade(s) desenvolvida(s). É esperado que, durante esse teste, seja gerada alguma solicitação de mudança, já que esse é o primeiro contato do cliente com a solução.
- **Teste de integração** – Um dos testes mais importantes para a escalabilidade do sistema. Esse teste observa todos os aspectos da integração da nova funcionalidade com o sistema (em caso de ser uma integração de uma nova funcionalidade a um sistema existente).

Durante a realização dos testes, normalmente são encontradas falhas, inconsistências e irregularidades. Dependendo de qual seja, a correção é encaminhada para a fase e atividade responsável por resolvê-las.

Uma funcionalidade só passará para a fase seguinte se ela passar em todos os testes submetidos. Isso especifica que ela está em estado de aceitação e funcionamento.

Atividades da fase de entrega

Planejar treinamento e integração

Nesta atividade deve ser feito o planejamento para a realização dos treinamentos dos usuários finais, além do planejamento de como será feita a implantação da solução no ambiente do cliente. Planejamentos que levarão à criação dos artefatos: Plano de Treinamento e Plano de Implantação. Estes serão criados através de um ou mais acordos firmados entre o cliente e a empresa de desenvolvimento, devidamente documentados na Ata de Reunião. Acordos visando suprir as necessidades do cliente em termo de tempo, espaço e disponibilidade no ambiente dele.

Treinar usuários

Esta atividade descreve como devem ser feitos os treinamentos dos usuários finais para que eles estejam preparados para a implantação e utilização da nova solução de software. Treinar os usuários da solução é uma tarefa muito importante e deve ser feita antes da implantação.

Sabe-se que o ser humano naturalmente possui resistência à mudança e por isso é importante treinar os usuários de uma solução antes da utilização efetiva. Isso é necessário porque, geralmente, é adicionada alguma mudança nos procedimentos normais de uma atividade que pode passar a ser totalmente ou parcialmente automatizada por um software, visando a exclusão ou diminuição de entraves típicos das atividades manuais. Assim, os treinamentos devem apresentar a nova solução esclarecer as dúvidas dos usuários até que eles sintam-se seguros.

Implantar solução e dar suporte

Esta atividade descreve como será feita a implantação da solução previamente planejada e também do suporte ao cliente.

No caso da implantação, é importante ter um bom conhecimento sobre o ambiente do cliente. Por exemplo: se o cliente for um supermercado que deseja automatizar as operações no caixa.

É importante que a implantação seja executada fora do horário de atendimento para não atrapalhar o atendimento ao cliente, ou possa vir a gerar uma situação que venha a causar transtornos ao cliente.

No caso do suporte, as opções disponibilizadas pela empresa serão oferecidas ao cliente e incluídas no contrato antes do início das atividades de especificação da solução. Assim, o cliente poderá ter várias opções de suporte disponíveis de acordo com os seus interesses.

Alguns exemplos são:

- Suporte via telefone;
- Suporte online (mensagem instantânea, e-mail, etc.);
- Suporte presencial (envio de um profissional até o cliente);
- Manual de usuário impresso ou em mídia eletrônica;
- Assistência remota.

As possibilidades de suporte ao usuário são inúmeras. Nesse momento uma empresa deve oferecer o melhor serviço possível, de forma a minimizar os esforços do cliente na busca de esclarecimentos sobre dúvidas.

Verificação da qualidade (SQA)

Levantamento do projeto

A atividade onde é feita a análise do andamento do projeto, se o mesmo está sendo realizado conforme o planejado e se obedece aos padrões de qualidade definidas para o processo e para o produto. Também é feita uma análise dos problemas e dificuldades com o fim de evitar que estes venham a ocorrer. Esta atividade é desenvolvida pelo Analista de Requisitos que observará os problemas que atrapalharam a fase e os principais efeitos causados no processo através da comparação das atividades realizadas e as programadas no plano de projeto de software.

Avaliação da qualidade

O Analista de Qualidade irá revisar as evidências para certificar que o processo de os padrões para qualidade do produto estejam sendo observados. O objetivo desta atividade é manter a qualidade na construção do produto e na execução das atividades conforme

observado no processo, o que facilitará futuras manutenções e fácil entendimento dos envolvidos no processo.

-Reunião

O Analista de Qualidade irá passar sua avaliação acerca da análise realizada ao gerente, para que este possa ter um maior controle sobre o andamento do projeto, já que pode haver a necessidade de realizar algum realinhamento ao que foi planejado, como também manter a aderência da execução dos procedimentos aos padrões de qualidade institucionalizados.

