UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA

Departamento de Ciências Exatas

Graduação em Ciência da Computação

ANÁLISE DO NÍVEL DE MATURIDADE DAS EMPRESAS DESENVOLVEDORAS DE SOFTWARE DE VITÓRIA DA CONQUISTA

Neylor Brito Rocha

Vitória da Conquista – Bahia
Janeiro de 2011

ANÁLISE DO NÍVEL DE MATURIDADE DAS EMPRESAS DESENVOLVEDORAS DE SOFTWARE DE VITÓRIA DA CONQUISTA

Neylor Brito Rocha

Projeto apresentado ao curso de Ciência da Computação do Departamento de Ciências Exatas da UESB, orientado pelo profº. Fábio Moura Pereira, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Vitória da Conquista – Bahia
Janeiro de 2011

Agradecimentos

Ao meu orientador, Fábio Moura, pelo apoio educacional e pela paciência.

Aos professores do curso de Ciência da Computação da UESB, que me ajudaram a trilhar esse caminho rumo a graduação.

Aos meus amigos, Mariana, Jorge, Tatiane, Marcos, Jadson e tantos outros que estiveram ao meu lado nesse caminho.

A minha amiga Juliane, pelos momentos de reflexão e entendimento, e pela maravilhosa amizade criada.

A amiga Celina, que, como foi desde início do curso, foi um anjo em minha vida.

Ao meu amigo Ricardo, pelo apoio moral e pelos ensinamentos passados.

A minha família, que foi e sempre será o pilar da minha vida.

Ao grandioso Deus, por me dar sabedoria, inteligência, paciência, perseverança e humildade.

E, principalmente, a mim mesmo, por não desistir.

Welcome 1 no swing na horizontal lambendo o feixe estrela com as 8 rainhas passeando com o cavalo no looping infinito..”

NEYLOR BRITO E MARIANA ROCHA

RESUMO

O gerenciamento correto de projetos, que é o uso de conhecimento e habilidades para criar tarefas que atinjam um determinado fim, levando em consideração os custos e esforços, se encontra como ponto fundamental para que empresas consigam atingir suas metas com qualidade, otimizando a utilização de recursos técnicos e humanos. Como um medidor desse gerenciamento existe os níveis de maturidade que são abordados por diversas metodologias de projeto. A partir da constatação da importância do gerenciamento de projeto e dos níveis de maturidade, foi que se viu necessário levantar a situação destes nas empresas desenvolvedoras de software da cidade de Vitória da Conquista para se obter conhecimento a respeito da qualidade do processo de desenvolvimento de software e indicar alternativas de melhorias no desenvolvimento de projetos mantendo-se os aspectos positivos já existentes. Essa análise foi construída, com a aplicação de questionário nas empresas alvos, e o resultado foi a averiguação de que as empresas desenvolvedoras de software necessitam de uma maior atenção com relação a forma com que elas gerenciam seus projetos para alcançarem níveis mais elevados de fatores como eficácia e eficiência na construção de seus produtos.

ABSTRACT

The correct management of projects, which is the use of knowledge and skills to create tasks to achieve a particular purpose, taking into consideration the costs and efforts, as the fundamental point is that companies are able to achieve their goals with quality, optimizing the use of technical and human resources. As a gauge of management are the maturity levels that are approached by various design methodologies. After noting the importance of project management and levels of maturity, was that he found it necessary to raise the status of software development companies in the city of Vitoria da Conquista to obtain results that indicate improvements in alternative development projects while keep the positives that already exist. This analysis was built, with a questionnaire on business targets, and the result was the finding that the software development companies need further attention regarding the way they manage their projects to achieve higher levels of factors such as effectiveness and efficiency in building their products.

SUMÁRIO

1. INTRODUÇAO.....11
2. O PROCESSO DE DESENVOLVIMENTO DE SOFTWARE NO BRASIL.....14
2.1. O MERCADO DE SOFTWARE NO BRASIL.....14
2.1.1. Histórico.....14
2.1.2. Características do mercado de software.....15
2.1.3. Dificuldades.....16
2.2. O MERCADO DE SOFTWARE EM VITÓRIA DA CONQUISTA.....17
2.3. O PROCESSO DE DESENVOLVIMENTO DE SOTWARE.....18
2.3.1. Metodologias Tradicionais.....19
2.3.2. Metodologias Ágeis.....24
3. O PROCESSO DE GERENCIAMENTO DE SOFTWARE.....28
3.1. QUALIDADE DE SOFTWARE.....28
3.1.1. Qualidade de Processo de Software.....29
3.2. GERENCIAMENTO DE PROJETO.....30
3.2.1. Gerenciamento de Projeto de Software.....32
3.2.2. Métricas de Gerenciamento de Projeto de Software.....33
3.2.3. Modelos de Processo de Gerência de Software.....34
3.4. CMMI.....35
3.5. PMBOK.....37
3.6. SWEBOK.....38
3.7. MPS-BR.....39
4. ANÁLISE.....44
4.1. AMOSTRAGEM.....44
4.2. O ESTUDO.....45
4.3. LEVANTAMENTO DOS RESULTADOS SOBRE GERENCIAMENTO DE PROJETO.....46
4.3.1. Estudo dos Resultados.....51
4.4. LEVANTAMENTO DOS RESULTADOS SOBRE GERÊNCIAMENTO DE REQUISTIOS.....53
4.4.1. Estudo dos Resultados.....54
5. CONCLUSÃO.....56

REFERÊNCIAS.....58
APÊNDICE 1.....65

LISTA DE FIGURAS

Figura 1: Gráfico da Segmentação do Mercado de Software(Prefeitura de Vitória da Conquista,2009).....18
Figura 2: Etapas do modelo de cascata.....20
Figura 3: Gráfico – "O escopo do projeto é bem definido?".....45
Figura 4: Gráfico - "Os processos seguem algum modelo de ciclo de vida?".....46
Figura 5: Gráfico - "O orçamento e o cronograma dos projetos, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos".....47
Figura 6: Gráfico - "Existe planejamento para os recursos humanos utilizados no projeto, como, por exemplo, a análise do perfil necessário para executar tal projeto?".....48
Figura 7: Gráfico - "O ambiente de trabalho e os recursos existentes neles são planejados?".....48
Figura 8: Gráfico - "Existem mecanismos de segurança para os dados existentes nos projetos?".....49
Figura 9: Gráfico - "Ao fim de cada marco de projetos são realizadas revisões?".....50
Figura 10: Gráfico - "Existe algum planejamento de ações de correções de projetos?".....50
Figura 11: Gráfico - "A ligação entre o plano de trabalho e os requisitos é revisada para não criar inconsistência?".....53

LISTA DE TABELAS

Tabela 1: Segmentação do Mercado de Software por Categoria – 2008 (ABES,2009).....16
Tabela 2: Segmentação do Mercado de Software – 2009 (Prefeitura de Vitória da Conquista,2009).....18
Tabela 3: Características de qualidade de software (ISO/IEC 9126,2003) 29
Tabela 4: Quadro dos Níveis de Maturidade segundo o CMMI (CMMI, 2004).....36
Tabela 5: MR-MPS: Níveis de Maturidade, Processo e AP(s) (SOFTEX ,2009).....42

1. INTRODUÇÃO

As práticas mais estruturadas de modelagem de software estão em crescimento. A Engenharia de Software, com suas técnicas e métodos, assume cada vez mais um papel de destaque dentro de empresas e indústrias e na criação e manutenção de sistemas, independente de tamanho, finalidade e público alvo dos mesmos. Nesse âmbito, os projetos de software e as formas de gerenciá-los entram como otimizadores das atividades necessárias para se obter sucesso na modelagem de sistemas.

A Gerência de Projeto de Software, tendo o gerente como seu principal recurso humano, vem como organizadora das atividades e conhecimentos vinculados aos projetos.

Outra ferramenta importante para as tomadas de decisões com relação as atividades vinculadas a implementação de software e, consequentemente, importante no gerenciamento dos projetos de software é a análise do ciclo de vida do mesmo. Com esta, é possível identificar e avaliar a ordenação dos procedimentos necessários para a obtenção do sistema, sempre levando em consideração que os passos desse processo são interdependentes.

Porém, mesmo tendo todos estes artefatos e muitos outros acessíveis e, a cada momento, mais fáceis e práticos de serem aplicados, existe um grande contingente de empresas, instituições, indústrias que não utilizam desses meios para aprimorar suas atuações ou que o fazem de forma reduzida, precária e até errada. Tal fato reflete no nível de maturidade da gestão de projetos que as empresas de software no Brasil alcançam. Segundo Kernez(2001), a maturidade de gestão de projetos consiste no desenvolvimento de processos repetitivos que garantam uma alta probabilidade de que os projetos alcancem o sucesso.

Um exemplo do cenário das empresas nacionais com relação a seus graus de maturidade é citado em um estudo do *Masschussets Institute Technology* – MIT (2003), onde é mostrado que, no ano de 2003, nenhuma empresa brasileira tinha alcançado nível 5 de maturidade, segundo os padrões CMM (*Compability Maturity Model*), ao passo que, na Índia, 32 empresas conseguiram o feito.

Apesar de nos últimos anos o mercado de software brasileiro estar se estruturando melhor para combater a concorrência, ele ainda se mantém muito aquém do necessário e desejado a um país de tão grande porte e com tanto potencial de mão-

de-obra e de mercado consumidor de inovação tecnológica.

Um dos fatores de maior impacto negativo nesses resultados é a má preparação dos profissionais que utilizam e até dos que desenvolvem as tecnologias de gerenciamento de projetos de software. Isso está diretamente ligado as contribuições que as instituições de ensino superior e técnico proporcionam ao desenvolvimento da mão-de-obra que atua no mercado.

O que se pode observar, também, é uma considerável discrepância entre os grandes pólos tecnológicos e as cidades de pequeno e de médio porte no que diz respeito aos investimentos proporcionados pelas instituições educacionais e cientificas às empresas de desenvolvimento de software.

Então, o problema que se instala a partir dessas variáveis é: como as empresas e as instituições educacionais de ensino superior e técnico devem se comportar para amenizar a situação brasileira de nível de maturidade na produção de software, principalmente nos ambientes menos favorecidos?

Observando a problematização erguida e levando em consideração todos os fatores depreciativos à maturidade das empresas de TI citados anteriormente, é que se fundamenta a necessidade de se analisar o mercado de desenvolvimento de software, tendo como foco da análise, neste trabalho, a cidade de Vitória da Conquista, por ser uma cidade de médio porte, ou seja, possui investimentos menores ao mesmo tempo em que é um pólo econômico importante na sua região. Além da verificação de mercado, as universidades, faculdades e instituições de ensino técnico da cidade também serão alvos dos estudos.

Este trabalho teve como objetivo principal mostrar, tanto às empresas quanto às instituições de ensino, a forma como elas estão atuando, onde são seus pontos fortes e fracos, para assim saberem como podem auxiliar no nível de maturidade do mercado de produção de software da cidade de Vitória da Conquista.

Hipoteticamente, ele poderá auxiliar a adaptação de tecnologias já existentes e o desenvolvimento de novas, já que apontará os erros, que necessitarão de atuação para serem corrigidos, e os acertos, que poderão servir como base perpetuada de sucesso.

O questionário foi o instrumento utilizado pelo qual, a partir dos resultados e estudos do mesmo, se chegou à análise desejada. A idoneidade dos dados e informações, a praticidade de se obter resultados tanto quantitativos quanto qualitativos, que são importantes para a análise dessa situação, e a possibilidade de grande abrangência do mini-mundo abordado para o estudo foram fatores importantes que justificaram a

utilização dessa técnica de recolhimento de dados e verificação de informações.

Outro ponto observado na preparação do questionário foi o fato de que ele foi moldado para a análise baseada no modelo de gerenciamento de processo MPS-BR, já que é um modelo nacional e que se encaixa melhor no estilo das empresas estudadas, que são de pequeno e médio porte.

As informações foram obtidas através de: cruzamento de dados; tratamento estatístico; apresentação de tabelas e gráficos.

O trabalho está estruturado da seguinte forma: no capítulo 2° é apresentada uma pesquisa sobre o Mercado de Software, partindo do âmbito mundial até chegar ao nível da cidade Vitória da Conquista. No 3° capítulo é mostrado um estudo sobre o Processo de Gerenciamento de Software e seus diversos aspectos. No 4° capítulo é apresentado o resultado da aplicação do questionário de avaliação. Por fim, é apresentada uma conclusão do projeto, apontando os conhecimentos adquiridos e os possíveis trabalhos futuros baseados neste.

As referências literárias são baseadas em artigos acadêmicos e científicos nas áreas de Desenvolvimento e Gerenciamento de Softwares e do uso de livros como “Engenharia de Software”, de Roger S. Pressman e “Engenharia de Software”, Ian Sammerville.

2. O PROCESSO DE DESENVOLVIMENTO DE SOFTWARE NO BRASIL

2.1 Mercado de Software no Brasil

2.1.1 Histórico

O Brasil, desde a década de 70 e, principalmente, na década de 80 com o surgimento dos sistemas bancários automatizados, vem se desenvolvendo de forma considerável na área de Tecnologia da Informação. Segundo Pires (1995), tal crescimento do mercado de software se dá pelas políticas de iniciativas dadas a tal setor decorrente da elevada demanda dos segmentos financeiros.

Esse início de crescimento fez com que o país se encontrasse, em 1990, segundo Schware(92), no 6° lugar dos maiores mercados mundiais de computares e serviços de informática, sendo que a renda com o software atingiu os US$ 234 milhões. Nesta fase, o mercado de software se aproveitava da atuação iniciante e crescente de uma área que se tornaria fundamental para este setor, a Engenharia de Software.

Em 2000, após uma década importante para o mercado de software, a participação das vendas de software no PIB brasileiro era de 0,7% do mesmo, o triplo do que era verificado em 1991.

> Um fato interessante é que, desde 1995, a indústria nacional de software cresce em uma taxa média de 11% ao ano, três vezes mais do que a de hardware. Em 2000, o Brasil possuía 5,4 mil empresas no setor de Tecnologia da Informação e Comunicação (TIC). (PINHEIRO, 2003)

Os olhares no mercado de software brasileiro se intensificaram ainda mais a partir dos anos 2000. Em 2001 a Dell, que já tinha instalações no país para faturamento e venda de produtos de hardware e de suporte a infra-estrutura de Internet, implementa um Centro de Desenvolvimento de Software voltado ao investimento em pesquisas. Este fato auxiliou na visualização do Brasil como importante produtor e exportador de software mundial.

Em 2009, o Brasil foi o 16° no *ranking* mundial dos maiores mercados de

software, segundo dados da ABES(2009). Existem hoje quase 9 mil empresas no país que atuam no setor e, dentre essas, 90% são de pequeno e médio porte. Um reflexo dessa situação pode ser verificado na empresa Totvs, maior empresa do seu ramo no país, que, até o final de 2009, tinha feito 13 aquisições e alcançado um aumento de mais de 20% de sua receita, tornado-se assim a oitava maior empresa, do seu setor, no mundo, tendo filiais em países como Argentina, México e Portugal.

2.1.2. Características do mercado de software

O mercado brasileiro de software, desde seu surgimento, se apresenta como um dos mais complexos e diversificados do mundo, fato esse acarretado pela própria diversidade do país com relação a variáveis como a cultura, a educação, a economia, entre outros. Dentre as características desse peculiar mercado, pode-se observar:

- Segundo Carvalho (2008), mais da metade da demanda da produção de software é destinada aos setores financeiros e industriais, seguidos pelos de serviços, comércio, governo e agroindústria.
- O Brasil, tanto quanto a China e a Coréia do Sul, possui um mercado interno mais significativo do que o externo, segundo Veloso(2003).
- O mercado de software emprega mais de 160 mil trabalhadores, o que é de grande significância no cenário mundial.
- "Quanto à formação de *clusters*[1], o Brasil conta com importantes centros de tecnologia de software nas principais capitais do sul e sudeste, além de várias cidades médias em várias regiões do país"(BARBOSA et. al.,2005, p. 8).
- Como tendências de curto e médio prazo, de acordo com a ABES (2009), as empresas devem investir em segurança de informação, *Business Intelligence* (BI)[2], *Business Process Management* (BPM)[3] e ferramentas de melhoria de gestão de TI, fato que proporciona crescimento em pesquisas e investimentos no mercado de software.

[1] conjunto de computadores que utilizam de sistema distribuído para executar uma determinada função.

[2] é a coleta, organização, armazenamento, compartilhamento e monitoramento de dados para dá suporte a gestão de negócios.

[3] conceito de tecnologia de informação com foco na otimização de resultados das organizações através da melhoria do processo de negócio.

As segmentações de mercado também sofrem grande influência das diversidades encontradas no país. A tabela abaixo mostra como a segmentação por categoria estava disposta em 2008:

Tabela 1 – Segmentação do Mercado de Software por Categoria – 2008 Fonte: ABES(2009)

Segmentação do Mercado de Software por Categoria - 2008		
Categoria de Software	Volume (US$ milhões)	Participação (%)
Aplicativos	1.560	40,6
ERM(*Enterprise Resource Management*)	514	
CRM (*Customer Relationship Management*)	91	
SCM (*Supply Chain Management*)	141	
Outros (Usuário Final, Engenharia, Manufatura)	814	
Desenvolvimento e Implementação	1.203	31,3
Gerenciamento de Bancos de Dados Relacionais	498	
Aplicativos para Implementação	334	
Análise e Entrega	91	
Outros (Ferramentas de Qualidade, Desenvolvimento, etc)	280	
Infra-Estrutura	1.077	28,1
Segurança	166	
Armazenamento	182	
Gerenciamento de Sistemas e Redes	302	
Outros (Sistemas Operacionais, Interfaces, Subsistemas)	427	
Total	3.840	100

2.1.3 Dificuldades

Mesmo tendo um desempenho considerado de pais de primeiro mundo, o mercado de software brasileiro ainda encontra alguns obstáculos para obter o sucesso desejado. Segundo Pinheiro(2003), pode-se destacar como percalço para a avanço do mercado o chamado Risco Brasil, que são as políticas adversas implementadas pelos governo brasileiro e que impedem o desenvolvimento na industria nacional. Outra falha apontada pela autora é a falta de empresas nacionais no mercado aberto, fato que

somente começou a ser verificado a partir da década de 90, tendo em vista que, nessa mesma época, a Índia, por exemplo, já tinha 10 anos de inserção de empresas nesse mercado.

O Caderno de Altos Estudos sobre o Mercado de Software no país, documento feito pela Câmara dos Deputados, em Brasília, em 2007, que traçava um panorama sobre o mercado de software brasileiro, colocou como principais desafios encontrados pelo setor: a dificuldade de financiamento decorrente da falta de patrimônio para servir como garantia, a rápida depreciação dos ativos causada pelo constante avanço das tecnologias, a preferência dos governos em comprar produtos produzidos em outros países, como China e Índia e o monopólio de empresas multinacionais na produção de sistemas operacionais e ferramentas profissionais.

Como alternativa de solução para tais problemas está a criação de políticas públicas para incentivo ao setor, tanto na questão comercial quanto nas pesquisas e inovações e a adoção como prioridade, pelo setor público, da compra de produtos fabricados no mercado interno.

2.2 O Mercado de Software em Vitória da Conquista

O mercado de software de Vitória da Conquista ainda se encontra em um estado inicial de maturação, apesar de seu surgimento não ser tão recente (datado do início dos anos 80).

A maioria das empresas que atuam no ramo da informática em Vitória da Conquista trabalha com prestação de serviços e venda de componentes. Já as que se dedicam ao desenvolvimento de software são, em sua maioria, de porte micro e que sustentam contas de empresas e do governo da própria cidade e de cidades circunvizinhas.

Os esquemas abaixo mostram como está segmentado o mercado de informática de Vitória da Conquista.

Tabela 2 – Segmentação do Mercado de Software – 2009 Fonte: Prefeitura de Vitória da Conquista(2009)

Segmentação das empresas de informática de Vitória da Conquista		
Tipo	Quantidade	%
Desenvolvedoras de Software e atuação similar	14	7,9
Hardware em geral(manutenção, venda, instalação)	119	67,2
Redes	19	10,7
Outros serviços de TI	25	14,1
Total	177	

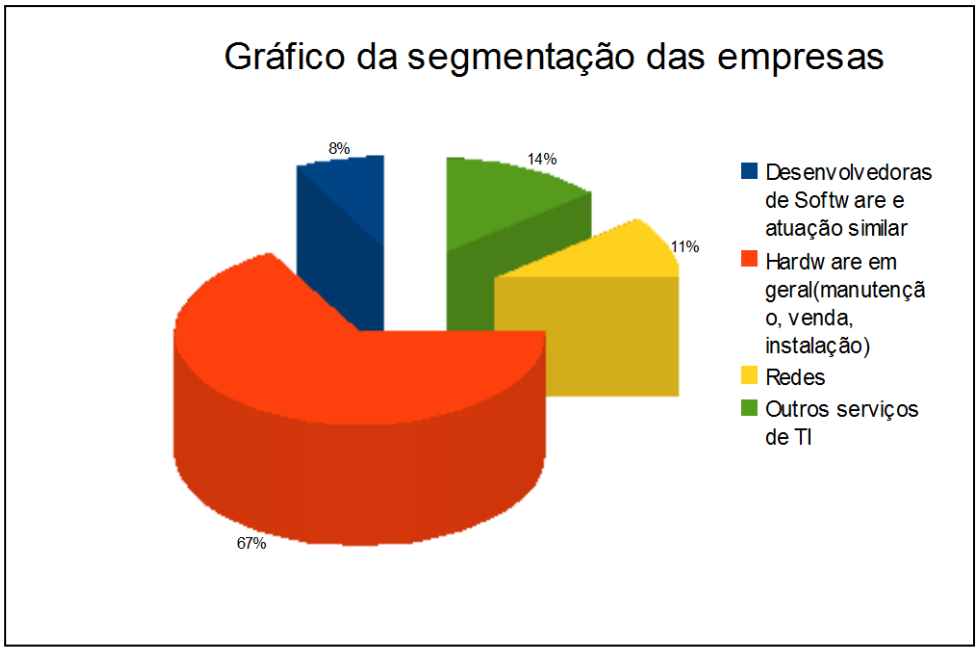

Figura 1 – Gráfico da Segmentação do Mercado de Software – 2009 Fonte: Prefeitura de Vitória da Conquista(2009)

A seguir será apresentado um esboço sobre os modelos de desenvolvimento de software.

2.3 O Processo de Desenvolvimento de Software

Segundo Sommerville(2007), um processo de software é um conjunto de atividades que se direcionam a resultados, gerando assim um produto de software.

Identificado como um dos fatores mais importantes da Engenharia de Software, o processo possui um modelo de representação e este, por sua vez, segue um método..

Existem diversos modelos de processo e métodos, sendo que a relação entre modelo e método pode ser de um para um ou de um para muitos, ou seja, um modelo pede ser operacionalizado por uma ou mais métodos. Essa quantidade diversa de modelos e metodologias faz com que o processo de software não seja algo totalmente automatizado e também faz com que não exista um ideal.

> Sendo assim, um processo de desenvolvimento de software deve ser implementado de acordo com um modelo previamente definido, seguindo uma metodologia que se satisfaça às necessidades e objetivos existentes, e tudo isto deve servir de guia para a correta utilização dos métodos e das ferramentas, tendo sempre em mente que a camada básica é o foco na qualidade. (GIRAFFA, MARCZAK e PRIKLADNICKI, 2005, p. 3).

2.3.1 Métodos Tradicionais

Segundo Sommerville (2007), um método de desenvolvimento de software é um conjunto de práticas, que são regidas por fases e/ou etapas, que tem como finalidade um melhor gerenciamento e ordenação do processo de software.

Dentre os métodos existentes, as Tradicionais foram as precursoras. Também chamadas de pesadas ou orientadas a documentação, de acordo com Royce(1970), surgiram num momento em que o software era destinado, basicamente, a servir a *mainframes* e a terminais burros. Esse tipo de abordagem era importante para a época, já que o desenvolvimento de software necessitava ser rigidamente planejado antes de ser implementado, pois as alterações eram extremamente complicadas e custosas.

Os modelo tradicionais seguem os mesmo princípios metodológicos de engenharias como Civil e Elétrica, nas quais os projetos de desenvolvimento são divididos em 2 fases: o planejamento e, após este concluído, a construção.

Possuem como principais características:

- Divisão de etapas e fases, que englobam as seguintes atividades: Análise, Modelagem, Desenvolvimento e Teste;
- O foco principal na previsibilidade dos requisitos do sistema (Oliveira, 2004);
- Dentre os modelos que utilizam os exemplos tradicionais, o Modelo Clássico, ou de Cascata, é o que melhor representa os conceitos desse tipo de método e também é o mais utilizado.