-Correção

O Analista de Qualidade irá comunicar os desvios aos padrões aos responsáveis. Os artefatos que possuírem inconformidades serão devolvidos aos seus respectivos responsáveis para que eles façam as devidas alterações. Uma vez as alterações realizadas, o processo seguirá seu curso normal.

8 METODOLOGIA

O tipo de pesquisa que foi utilizada neste trabalho foi a Bibliográfica. Segundo LAKATOS (1983), pesquisa bibliográfica é desenvolvida a partir de um material elaborado (livros, artigos científicos e material eletrônico).

A pesquisa bibliográfica é o primeiro passo para qualquer pesquisa científica, com o objetivo de revisar a literatura existente. Por meio dela, buscam-se informações, seleciona-se documentação que se relaciona ao problema de pesquisa e faz-se o respectivo arquivamento para posterior utilização.

Para uma melhor compreensão do conteúdo desta monografia, ela está dividida da seguinte forma:

No primeiro momento é abordado à parte introdutória, bem como a metodologia usada para o desenvolvimento da pesquisa e sua estrutura.

No segundo momento inicia-se a fundamentação teórica apresentando alguns conceitos da Gestão de Projetos, o que é e como funciona e uma breve explicação do papel do gestor na organização.

No terceiro momento abordamos a atual realidade da Gerência de Projetos em pequenas e médias empresas e suas dificuldades de implantá-la.

No quarto momento apresentamos uma explanação de algumas ferramentas e metodologias necessárias para a implantação da gerência de projetos em organizações de pequeno e médio porte, assim também como um modelo de gestão eficaz.

Por fim, concluímos este trabalho com algumas considerações finais sobre o mesmo com uma base para possíveis melhorias futuras.

9 CONSIDERAÇÕES FINAIS

Tendo em vista todas as informações reunidas para o desenvolvimento desse projeto, observa-se que muito há para fazer e estudar sobre a melhoria do desenvolvimento de software, principalmente quanto ao desenvolvimento de software por pequenas empresas. O objeto alcançado neste projeto foi a modelagem de um processo de desenvolvimento reunindo características que o torne simples. Porém, completo o suficiente para que equipes e empresas, iniciantes ou não, tenham todo o arcabouço para seguirem durante o desenvolvimento dos seus projetos. Foi observado também que o sucesso na implantação de um processo de software está intimamente ligado às características do problema a ser atacado e aos recursos disponíveis para sua resolução.

O processo aqui apresentado e descrito está apto a gerir as problemáticas envolvidas no desenvolvimento de um software, de forma que, qualquer profissional que tenha o mínimo de conhecimento dos conceitos da Engenharia de Software possa desenvolvê-lo sem dificuldades atingindo um alto grau de qualidade.

Em resumo, no decorrer do desenvolvimento do projeto foi reunida uma carga de conhecimentos que irá contribuir para a vida profissional dos participantes da equipe envolvida no desenvolvimento do Processo proposto. Além de resultar em um processo de software que permitirá a profissionais iniciantes, desenvolverem um projeto seguindo os conceitos de alguns dos principais autores contemporâneos sobre o tema deste trabalho.

REFERENCIAS BIBLIOGRÁFICAS

ABEPRO (Associação Brasileira de Engenharia de Produção) **Projetos de Tecnologia da Informação: Caracterização da Gestão de Projetos de TI no Estado de Pernambuco.**

Disponível em:

<http://www.abepro.org.br/biblioteca/enegep2008_TN_STO_076_538_12018.pdf>. Acesso em: 16 mai. 2013.

AMBER, S. W. **Process Patterns: Building Large-Scale Systems Using Technology.** Cambridge: Cambridge University Press, 1998.

AGUIAR, H. V.; ROULIER, A. C. Primitivas para Definição de Processo - PEPP. **SWQuality**, Recife, 2004. Disponível em: <www.swquality.com.br/pepp/>. Acesso em: 5 Dezembro 2009.

BONNER, W. Setor de tecnologia da informação cresce com a crise. **Site da Globo.com/jornalnacional**, São Paulo, 13 de Fevereiro de 2009. Disponível em: <<http://video.globo.com/Videos/Player/Noticias/0,GIM965786-7823-SETOR+DE+TECNOLOGIA+DA+INFORMACAO+CRESCE+COM+A+CRISE,00.html>>. Acesso em: 26 de Outubro de 2012..