Na atualidade, estes métodos perdem cada vez mais espaço, pois o seu uso se

tornou inadequado às novas tendências de projeto de software, às novas exigências de participação dos clientes e usuário no projeto como um todo, já que nesse tipo de método eles só são requisitados no início e no fim dos trabalhos, e às freqüentes necessidades de redução de custo e tempo e otimização do trabalho que, por conseqüência da dificuldade de iterações, de mudanças de requisitos e de alterações e verificações do produto no decorrer do projeto, são prejudicados nesses métodos.

Modelo Clássico (ou de Cascata)

O modelo de cascata foi o primeiro modelo de processo de software publicado e é o mais famoso dentre as metodologias tradicionais. De acordo com Pressman(2006), este modelo possui uma abordagem seqüencial e sistemática das atividades que vão da especificação de requisitos, passando pela modelagem e implementação, culminando na manutenção progressiva do software.

Por ser um modelo seqüencial e rígido, cada etapa tem que possui uma documentação padronizada que deve ser aprovada para que a próxima etapa possa ser iniciada.

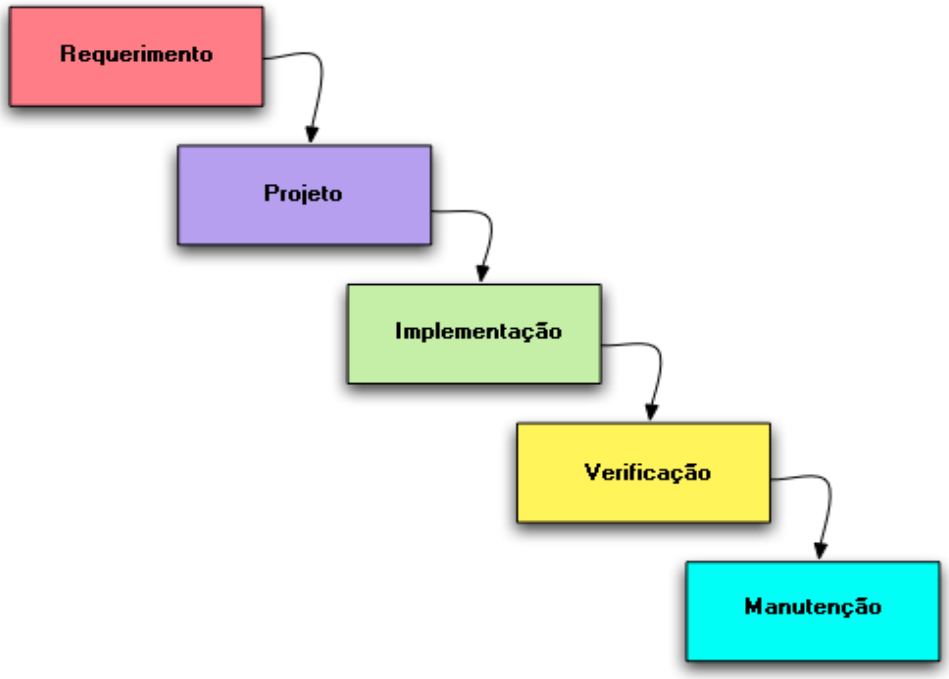

Figura 2 – Etapas do modelo de cascata

O modelo de cascata foi dominante no cenário de desenvolvimento de software até o início dos anos 90, apesar de ser muito criticado por utilizar a forma sequencial como base. Porém, um documento do *Standish Group* (1995), em 1995, feito a partir da verificação de 8380 projetos, mostrou ao universo dos desenvolvedores de software dados concretos dos problemas causados, na sua maioria, pela utilização do modelo cascata:

- Apenas 16,2 % dos projetos foram entregues respeitando os prazos, os custos e todas as funcionalidades especificadas. 31% foram cancelados e 52,7% foram entregues, mas com prazos alargados, custos maiores e não implementado com todas as funcionalidades previstas;
- Dentre os projetos que não foram entregues no prazo e com os custos predefinidos, o aumento do tempo de projeto foi de, na média, 222% e dos custos foi 189%;
- Ainda levando em consideração os projetos entregues em prazos e custos desrespeitados, o número de funcionalidades acopladas ao software foi de 61% das projetadas originalmente;
- Mesmo nos sistemas que foram entregues com prazos, custos e funcionalidades respeitados, se pôde observar que houve uma grande pressão nos desenvolvedores, fato que pode quadruplicar o número de erros.

> Hoje em dia, o trabalho de software é um ritmo rápido e sujeito a uma torrente sem fim de modificações (de características, funções e conteúdos de informações). O modelo em cascata é freqüentemente inadequado para esse tipo de trabalho. No entanto, pode servir como um modelo de projeto útil em situações nas quais os requisitos são fixos e o trabalho deve seguir até o fim de modo linear.(PRESSMAN, 2006, p. 39).

De acordo com Pressman(2006), os problemas mais encontrados nos projetos em que os modelos cascata são implementados são:

- A maioria dos projetos não segue um fluxo linear. Utilizar esse tipo de fluxo pode trazer confusões no decorrer do projeto, já que as iterações são feitas indiretamente;
- O modelo é pouco adaptável a mudanças de requisitos no decorrer do projeto. Ao utilizar esse modelo, é imperativo que se tenha os requisitos bem definidos no início do trabalho, fato que, quase sempre, é impraticável e inviável;

- Satisfação do cliente observando um protótipo resultante do projeto é quase impossível quando se utiliza esse modelo, pois um executável só é disponível no final do tempo do projeto;
- Tendo como base o estudo de Bradac e outros (1994), pode-se averiguar que esse método cria um tipo de "estado de bloqueio", no qual membros da equipe ficam esperando que outros terminem seus trabalhos para começar os seus, o que acarreta em grande perda de tempo e recursos.

Método Espiral

O modelo espiral foi arquitetado por Barry Bohem, em 1988, com o intuito de, segundo Presman(2006), unir o fator iterativo do modelo de prototipação com a rigidez e o aspecto sistemático do modelo de cascata, inserindo a um fator até então não utilizado, a análise de riscos. Este foi um dos primeiros modelos a consolidar a utilização das iterações nos projetos de software.

Apesar desta modelagem utilizar do modelo de cascata como um dos seus fundamentos, ela não está totalmente inserida nas metodologia clássicas, sendo considerado um modelo iterativo e incremental.

De acordo com Bohem(2001), o modelo espiral possui duas principais características: o modelo tem um abordagem cíclica, que aumenta o grau de definição e implementação do sistema ao passo que diminui os riscos, e possui marcos de ancoragem, que garante soluções exeqüíveis e satisfatórias ao sistema.

> Ele usa uma abordagem "evolucionária" à engenharia de software, capacitando o desenvolvedor e o cliente a entender e reagir aos riscos em cada fase evolutiva. O modelo espiral usa a prototipação como um mecanismo de redução de riscos, mas, o que é mais importante, possibilita que o desenvolvedor aplique a abordagem de prototipação em qualquer etapa da evolução do produto. Ele mantém a abordagem de passos sistemáticos sugerida pelo ciclo de vida clássico, mas incorpora-a numa estrutura iterativa que reflete mais realisticamente o mundo real.(LESSA, JUNIOR, 2008, p. 5)

Método RUP

O RUP, abreviatura de *Rational Unified Process*, é um processo de desenvolvido pela *Rational Software Corporation*, mas em 2003 foi adquirido pela IBM sendo, após esta data, chamado de IRUP. Este método é oriundo das teorias do UP, *Unifid Process*, e tem como intuito melhorar a produção e a qualidade do produto de software utilizando técnicas comerciais.

Apesar do RUP ser aplicável a classes variadas de projetos, por ser um método considerado "pesado" ele é preferencialmente utilizado a grandes equipes de desenvolvimento e a grandes projetos.

De acordo com Kruchten (2000), a arquitetura RUP e composta por, de forma primária, fases que representam a ênfase que é dada a cada uma das etapas do projeto em determinado período. São 4 as fases:

- Fase de concepção – essa fase possui os *wokflows* necessários para se chegar a um entendimento dos *stakeholders* (partes atuantes no projeto) com relação aos objetivos, arquitetura e planejamento;
- Fase de elaboração – serve como uma complementação da fase de concepção. Nela se verifica a documentação e levantamento dos casos de uso, revisa a modelagem de negócio e se inicia a confecção do manual ao usuário;
- Fase de construção – é nela que se inicia a produção dos códigos, dos testes, ou seja, o desenvolvimento físico do projeto;
- Fase de transição – nessa fase é feita o plano de implantação e entrega do software e também o plano de verificação de qualidade.

Além das fases, a arquitetura do modelo RUP apresenta o conceito de disciplinas que, segundo Neto(2004), servem para se alcançar uma organização lógica das atividades. Elas descrevem como as atividades, os papéis e a documentação devem ser geridos em cada iteração. São 9 as disciplinas: Modelagem de Negócio, Requisitos, Análise e Projeto, Implementação, Teste, Implantação, Ambiente, Configuração e Gerência de Negócio, Gerência de Projeto. As 6 primeiras são disciplinas da própria Engenharia de Software, já as 3 últimas são de suporte.

2.3.2 Metodologias Ágeis

Sob a visão de Pressman(2006), a engenharia de software ágil une uma filosofia, que aborda a satisfação do cliente, a motivação das pequenas equipes e a simplificação dos métodos de desenvolvimento, com diretrizes de desenvolvimento, que focalizam na entrega do produto e na maior ligação entre clientes e desenvolvedores. Essa união se torna importante porque cria uma atmosfera de satisfação das equipes, fato que acarreta em ganho de tempo e de qualidade de produção, e de maior cooperação entre todas as partes que compõe o projeto, obtendo mais êxito no produto final.

Apesar de serem desenvolvidas desde meados da década de 90 quando foi observada a ineficiência do modelo cascata, somente em 2001 que os então métodos leves foram conhecidos como métodos ágeis. Neste ano foi publicado o Manifesto Ágil que, além de modificar a nomenclatura dos métodos, reuniu os princípios e práticas destas metodologias de desenvolvimento. Tempos após a publicação do manifesto foi criada a *Agile Alliance*, uma organização sem fins lucrativos que promove o desenvolvimento ágil.

De acordo o manifesta da *Agile Alliance*(2010), existem 12 princípios para um método ágil eficiente:

Satisfação do Cliente;
Mudança dos Requisitos são bem-vindas;
Entrega constante do software já em funcionamento;
Atuação conjunta dos desenvolvedores e dos responsáveis pelo negócio;
Motivação da equipe;
Conversação cara a cara;
Funcionamento do software é medida primária;
Desenvolvimento sustentável;
Contínua atenção a excelência técnica e bom design;
Simplicidade;
Equipes auto-organizáveis;
Reflexão perante a equipe.

Comparando os métodos ágeis com os incrementais, pode-se verificar a maior

ênfase dada, pelos primeiros, às interações feitas em curtos períodos de tempos (em vez de meses, elas são feitas em semanas). Já se a comparação for feita com o método cascata as diferenças são ainda maiores, pois os ágeis prezam o desenvolvimento de pequenas partes de funcionalidades no decorrer do projeto para agregar mais valor ao mesmo.

Segundo Cohen e et. al. (2004), a melhor aplicabilidade dos métodos ágeis pode ser verificada a partir das seguintes perspectivas:

- Produto – os métodos ágeis são melhores quando os requisitos estão sempre em processo de mudança;
- Organizacional – nesta perspectiva, os métodos ágeis são apropriados quando os trabalhadores são confiantes, as decisões são tomadas por todos e a comunicação é facilitada;

Os métodos ágeis têm sido alvo de inúmeras críticas, principalmente pelo fato de serem considerados desorganizados. Levando em consideração trabalho de críticos McBreen(2003) e Bohen(2004), os principais pontos de críticas são:

- Falta de estruturação;
- Apoio somente em desenvolvedores com alto grau de conhecimento;
- Necessita de grandes mudanças culturais para sua implantação;
- Dificulta em processos contratuais.

A seguir apresentamos as características dos principais métodos ágeis

XP – Extreme Programming: Desenvolvido na década de 80 do século passado, mas só tendo os primeiros trabalhos publicados no final da década de 90, o XP é, segundo Pressman(2006), um método que tem como paradigma de uso principal a orientação a objeto e possui suas regras e práticas voltadas a quatro atividade de arcabouço: planejamento, projeto, codificação e teste. A partir dessas atividades, o XP adota os seguintes princípios básicos: *feedback* rápido, presumir simplicidade, mudanças incrementais, aderir a mudanças e a qualidade no trabalho.

O XP ainda possui as características de: ser focalizado no escopo do projeto e de incentivar o uso de controle de qualidade como variável do projeto.

DAS – Desenvolvimento Adaptativo de Software: Desenvolvido como método para construção de sistemas mais complexos, de acordo com Pressman(2006), o DAS focaliza na relações humanas e na auto-organização das

equipes. O autor ainda aborda que o ciclo de vida do DAS é composto por 3 etapas: especulação, onde se inicia o projeto; colaboração, onde os requisitos são conseguidos e a relação interpessoal é priorizada; aprendizado, implementação dos componentes do sistema que, conseqüentemente, acarreta num ganho de conhecimento individual e coletivo.

DSDM – Método de Desenvolvimento Dinâmico de Sistemas: Segundo a CS3 Consulting Service (2002), o DSDM é um método que tem como finalidade manter projetos de sistemas que necessitam serem feitos em curtos prazos. Para isso ele utiliza da prototipagem incremental.

O DSDM "define três diferentes ciclos iterativos – iteração do modelo funcional, iteração de projeto e construção e implementação – precedidos por duas atividades de ciclos de vidas adicionais – estudo de viabilidade e estudo de negócio"(PRESSMAN, 2006, pg. 74).

Pressman(2006) ainda cita que o modelo necessita de cronometragem a cada intervalo de tempo e sugere que se faça somente o necessário em cada iteração.

SCRUM: Desenvolvido na década de 90, este método, de acordo com Pressman(2006), focaliza o uso de padrões de processo de software para projetos de curto prazo, com requisitos mutáveis e negócio crítico. Os padrões adotados são os seguintes:

- Pendência – se trata de uma lista de requisitos do projeto que agregam valor ao negócio;
- Sprints – são unidades de trabalho que são necessárias para satisfazer um dos itens da lista de pendências em um determinado especo de tempo;
- Reuniões Scrum – reuniões curtas e diárias, que discorrem sobre as atividades já feitas, os problemas a serem ultrapassados e a expectativa de realização de trabalho das equipes.

FDD – Desenvolvimento Guiado por Característica: Criado inicialmente para ser um modelo de processo mais prático, o FDD foi modificado para ser ágil e adaptativo e aplicável preferencialmente a projetos de médio e grande porte.

Este método é "algo mais "formal" do que os outros métodos ágeis, mas ainda mantém agilidade para concentrar a equipe de projeto no desenvolvimento das características – funções valiosas para os clientes que podem ser implementadas em duas semanas ou menos"(PRESSMAN, 2006, p. 74). A seguir será apresentado um

breve estudo sobre os processos de gerenciamento de software, abordando a qualidade e os modelos de gerência.

3. O PROCESSO DE GERENCIAMENTO DE SOFTWARE

3.1. Qualidade de Software

Segundo a norma ISO 9000(2008), qualidade é uma medida que avalia o nível que um conjunto de características de um produto, processo ou sistema pode alcançar em comparação ao que foi estipulado inicialmente. Essa definição mais genérica de qualidade corrobora as definições mais específicas, dentre as quais estão as de qualidade de software.

Pressman (2006) utilizou de um conceito interessante para definir a qualidade de software quando disse que a mesma era uma satisfação de requisitos, de normas de desenvolvimento e características implícitas que são esperadas por todo o software desenvolvido profissionalmente.

Outra situação a ser levada em consideração para se entender e buscar a qualidade de software é a relação da mesma com a qualidade do processo de desenvolvimento, de acordo com Arouk(2001). Isso faz com que não se prime somente pela perfeição do produto final, mas também de suas etapas de concepção, ou seja, deve-se dar atenção necessária para as etapas intermediárias de criação de software para que o produto alcançado seja mais condizente como desejado.

Os fatores que caracterizam um software de qualidade são aqueles que funcionam de maneira a satisfazer necessidades implícitas e explícitas inerentes aqueles produto.

A tabela 3 apresenta os 6 grupos de atributos de qualidade de software segundo a norma ISO/IEC 9126 (2003) Atributos de qualidade são:

Tabela 3 – Características de qualidade de software Fonte: ISO/IEC 9126(2003)

CARACTERÍSTCA	SUB-CARACTERÍSTICA	DESCRIÇÕES DAS SUB-CARACTERÍSTICAS
Funcionalidade	Adequação	Medir o quanto as funcionalidades são adequadas aos usuários
	Acurácia	Representa o quanto o software pode fornecer resultados de acordo ao que foi solicitado
	Interoperabilidade	Representa como o software pode interagir com outro
	Segurança de acesso	Proteger as informações do sistema, só as apresentando a quem for de direito
	Conformidade*	Capacidade de se adequar a padrões e leis
Confiabilidade	Maturidade	Evitar falhar decorrentes de problemas no sistema
	Tolerância de Falhas	Manter o software funcionando mesmo após apresentar defeito
	Recuperabilidade	Capacidade do software de recuperação pós-falha
Usabilidade	Inteligibilidade	Facilidade de ser entendido pelo usuário, que pode identificar se o software pode satisfazer suas necessidades
	Apreensibilidade	Facilidade de aprendizado do sistema para seus usuários
	Operacionalidade	Facilidade do software de se tornar operável ao seu usuário
	Atratividade	Medida que caracteriza quanto o software pode ser atrativo ao seu público alvo
Eficiência	Comportamento/Tempo	Avalia o tempo de resposta com relação ao que foi especificado
	Utilização de Recursos	Mede como o sistema usa os recursos disponíveis
Manutenibilidade	Analisabilidade	Identificação de problemas e suas causas
	Modificabilidade	Mostra como o software pode ser modificado
	Estabilidade	Avalia a capacidade do software não apresentar efeitos colaterais após modificações
	Testabilidade	Avalia a capacidade de testar o software modificado
Portabilidade	Adaptabilidade	Identifica como o software se adapta a qualquer ambiente, sem precisar de ajustes adicionais
	Capacidade de Instalação	Mostra a facilidade que o software tem em ser instalado
	Coexistência	Identifica como o software convive com outros instalados na mesma máquina
	Capacidade de Substituir	Capacidade do sistema de substituir outro especificado
* a conformidade se encaixa em todas as características		

3.1. Qualidade de Processo de Software

Um processo, em âmbitos gerais, é um conjunto de ações e regras estipuladas para se conseguir alcançar uma determinada finalidade. Na computação, mas especificamente na Engenharia de Software, esse conjunto de ações e regras, parcialmente ordenado, tem uma meta principal: entregar um produto (software) final,

cumprindo prazos e servindo às necessidades do usuário.

Segund Pressman (2006, p. 16),“ ...definimos processo de software como um arcabouço para as tarefas que são necessárias para construir um software de alta qualidade...”

Como é levantado nas próprias definições, um processo é constituído de tarefas, ações, atividades. Para Schwartz (1975), um processo de software é composto por 4 fases principais: Especificação de Requisitos; Projeto do Sistema; Codificação; Verificação e Integração.

Inseridas nessas fases, existem atividades relacionadas de forma que se consiga uma harmonia e um resultado satisfatório no que se deseja obter a partir da aplicação do processo. Levando em consideração o custo, o desprendimento de mão-de-obra e o tempo gasto para por em prática essas atividade é que se sustenta a necessidade dos processos de softwares passarem eles próprios por determinados tipos de processos, os avaliativos e de controle de qualidade.

De acordo com Oliveira(2007), o passo a passo para se conseguir um processo de software mais aprimorado deve conter as seguintes características:

- Estudar os processos já existentes e que se equivalem ao que se deseja construir. Após o passo de análise citado anteriormente, deve-se verificar as melhorias que poderão ser feitas no modelo criado para que o processo possa produzir um produto final de maior qualidade.
- A inserção de novos métodos, procedimentos e ferramentas necessita de um tempo para implantação e de uma monitoramento adaptativo com os outros métodos, procedimentos e ferramentas já existentes na empresa.
- Continuando no estágio de inserção de mudança nos processos, o treinamento para lidar com essas mudanças é algo primordial, pois o resultado de artefatos novos utilizados por pessoal mal treinado pode ser o oposto ao desejado.
- Por fim, as mudanças devem passar por um ajuste final para se tornarem, de fato, parte integrante do processo de software.

3.2. Gerenciamento de Projeto

Segundo o *Project Management Institute* (PMI) (2000), um projeto é um

empreendimento único, com definição de começo e término, utilizando de recursos limitados e sendo conduzido por pessoas, visando atingir metas pré-definidas dentro de parâmetros de prazo, custo e qualidade.

A base de qualquer atividade em que se envolva engenharia está no seu projeto. Daí a necessidade de se compilar as práticas que conduzem a uma fomentação mais correta de um projeto em uma aplicação única, a Gestão ou Gerência de Projetos.

Sobre a ótica do PMI(2000), Gerência de Projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas para se criar atividades que atinjam o desejado pela construção do projeto.

> O gerenciamento de projetos evoluiu ao longo dos anos. Hoje, mais do que uma disciplina técnica e misteriosa, o gerenciamento de projetos inclui uma série de princípios cujo objetivo é fornecer uma abordagem estruturada para a tomada das decisões diárias que mantêm um negócio em andamento, mesmo que seja um pequeno negócio, ou um laboratório. (PORTNY;AUSTIN, 2002)

Segundo a ABNT (2001), um projeto é “um processo único, consistindo de um grupo de atividades coordenadas e controladas com datas para início e término, empreendido para alcance de um objetivo conforme requisitos específicos, incluindo limitações de tempo, custo e recursos.”

Segundo Vargas(2005), há na literatura diversas visões de como o ciclo de vida de um projeto deve ser divido, visões essas que podem assumir de cinco a nove fase. Apesar das inúmeras visões, o autor focaliza cinco fases características:

- Iniciação: Fase inicial do projeto, na qual é conhecida a problematização a ser resolvida pelo mesmo.
- Planejamento: Fase que detalha tudo o que será realizado pelo projeto (cronogramas, atividades, alocação de recursos, análise de custos etc.).
- Execução: Fase que põe em prática tudo o que foi levantado nas etapas anteriores. Os erros das fases passadas são determinantes para o bom andamento desta.
- Monitoramento e Controle: Fase que acontece paralela às de Planejamento e Execução. Tem com objetivo acompanhar e controlar o que esta sendo feito no projeto e propor correções para as anomalias que acontecerem no decorrer do mesmo.

- Encerramento: Nessa fase é feita uma avaliação da execução dos trabalhos, os livros e documentos são encerrados e os erros são discutidos para se conseguir algum aprendizado na tentativa de não perpetuá-los.

De acordo com Vargas(1998), o conhecimento das fases do ciclo de vida de um projeto acarreta em algumas vantagens: saber o que foi ou não feito pelo projeto, avaliar previamente a progressão do projeto até determinado momento, observar a situação do projeto em determinado momento e identificar mudanças em finais de fases.

3.2.1. Gerenciamento de Projetos de Software

Segundo Pressman(2006), a gerência de projeto de software é focalizada, nos quatro Ps: pessoal, produto, processo e projeto. Cada qual tendo seu peso na evolução da engenharia do software.

O direcionamento e a atenção na questão do pessoal, criando um modelo de maturidade para este tipo de gestão, é importante "para melhorar a capacidade de organizações de software desenvolverem aplicações cada vez mais complexas, ajudando-as a atrair, desenvolver, motivar, dispor e reter o talento necessário para melhorar sua capacidade de desenvolvimento de software "(CURTIS et. al., 1995).

O produto, ou o software em si, tem, por questões de boa prática, que ser um dos primeiros aspectos a ser analisado por uma equipe de gestão de projeto de software. Segundo Pressman(2006, pg. 490), a determinação do escopo do software é a primeira atividade a ser levantada pela equipe. Apesar do impasse que existe entre a precisão de se ter um plano concreto dos requisitos, das atividades, das vias indispensáveis para se construir determinado software e a variação, a inconstância dos requisitos e necessidades para essa mesma construção, é unânime o fato de se precisar analisar a problematização, o próprio produto e seu escopo, no início de cada projeto.

O modelo de processo deve ser bem definido pela equipe de gerência, para ser bem trabalhado.

De acordo com Boehm(1996), uma equipe de gestão de projeto precisa ter uma organização que torne o projeto o mais simples possível. O autor sugere um princípio, envolvendo uma série de questionamentos, que caracterizam o projeto e seu plano resultante. Esse princípio é chamado de W5H2:

- Porque o sistema está sendo desenvolvido?(*Why*);
- O que precisa ser feito?(*What)*;
- Quando deve ser feito? (*When*);
- Quem é responsável por cada função? (*Who*);
- Onde os responsáveis se localizam na organização? (*Where*);
- Como será a condução do trabalho, em âmbitos técnicos e gerenciais? (*How*);
- Quanto é necessário de cada recurso? (*How much*).