DUTRA, D. Nerd Inside. **Blog Xande-broba**, 17 de Outubro de 2008. Disponível em: <<http://xande-borba.blogspot.com/2008/08/atividades-21-08-2008.html>>. Acesso em: 26 de Outubro de 2009.

DUTRA, L. R. Paradigmas de Engenharia de Software. **Site da Universidade de Brasília**, 2008. Disponível em: <<http://www.redes.unb.br/material/Metodologia%20de%20Desenvolvimento%20de%20Software/aula3.pdf>>. Acesso em: 8 de Novembro de 2012

FILHO, E. M. G. et al. Tributação e Desenvolvimento no Setor de Software Brasileiro. **Site da Abes - Associação Brasileira das Empresas de Software**, Dezembro 2006. Disponível em: <<http://www.faltaesselink.com.br>>. Acesso em: 15 de Outubro de 2012

GERVAZONI, Thiago Pastorello. **Conceitos Básicos de Gerenciamento de Projetos**. Disponível em: <<http://www.linhadecodigo.com.br/artigo/1167/conceitos-basicos-de-gerenciamento-de-projetos.aspx/>>. Acesso em: 16 mai. 2013.

INTERNATIONAL BUSINESS MACHINES - IBM. IBM Rational Unified Process. **IBM - International Business Machines**, 2009. Disponível em: <<http://www-01.ibm.com/software/awdtools/rup/>>. Acesso em: 14 de Novembro de 2012

ISO/IEC 12207. 12207:2008(E) Systems and software engineering - Software life cycle processes. **Site da ISO - International Organization for Standardization**, 2008. Disponível em: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43447>. Acesso em: 4 de Novembro de 2009.

LAKATOS, Eva M. Fundamentos da Metodologia Científica. São Paulo: Atlas, 1991.

MACHADO, C. Â. F.; WEBER, K. C. **Qualidade e Produtividade em Software**. São Paulo: Makron Books, 2007..

MELO, C. D. O. Reutilização de Software: Classificação e Seleção de Artefatos Reutilizáveis. **Site do Instituto de Matemática e Estatística da USP**, São Paulo, 25 de Junho de 2004. Disponível em: <<http://www.ime.usp.br/~yw/2004/mac5701i/monografias/claudia-acvm.pdf>>. Acesso em: 28 de Outubro de 2012.

PRESSMAN, R. S. **O papel evolutivo do software**. Tradução de Carlos Barbosa Santos. São Paulo: Makron Books do Brasil Editor Ltda, 1995. 4-6 p. ISBN ISBN: 8586804576.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. [S.l.]: McGraw-Hill, 2006. 17-55 p. ISBN 8586804576.

RATIONAL SOFTWARE CORPORATION. Papel: Arquiteto de Software. **Site da WThreex**, 2002. Disponível em: <<http://www.wthreex.com/rup/portugues/index.htm>>. Acesso em: 31 de Outubro de 2012.

REZENDE, J. L. D. Aplicando Técnicas de Conversação para a Facilitação de Debates no Ambiente AulaNet. **PUC-RIO**, 2003. ISSN 0115649/CA. Disponível em: <http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0115649_03_cap_08.pdf>. Acesso em: 7 Dezembro 2012.

SIMÃO, I.; VARELA, P. A Engenharia de Requisitos como processo inovador nas organizações. **Site da RUN - Repositório Universidade Nova**, Monte de Caparica, Julho 2009. Disponível em: <http://dspace.fct.unl.pt/bitstream/10362/1972/1/WPSeries_08_2009ISimao_PVarelaB.pdf>. Acesso em: 26 de Outubro de 2012.

SOFTEX. MPS.BR, capacitação e empreendedorismo. **Guia de Implementação – Parte 9: Implementação do MR-MPS em organizações do tipo Fábrica de Software**, 02 Outubro 2009. ISSN ISBN 978-85-99334-16-4. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte%209.pdf>. Acesso em: 12 de Outubro de 2012.

SOMMERVILLE, I. **Engenharia de Software**. 6. ed. [S.l.]: Addison Wesley, 2009

UENO, M. www.assespro.org.br. **Site da ASSESPRO - Associação das Empresas de Tecnologia da Informação, Software e Internet**, Abril 2007. Disponível em: <<http://www.assespro.org.br/images/pdti.pdf>>. Acesso em: 12 de Outubro de 2012.