> Especificamente, a área de gerência de projetos de *software* possui particularidades que dificultam ainda mais o gerenciamento, tais como: mudança da tecnologia, rodízio de pessoal que possui conhecimento específico sobre a tecnologia e a intangibilidade do *software*. Este último, torna necessário a criação de artefatos e documentações para avaliar o *software* e, que, muitas vezes, não corresponde ao *software* realmente implementado. (ENANI et. al., 2006, pg. 55)

Os projetos também possuem características que envolvem as incertezas de seu resultado final e do caminho para chegar ao mesmo e o grau de complexidade com relação às variáveis que estão infiltradas em seu processo.

3.2.2. Métricas para Gerenciamento de Projeto de Software

Segundo Campêlo(2002, pg.47), as métricas são de grande valia para uma boa qualidade no desenvolvimento de software pois os custos e prazos estimados são decisivos para o alcance das expectativas do cliente.

Para Pressman(2006, pg.500), exitem dois tipos principais de domínios nos quais são utilizadas métricas: métricas de processo, que são utilizadas em todo o projeto e tem como objetivo apontar indicadores de processo que conduzem ao aperfeiçoamento do processo de software; métrica de projeto, que permite, ao gerente do projeto, avaliar a situação do projeto, verificar riscos, descobrir possíveis pontos críticos, avaliar a equipe etc.

Os dados obtidos através das métricas proporcionam maior conhecimento dos acontecimentos do projeto e dos processos. A visibilidade também proporcionada cria ajustes para os caminhos mais corretos a serem seguidos. Esses dados adquirem

importância até mesmo para serem usados em trabalhos futuros como pontos de acertos que poderão ser copiados ou aprimorados ou de erros que certamente serão descartados, corrigidos ou modificados.

As métricas podem ser qualitativas, que levam em consideração questões como confiabilidade, usabilidade e integridade, as quantitativas, que medem partes físicas do software como número de linhas de código e pontos por função.

O *Software Engineering Institute* (SEI) (2002), criou outros 2 tipos de classificação para as métricas:

- Básica: medida que é obtida diretamente em processos, produtos, pessoas etc.
- Derivada: medida obtida através de medidas básicas.

3.2.3 Modelos de Processo de Gerência de Software

Segundo Takano(2006), um modelo de processo de software é um representação abstrata dos elementos que compõe o processo de criação de um software, sendo que estes elementos podem ser utilizados por pessoas ou por máquinas.

Continuado a acompanhar o raciocínio de Takano, pode-se observar que as descrições usuais de ciclo de vida de software não oferecem uma visão mais detalhada dos processos envolvidos e nem das atividades que deverão ser executadas pelo atores do processo, por isso que a modelagem vem como um auxiliar na apresentação esmiuçada dessas atividades.

Dentre os modelos de processo de gerenciamento de software destacamos:

- PSP(*Personal Software Process*), ou Processo Pessoal de Software, é aquele em que o desenvolvedor utiliza para si mesmo, como medidor e analista de seu trabalho e como ferramenta para auxiliar na incorporação de novas técnicas e práticas. “...Além disso, o PSP torna o profissional responsável pelo planejamento do projeto (por exemplo, fazer orçamentos e cronogramas) e dá poder ao profissional para controlar a qualidade de todos os produtos do trabalho de software que são desenvolvidos.”(PRESSMAN, 2006, p. 29). De acordo com Pressman(2006), são definidas 5 atividades para o PSP: planejamento, projeto de alto nível, revisão do projeto de alto nível, desenvolvimento e pós-conclusão. As técnicas abordadas pelo PSP,

além de ajudar o profissional no seu próprio crescimento, servem como auxiliares na preparação de equipes.

- De acordo com Humphrey(2006), o TSP (*Team Software Process*), ou Processo de Equipe de Software é uma *framework* de processo de criação de software que se baseia no trabalho em equipe, dando enfoque ao planejamento e gerenciamento de projetos. Segundo o autor, os objetivos do TSP são os seguintes: formar equipes autodirigidas, que planejem e monitorem seu trabalho; evidenciar aos gerentes como monitorar e motivar suas equipes; acelerar o processo de aperfeiçoamento de software; ajudar no ensino universitário das habilidades de equipes industriais. O TSP segue alguns princípios como: equipes necessitam ter objetivos comuns, reutilização de técnicas de sucesso pode resolver problemas, projetos precisam ter um planejamento detalhado, papéis devem ser bem estabelecidos, a comunicação entre os membros da equipe deve ser aberta, entre outros.

A seguir é apresentado características de alguns modelos de gerência de software.

3.4. CMMI

> O CMMI (*Capatility Maturity Model Integration*) é um modelo de referência desenvolvido pelo SEI (*Software Enginnering Institute*) que visa orientar as organizações de software na implementação de melhorias contínuas em seu processo de desenvolvimento... (CARVALHO, 2007, p. 34).

Esse modelo possui atividades necessárias a maturidade de disciplinas como Engenharia de Software, Engenharia de Sistemas, entre outras.

> No CMMI uma organização pode optar por dois enfoques para melhorar os seus processos: capacitação de uma determinada área de processo ou maturidade da organização como um todo. O suporte para tais enforques é feito através da representação por estágio e da representação contínua.(COUTO, 2007, p. 4)

Os processos abordados pelo CMMI são estruturados em 4 áreas de conhecimento: Engenharia de Sistemas, Engenharia de Software, Desenvolvimento de Processo e Produto Integrado e Contrato com Fornecedores. Essas disciplinas englobam desde o

desenvolvimento do sistema como um todo até a aquisição de produtos através de terceiros.

Segundo as regras do CMMI versão 1.1(2002), os processos encontrados em determinada área de processos, quando utilizados coletivamente, acarretam no alcance de objetivos determinantes para a melhoria da área em que se encontram. As 25 áreas de processo, ainda considerando o CMMI versão 1.1, são: Foco no Processo Organizacional; Definição de Processo; Treinamento Organizacional; Desempenho do Processo Organizacional; Desenvolvimento e Inovação Organizacional; Planejamento de Projeto; Controle e Monitoramento de Projeto; Gerência de Contrato de Fornecedores; Gerência de Projeto Integrado; Gerência de Riscos; Gerência Quantitativa de Projeto; Gerência de Requisitos; Gerência de Requisitos; Solução Técnica; Integração de Produto; Verificação; Validação; Gerência da Configuração; Garantia de Qualidade de Produto e Processo; Medições e Análises; Resolução e Análise de Decisão; Resolução e Análise das Causas; Gerência Integrada de Fornecedores; Integração do Time; Ambiente Organizacional para Integração.

De acordo com Santos(2007), os níveis de maturidade abordados pelo CMMI são úteis tanto para melhor projeção e manutenção do software como também na melhora nos processos das empresas de software.

A tabela 4 apresenta os níveis de maturidade e suas descrições no modelo CMMI.

Tabela4: Quadro dos Nívis de Maturidade segundo o CMMI Fonte: CMMI(2002)

Níveis de Maturidade – CMMI	
NÍVEIS	DESCRIÇÃO
Nível 0 – Incompleto	sem área de processo ou com área de processo mas que não atinge as especificações para o próximo nível de maturidade
Nível 1 – Executando	especificações da área de processo definidas
Nível 2 – Gerenciado	nível em que os processos estão condizentes com a política da empresa, há atuação dos interessados e produtos e tarefas são monitorados, controlados revisados
Nível 3 – Definido	além de abranger os avanços do 2, neste nível o processo é feito para o conjunto-padrão de processos da empresa
Nível 4 – Quantitativamente Gerenciado	áreas de processo são avaliados e melhorados por meio do uso de medições e atividades avaliativas
Nível 5 – Otimizado	além de abranger os avanços do 4, neste nível os processos são otimizados, utilizando métodos quantitativos, para satisfazer o cliente

O CMMI é representado de 2 formas: por estágio e contínua. De acordo com Carvalho (2007) a abordagem por estágio é a mais utilizada e mais fácil de ser implementada. Ela inicia com processos básicos de gerenciamento e evolui por vias pré-definidas, sendo que os níveis de maturidade são interdependentes. Já a abordagem contínua é mais complexa de ser implementada, porém ele é mais flexível, pois por meio dela pode-se escolher qual área de processo pode ser melhorada, fazendo com que a empresa de enfoque em áreas mais necessitadas

3.5. PMBOK

O *Project Management Institute* (PMI) que, segundo o PMI Brasil (2010), é a mais importante instituição associativa mundial voltada ao gerenciamento de projeto e que tem como objetivo principal o avanço na prática, ciência e profissionalização da gerencia de projeto em todo o planeta, publicou no ano de 1987 o primeiro documento representativo dos padrões de gerenciamento de projeto, o *Project Management Body of Knowledge* (PMBOK *Guide*). Este documento inicial foi modificado nos anos de 1996 e 2000, sendo que a atualização vigente, a terceira edição, do PMBOK *Guide* é do ano de 2004.

> ...Ele descreve o conjunto de conhecimentos geralmente aceitos dentro da profissão de gerenciamento de projetos. ‘Geralmente aceito’ significa que os conhecimentos e práticas descritas aplicam-se à maioria dos projetos, na maioria das vezes e que há amplo consenso sobre seu valor e aplicação. O objetivo geral do PMBOK é oferecer o léxico comum dentro da profissão e prática de gerenciamento de projetos para o diálogo e a escrita sobre Gerenciamento de Projeto.(STANLEIGH, 2006, p. 2).

Um dos pontos a ser levado em consideração no guia PMBOK é que ele é um documento de caráter totalmente genérico, ou seja, ele consegue auxiliar tanto aos projetistas em projetos de indústrias da construção, quanto a projetistas de software.

O guia é organizado em função de processos, semelhantes aos do CMMI. Na versão atual contabiliza-se 44 processos divididos em cinco grupos e em nove áreas de conhecimento.

Os grupos de processo de software são os seguintes: iniciação, planejamento, controle, execução e encerramento.

Segundo Sotille(2006), os processos se referem a aspectos abordados na gerência de projeto e, para que o projeto alcance sucesso, todos esses grupos devem constar na execução do mesmo.

Além dos 5 grupos, os processos se encaixam em 9 áreas de conhecimento, que representam os setores de conhecimento técnicos em gerência de projeto fundamentais para o desempenho perfeito das ações que comporão todos os processos.

As principais áreas de conhecimento e práticas abordadas nas gerências de projeto são: Gerencia de Integração do Projeto, de Escopo do Projeto, de Tempo do Projeto, de Custos do Projeto, de Qualidade do Projeto, de Recursos Humanos do Projeto, de Comunicação do Projeto, do Risco do Projeto e da Contratação do Projeto.

> Escopo, Tempo, Custos e Qualidade são os principais focos para o objetivo de um projeto: entregar um resultado de acordo com o escopo, o prazo e o custo definidos, com qualidade adequada. Recursos Humanos e Aquisições são os insumos que movem um projeto. Comunicações e Riscos são elementos aos quais deve haver sempre atenção e tratamento constantes em um projeto. E Integração abrange a orquestração de todos estes aspectos. (D´ÁVILA, 2007)

3.6. SWEBOK

O *Software Engineering Body of Knowledge*, ou SWEBOK, foi concebido pelo IEEE (*Institute of Electrical and Electronics Engineering*) para servir como artefato de referência para assuntos pertinentes a área de Engenharia de Software.

Para alcançar a finalidade de organização do vasto material e das várias técnicas utilizadas, segundo o SWEBOK Guide (2004), o guia desmembra a Engenharia de Software em 10 Áreas de Conhecimento: Requisito de Software; Projeto de Software; Construção de Software; Teste de Software; Manutenção de Software; Gerência de Configuração de Software; Gerência de Engenharia de Software; Processo de Engenharia de Software; Ferramentas e Métodos da Engenharia de Software; Qualidade de Software.

Utilizando desse tipo de metodologia, o SWEBOK constitui um arcabouço básico para a profissionalização em Engenharia de Software, pois ele promove uma visão consistente da própria engenharia com relação ao mundo e caracteriza o conteúdo de suas disciplinas.

Segundo Lessa e Junior (2008), o SWEBOK aborda outras disciplinas que, apesar dele não considerá-las como pertencentes à Engenharia de Software, são auxiliares para a mesma. São elas: Engenharia da Computação, Ciência da Computação, Gerenciamento, Matemática, Gerência de Projetos, Gerenciamento de Qualidade, Ergonomia de Software e Engenharia de Sistemas.

3.7. MPS-BR

O MPS-BR, ou Melhoria de Processo de Software Brasileiro, é um programa criado em 2003 e coordenado pela Associação para Promoção de Excelência em Software Brasileiro (SOFTEX), tendo grande apoio de órgãos governamentais, como o Ministério da Ciência e Tecnologia (MCT), a Financiadora de Estudos e Projetos (FINEP), o Banco Interamericano de Desenvolvimento (BID), e ajuda de universidades para o seu desenvolvimento.

> Uma das metas do programa MPS.BR é definir e aprimorar um modelo de melhoria e avaliação de processo de software, visando preferencialmente às micro, pequenas e médias empresas, de forma a atender as suas necessidades de negócio e ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software.(MPS-BR: Guia Geral, 2009, p. 12).

Segundo a SOFTEX(2009), no Guia Geral, como ferramentas para a melhoria e a avaliação de processos de software, o MPS-BR estabelece um modelo de processos de software, um processo e método de avaliação de processos e um modelo de negócio. O primeiro artefato é a modelagem do processo, propriamente dito, de criação de software, os dos seguintes fornecem sustentação para se ter certeza de que o MPS-BR está sendo aplicado de forma correta e o último é um modelo que proporcionam apoio às empresas que resolvem adotar este método.

O modelo MPS-BR é compatível com o CMMI e ainda possui sua base técnica fundamentada pelas normas ISO/IEC 12207:2008 e ISO/IEC 15504.

De acordo com a SOFTEX(2009), o modelo possui duas principais metas para serem alcançadas em médio e longo prazos. São elas:

- A meta técnica – esta se refere à criação e aprimoramento do modelo MPS, com resultados apresentados através de guias do modelo, da criação de instituições implementadoras e avaliadoras, entre outros.
- A meta mercantil – esta se refere a grande adoção do modelo por empresas de todo o pais, com intervalo de tempo e custo condizentes as realidade, tanto em pequenas e médias empresas quanto em grandes estatais, por exemplo. Os resultados esperados vêem através da implantação do modelo de negócio MN-MPS, de cursos e provas e da publicação de resultados de empresas avaliadoras.

O modelo MPS-BR está divido em 3 partes: Modelo de Referência (MR-MPS), Modelo de Avaliação (MR-MPS), Modelo de Negócio (MN-MPS).“Cada componente é descrito por meio de guias e/ou documentos do modelo MPS”(SOFTEX, 2009, p. 13).

O guia MPS-BR, segundo a SOFTEX(2009), descreve o modelo MR-MPS, ou Modelo de Referência, como uma descrição dos níveis de maturidade empresarial. Estes níveis são obtidos através da compilação entre os processos, que seguem a forma de referência de processo mostrada no ISO/IEC 15504-2, e suas capacidades de alcançar os objetivos de negócios, tanto atuais quanto futuros.

O modelo MPS-BR possui 7 níveis de maturidade que “estabelecem patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos na organização. O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos

O guia MPS-BR mostra que cada nível de maturidade aborda uma área de processo, na qual são analisados três tipos de processos: os fundamentais, que tem como

exemplo a aquisição, desenvolvimento de requisitos, integração de produto, instalação de produto, entre outros, os organizacionais, como a gerência de projetos, a análise de decisão e resolução, a gerência de risco, e os de apoio, que tem como exemplos a validação, a medição e treinamento.

A capacidade dos processos também é vista pelos níveis de maturidade. Essa capacidade, de acordo com o guia, é uma via de demonstração do grau de refinamento e institucionalização com que o processo é posto em prática na organização.

Segundo o guia, os atributos de processo (AP) são as descrições dos níveis de capacidade de processo. São eles:

- AP 1.1 – O processo é executado: Mede o quanto o processo atinge a finalidade;
- AP 2.1 – O processo é gerenciado: Mede o quanto a execução do processo é gerenciada;
- AP 2.2 – Os produtos de trabalho do processo são gerenciados: Mede o quanto os produtos dos processos são corretamente gerenciados;
- AP 3.1 – O processo é definido: Mede o quanto um processo padrão é sustentado para auxiliar na implementação do processo definido;
- AP 3.2 – O processo é implementado: Mede o quanto um processo padrão é implementado como um definido para atingir seus resultados;
- AP 4.1 – O processo é medido: Mede o quanto os artefatos resultantes de medições são utilizados para assegurar que o processo atingiu seus objetivos e foi coerente com os objetivos do negócio definidos para o mesmo;
- AP 4.2 – O processo é controlado: Mede o controle do processo com relação às estimativas. A partir de um controle correto das estimativas pode-se conseguir um processo estável e capaz dentro de suas limitações;
- AP 5.1 – O processo é objeto de melhorias e inovações: Mede a identificação de mudanças do processo necessárias para resolução de problemas, de defeitos e para alcançar enfoques inovadores para a definição e implementação do processo.

A seguir será mostrado um quadro, adaptativo da Tabela 8-1-Níveis de Maturidade MR-MPS do Guia Geral MPS-BR da SOFTEX(2009), em que os níveis de processos, seus processos e seus atributos de processo são relacionados:

Tabela 5: MR-MPS: Níveis de Maturidade, Processo e AP(s) Fonte: SOFTEX (2009)

MR-MPS: NÍVEIS DE MATURIDADE, PROCESSOS E AP(s)	
Nível	Processos
G	Gerência de Requisitos Gerência de Projetos
F	Medição Garantia de Qualidade Gerência de Portifólio de Projetos Gerência de Configuração Aquisição
E	Gerência de Projetos (evolução) Gerência de Reutilização Gerência de Recursos Humanos Definição de Processo Organizacional Avaliação e Melhoria de Processo Organizacional
D	Verificação Validação Projeto e Construção de Produto Integração do Produto Desenvolvimento de Requisitos
C	Gerência de Riscos Desenvolvimento de Reutilização Gerência de Decisões
B	Gerência de Projetos (evolução)
A	Otimização

O MPS-BR possui ainda aas seguintes documentação/guias:

MA-MPS

Segundo a SOFTEX(2009), que descreve tal fato no MPS-BR: Guia de Avaliação, o Modelo de Avaliação, ou MA-MPS, é um processo, um método criado para se obter uma avaliação abrangente dos processos de software de uma empresa, independente do tamanho e aplicável a qualquer domínio das organizações de software. Esse processo avaliativo está intimamente ligado com o nível de maturidade proporcionado pelo MR-MPS, pois o propósito principal do MA-MPS é observar e quantificar a maturidade organizacional no que diz respeito aos processos de software. Como resultado da execução do processo, segundo a SOFTEX(2009), o MA-MPS

alcança: a obtenção dos dados e informações que caracterizam os processos utilizados pela organização; o grau do alcance dos resultados e do êxito do processo para atingir seu objetivo; atribuição de um nível de maturidade MR-MPS para a empresa.

MN-MPS

O Modelo de Negócio para Melhoria de Processo de Software, ou MN-MPS, consiste em um documento que contém as estratégias para a implantação do modelo, para a escolha e capacitação de pessoal para trabalhar como consultores do MR-MPS e para serem avaliadores, além de contar listas de consultores e avaliadores já treinados e aprovados.

4. ANÁLISE

4.1 Metodologia

Para o levantamento do nível de maturidade das empresas desenvolvedoras de software de Vitória da Conquista, inicialmente foi estipulada a metodologia de pesquisa a ser utilizada.

A metodologia escolhida foi baseada em uma das fases metodológicas usadas pelas instituições implementadoras MPS-BR como auxiliar no processo de diagnostico dos procedimentos de software das empresas: a aplicação de uma planilha de verificação.

Servindo as necessidades do projeto, a planilha de verificação de aderência proposta pelo guia de Avaliação MPS-BR (SOFTEX, 2009) foi adaptada para se transformar em um questionário avaliador (ver Anexo A). Foi utilizado o processo de amostragem para o estudo das respostas. Como a análise da maturidade tem um caráter inicial, as questões levantadas correspondem aos aspectos de gerenciamento de projeto e gerenciamento de requisitos abordados pelo nível G do MPS-BR.

4.2. Amostragem

O questionário avaliador foi aplicado em somente 15 empresas, pois das empresas desenvolvedoras ativas, somente essas 15 estavam disponíveis para comunicação através do meio utilizado, Internet. O meio de contato com as empresas pesquisadas necessitou ser a rede mundial de computadores por motivos de incompatibilidade de tempo e de indisponibilidade de translado do pesquisador. Somente 5 das empresas remeteram as respostas. Apesar de a amostragem possuir uma quantidade baixa, a representatividade desses números contém uma significativa verossimilhança com a realidade dos fatos. Os tópicos seguintes mostram a análise categorizada dos resultados. Para essa análise foi levado em consideração principalmente os dois pontos principais que caracterizam a implementação do nível G: a mudança de cultura organizacional e a definição de conceito de projeto.

Outro fator relevante da amostragem é que as empresas pesquisadas eram

somente privadas e com finalidade única de desenvolver softwares.

4.3 Levantamento dos resultados sobre Gerenciamento de Projeto

Segundo o Guia de Implementação MPS-BR nível G (SOFTEX, 2009), o processo de Gerência de Projeto, ou GPR, envolve alguns tipos de atividades, dentre elas: o desenvolvimento de um plano global de projeto, obtendo o cumprimento do mesmo até o final do projeto e obtendo, também, o conhecimento dos processos que envolvem o projeto, para que as medidas preventivas e as medidas de correção sejam aplicadas de forma correta e eficaz. Tendo em vista isso, os assuntos abordados as questões/respostas foram os seguintes:

Processo gerenciado

Em 80% das empresas de software pesquisadas, os processos de desenvolvimento são considerados parcialmente gerenciados e os outros 20 % consideram totalmente gerenciados.

Escopo do projeto

Os escopos dos projetos de desenvolvimento de software, para 40% das empresas pesquisadas, não é bem definido para nenhum dos projetos. Já para outros 40%, o escopo é definido ou parcialmente definido para a maioria dos projetos e, somente para 20% das empresas consideram boa definição de escopo para 100% de seus projetos. O gráfico da figura 3 mostra esse valores.

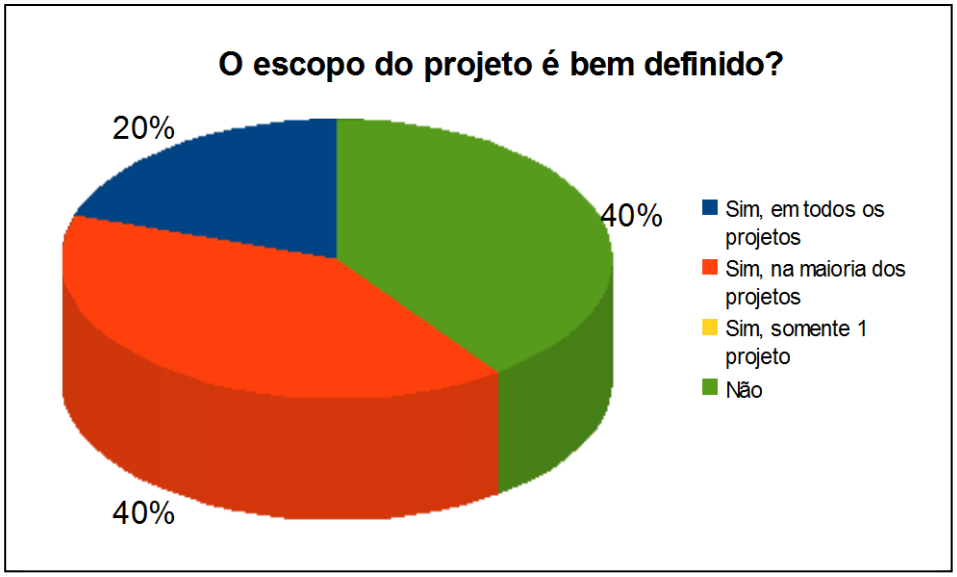

Figura 3: O escopo do projeto é bem definido?

Tarefas e produtos dos processos

A grande maioria das empresas (80%) não utilizam nenhum método de dimensionamento de tarefas e produtos necessários a implementação dos projetos, como a Análise de Ponto Por Função.

Modelo de Ciclo de Vida

Como é evidenciado no esquema da figura 4, a utilização de um modelo pré-determinado de Ciclo de Vida de projetos é ainda pouco difundida nas empresas, pois 60% das mesmas utilizam um modelo, mas não em todos os seus projetos, e os outros 40% não utilizam nenhuma forma de modelagem de ciclo de vida.

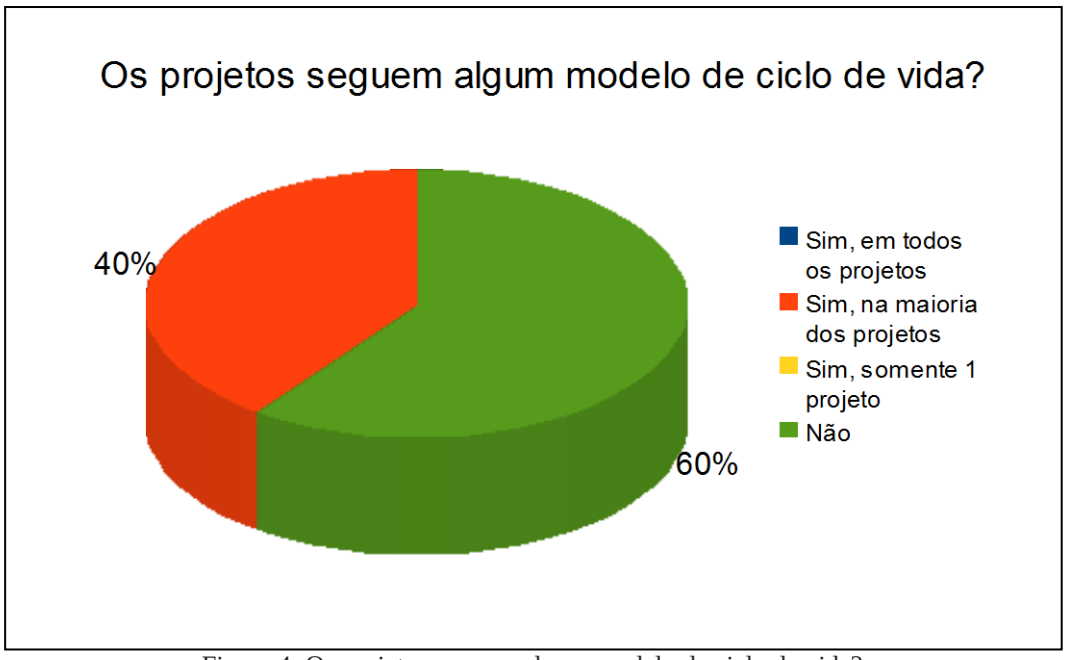

Figura 4: Os projetos seguem algum modelo de ciclo de vida?

Custos e esforços das tarefas

Outro fator pouco explorado pelas empresas é a estimativa de custos e esforços das tarefas que compõe os projetos. 60% das empresas nunca utilizaram essa estimativa como fator agregador em seus projetos, já 40 % delas utilizam ou já utilizaram desta em seus projetos.

Orçamento e cronograma.

Somente 17% das empresas pesquisadas, como mostra o gráfico da figura 6, estabelecem e mantêm cronogramas e orçamentos em todos os seus projetos de desenvolvimento de software. Em 33% delas, a definição de marcos e pontos de controle para o cronograma e orçamento é utilizada na maioria dos projetos, porém, para as 50% restantes, essa definição é não praticada.

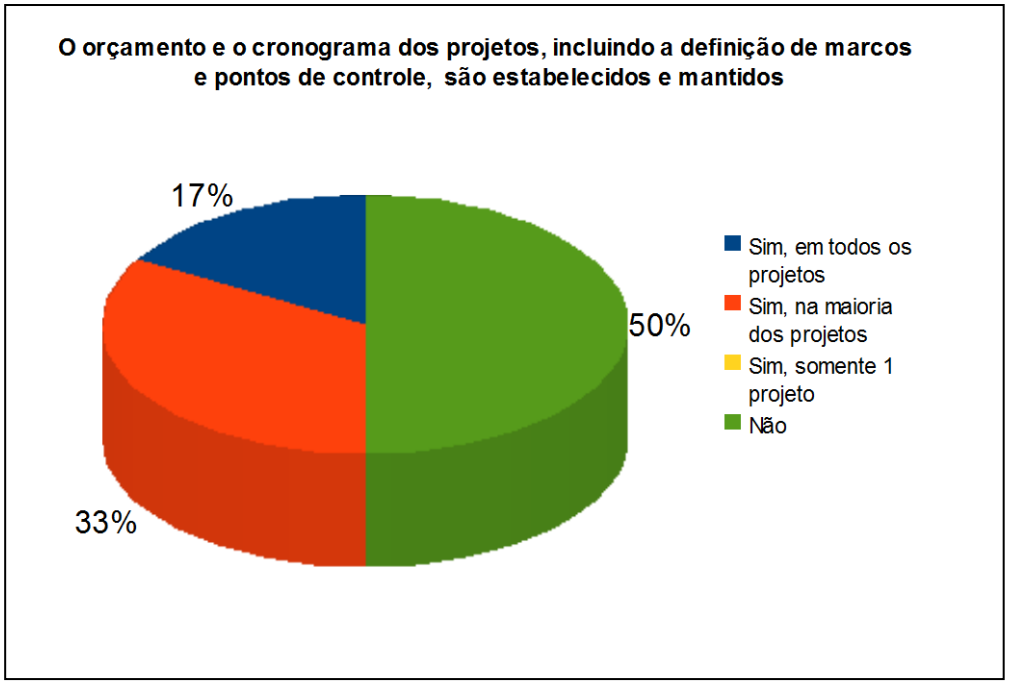

Figura 5: O orçamento e o cronograma dos projetos, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos

Recursos Humanos

No que diz respeito ao planejamento e manutenção dos recursos humanos, 80% das empresas não empregam uma política de análise mais apurada para a obtenção de mão-de-obra especializada para seus projetos, em contra partida, a maioria das mesmas mantêm um ambiente de trabalho planejado e com recursos necessários para a implementação dos projetos. As figuras 6 e 7 mostram os gráficos desses dois fatores de recursos humanos.

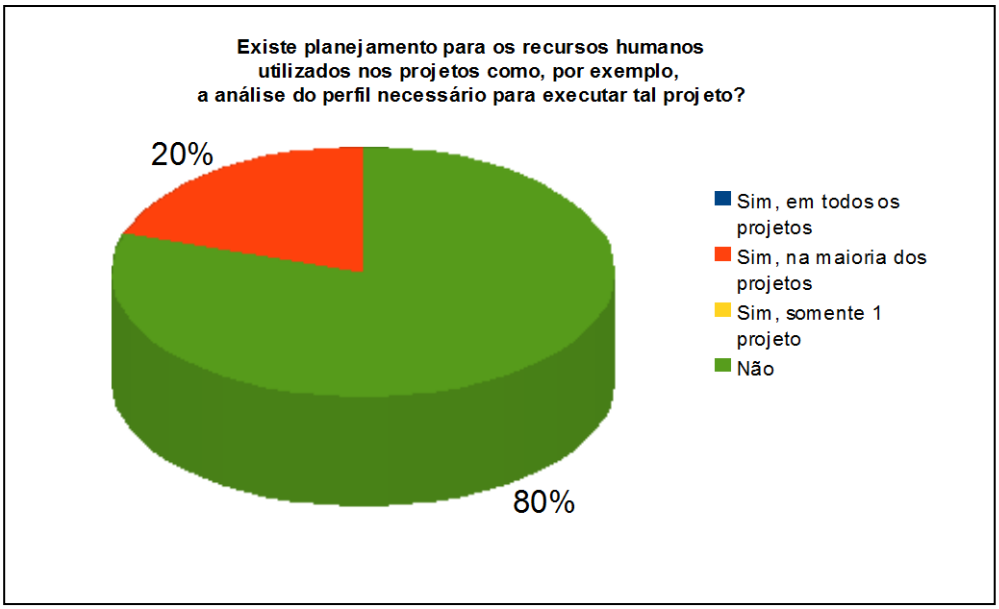

Figura 6: Existe planejamento para os recursos humanos utilizados nos projetos como, por exemplo, a análise do perfil necessário para executar tal projeto?

Figura 7: O ambiente de trabalho e os recuros existentes nele são planejados?

Segurança de dados

Como é visto no gráfico da figura 8, para 60% das empresas desenvolvedoras de software pesquisadas a questão da segurança de dados é de extrema importância por isso elas aplicam esse mecanismo em todos seus projetos. Em 20% delas, a segurança de dados é aplicada na maioria dos projetos e nos últimos 20% essa segurança não

existe.

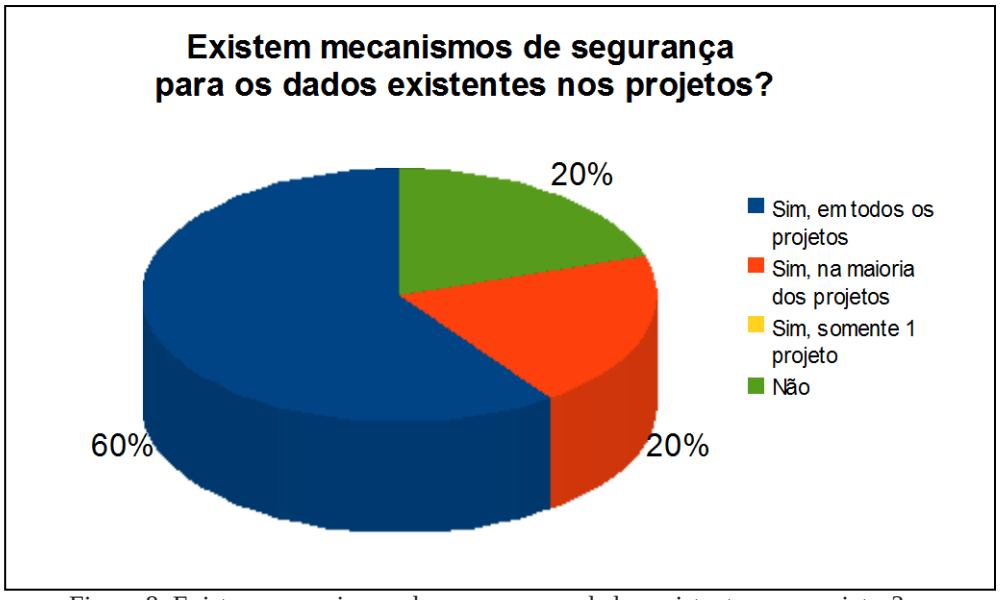

Figura 8: Existem mecanismos de segurança os dados existentes nos projetos?

Partes envolvidas

O envolvimento constante das partes interessadas no projeto e o planejamento do mesmo são feitos na maioria dos projetos em 60% das empresas pesquisadas. 20% das empresas tem atuação das partes na maioria dos projetos, e os 20% restantes não possui envolvimento necessário das partes.

Revisões e Correções

Com relação as revisões feitas ao fim de cada marco do projeto, os dados ficaram bem divididos: 40% das empresas fazem as revisões em todos seus projetos, 40% não fazem e nunca fizeram revisões e 20% aplicam as mesmas na maioria de seus trabalhos. Sobre o planejamento das ações de correção, 60% das empresas não são adeptas dessa prática, 20% fazem em todos os seus projetos e o restante planejam as correções para a maioria de seus trabalhos, conforme pode ser visto nos gráficos a seguir.

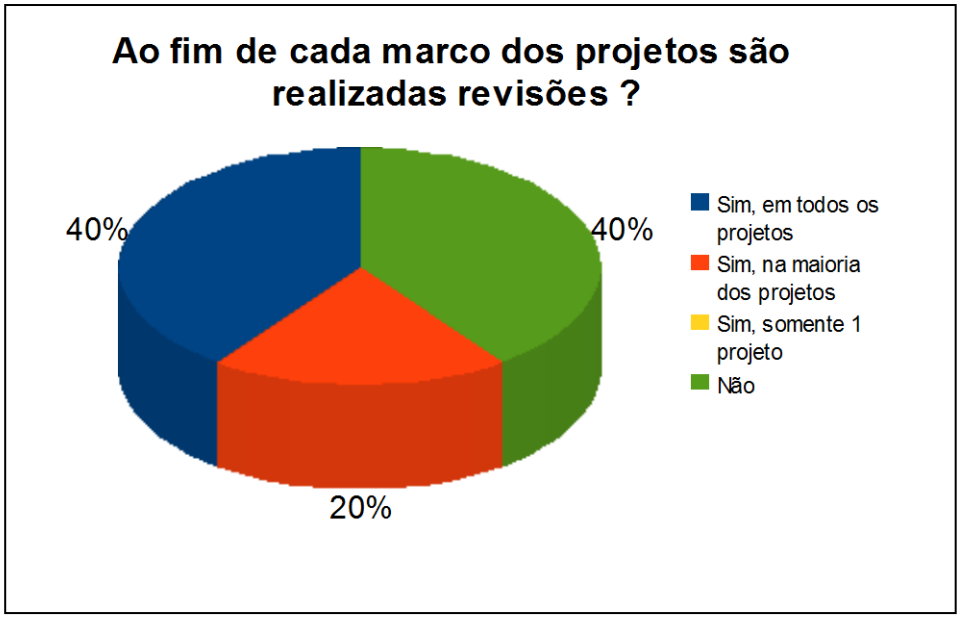

Figura 9: Ao fim de cada marco dos projetos são realizadas revisões?

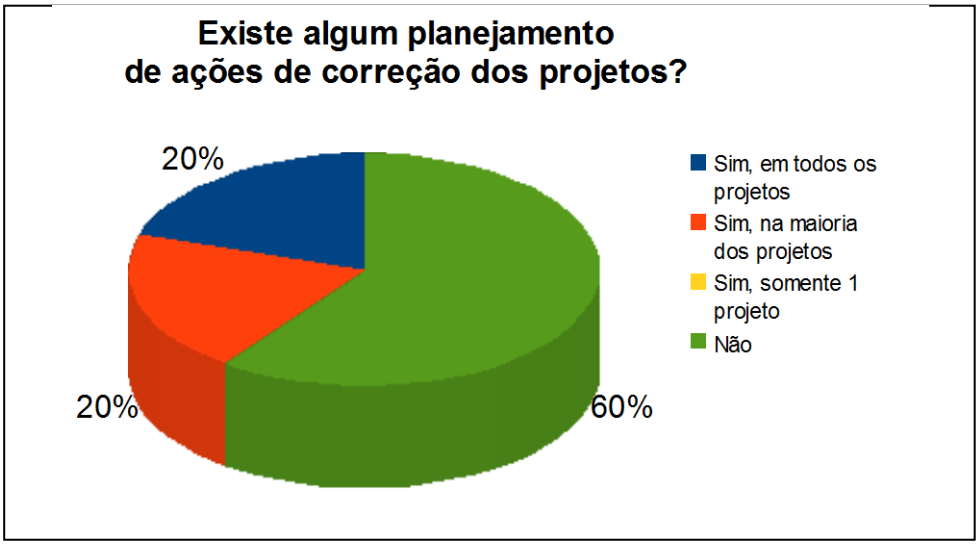

Figura 10: Existe algum planejamento de ações de correção dos projetos?

4.3.1 Análise dos resultados

Observando os resultados obtidos após a aplicação do questionário nas empresas, levando em consideração o fator Gerência de Projetos e baseando-se nas idéias abordadas pelos guias de implementação e avaliação MPS-BR, pôde-se chegar as seguintes conclusões:

- A Estrutura Analítica do Projeto (EAP), que é a representação do projeto como

um todo, ou seja, o escopo do mesmo, é um artefato ainda pouco explorado pela empresas desenvolvedoras de software de Vitória da Conquista. Esse fato é explicado pelos baixos índices de boa definição de escopo dos projetos e na quase nulidade de utilização de métodos de dimensionamento das tarefas e produtos dos processos de software.

- A falta de uma boa definição de escopo de projeto é prejudicial pois, como explica o Guia de Implementação Nível G MPS-BR (MPS-BR, 2009), é a partir dele que o produto, no caso o software, começa a ser fundamentado e é ele que especifica a definição do projeto, a motivação, os produtos a serem gerados, os produtos a serem entregues, entre outros. Um projeto iniciado sem uma boa estrutura base tende sofrer com problemas tais como custo, tempo, prevenção e tratamento de erros e manutenção das finalidades estabelecidas no decorrer do mesmo.
- Outros dois pontos importantes no planejamento dos projetos são a utilização de estimativas de custos e esforços para a execução das tarefas e o estabelecimento, do início ao fim do projeto, do cronograma e orçamento. Isso faz com que o tratamento dos produtos gerados, dos possíveis riscos, das mudanças já esperadas, do ciclo de vida escolhido, da dependência entre as fases, dos custos das fases e da competência da equipe seja feito de forma precoce, eficaz e ágil, na medida do possível.
- As empresas de Vitória da Conquista pesquisadas não empregam de forma unânime esses planejamentos, o que se explica, possivelmente, pelo fato de a maioria delas serem de tamanho micro o que faz com que despesas que as mesmas poderiam dedicar a tal prática sejam utilizadas para outros fins.
- O pouco uso de modelos de ciclo de vida predeterminados, também influencia no planejamento e desenvolvimento do software, já que sem eles as tarefas de cada fase do projeto vistas na definição do escopo dificilmente são organizadas para gerarem os produtos desejados e manterem uma ligação com as outras fases.
- Sobre os recursos humanos, o importante a ser avaliado é como a empresa está planejando a seleção das pessoas, a depender de suas aptidões, para exercer determinada função no projeto e o como a empresa está planejando o ambiente de trabalho e os recursos existentes nele necessário a implementação do projeto.

Em Vitória da Conquista, as empresas questionadas mantêm uma constância positiva no planejamento do ambiente e dos recursos, porém o oposto é visto na análise da mão-de-obra para alocação nas funções. Isso pode acarretar em perda de desempenho e uso excessivo e não apropriado de recurso financeiro e também humano.

- Um importante ponto tratado positivamente pelas empresas pesquisadas é o planejamento e implantação de instrumentos e métodos de segurança de dados. Tal fato é crucial, pois protege as informações em todos os seus instantes (coleta, tratamento, armazenamento e distribuição) tanto de ataques internos, como o acesso inapropriado dentro da própria empresa, como em acessos indevidos externos.
- Por fim, a correção e prevenção dos desvios no projeto estão em franco processo de crescimento nas empresas pesquisadas, o que não é de total desvio com relação a empresas de outros pólos, já que essas duas práticas também se encontrar em processo de desenvolvimento e aprimoramento.

4.4 Levantamento dos resultados sobre Gerenciamento de Requisitos

A Gerência de Requisitos, de acordo com o Guia de Implementação Nível G MPS-BR (SOFTEX, 2010), tem como objetivo a identificação dos requisitos do produto e dos componentes do produto do projeto, ao mesmo tempo em que tem como finalidade manter um acordo entre o cliente e os desenvolvedores sobre os requisitos Além disso, essa gerência cuida do tratamento e do controle das mudanças nos requisitos que poderão ocorrer no decorrer do projeto.

A seguir é mostrado o levantamento das respostas do questionário sobre a Gerência de Requisitos:

Entendimento, avaliação e aceitação dos requisitos

Em 60% das empresas de desenvolvimento de software de Vitória da Conquista pequisada, o entendimento, a avaliação e a aceitação dos requisitos do projeto por todos os interessados são práticas constantes na maioria dos projetos. Em 20% delas, essas práticas são feitas em todos os projetos executados e, somente nas 20% restantes é que

as partes interessadas não teem total controle sobre os requisitos.

Plano de trabalho e requisitos

A ligação entre o plano geral de trabalho e os requisitos do projeto é, como mostrado na figura 11, em 60% das empresas pesquisadas, revisada na maioria dos projetos para que inconsistências não sejam criadas.

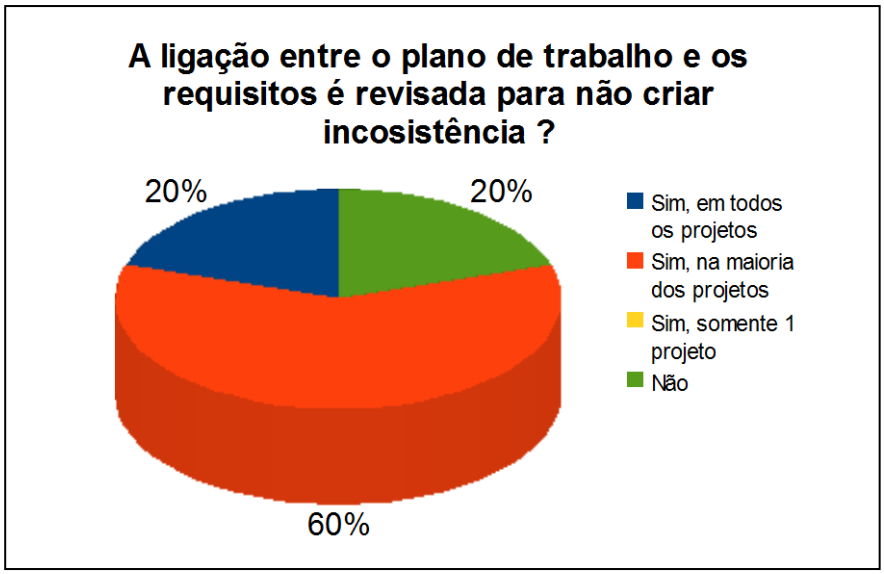

Figura 11: A ligação entre o plano de trabalho e os requisitos é revisada para não criar inconsistência?

Mudança nos requisitos

Com relação as mudanças dos requisitos que tendem a ocorrer no tempo de vida do projeto, 60% das empresas questionadas não gerenciam esse fato, as 40% restantes fazem o gerenciamento em todos os seus projetos.

4.4.1 Análise dos resultados

Observando os resultados obtidos após a aplicação do questionário nas empresas, levando em consideração o fator Gerência de Requisitos e baseando-se nas idéias abordadas pelos guias de implementação e avaliação MPS-BR, pôde-se chegar as seguintes conclusões:

- A avaliação dos requisitos, englobando seu entendimento e aceitação, é um estágio bem praticado pelas empresas, o que garante que os requisitos, pelo menos parcialmente, sejam bem levantados e passados pelo fornecedores e entendidos e transformados pelos desenvolvedores. Porém, numa análise mais profunda, as empresas pesquisadas não documentam fortemente esse entendimento, o que prejudica a consistência, durabilidade e perpetuação dos requisitos, influenciando consideravelmente no produto resultante. Outro fator depreciado pela má documentação dos requisitos é a vinculação do plano de trabalho aos mesmos, pois a falta de documentos detalhados que especifiquem os requisitos prejudica as revisões necessárias para manter a consistência entre o plano de trabalho e os requisitos.
- Tão importante quanto o levantamento e entendimento dos requisitos é o gerenciamento de suas possíveis mudanças. A grande maioria das empresas de software pesquisadas não leva isso em consideração o que, geralmente, produz inconsistências entre o que foi inicialmente proposto como resultado final desejado e o resultado final obtido, ocasionando em uma necessidade de retrabalho, aumentando assim os custos, esforços e tempo de conclusão e, conseqüentemente, aumentando o grau de insatisfação dos clientes.

5. CONCLUSÃO

A utilização de método que auxiliem no gerenciamento dos processos de desenvolvimento de software é algo imprescindível para se obter sucesso no atual concorrido mercado de informática. Em contra partida, o não uso desses métodos pode criar nas empresas um ambiente pouco produtivo, com um grau de eficiência e eficácia que não correspondam as necessidades impostas pelos clientes, tornando a empresa uma referência pouco visada e com pouca credibilidade.

A pesquisa do nível de maturidade das empresas de software de Vitória da Conquista, segundo as regras do MPS-BR, mostrou o quão a utilização de método, ainda está aquém do necessário para que as mesmas possam por em prática todo os seus possíveis potenciais.

O tamanho das empresas ainda é um empecilho para que alguns tipos de tecnologias sejam implantadas, porém, a partir dessa pesquisa, pôde-se observar que não é o tamanho o principal problema, mas sim a falta de conhecimento sobre alternativas mais baratas e viáveis para se implantar.

A aplicação e estudo dos questionários ainda proporcionaram o conhecimento do pouco investimento que as empresas dão a alguns aspectos importantes para se conseguir um bom projeto de software, como a documentação mais detalhada dos requisitos e a melhor pesquisa da mão-de-obra necessária a determinado projeto.

Porém, o estudo também mostrou que as empresas valorizam pontos fortes para o sustento das bases de criação de software como, por exemplo, a análise dos requisitos perante os envolvidos e a segurança de dados.

Portanto, unindo todos os fatores citados anteriormente, conclui-se que o nível de maturidade das empresas de software de Vitória da Conquista está em um patamar inicial, que necessita ser revisado e incrementado com o auxilio de tecnologias, método e de, principalmente, informação e conhecimento. O patamar de maturidade em que se encontram, apesar de revelar um nicho de empresas promissor e forte, ainda não esta condizente com a realidade de um pólo de informática como a cidade de Vitória da Conquista.

O objetivo inicial deste trabalho, que era mostrar como está o nível de maturidade nas empresas de desenvolvedoras de software em Vitória da Conquista ação empresarial e apontar possíveis melhorias ao mesmo, foi alcançado. Os dados coletados

apresentaram fielmente a situação geral do mercado de software conquistense e ajudaram consideravelmente na formulação de alternativas de alavancamento deste.

Este trabalho pode contribuir para uma melhor visão da situação das empresas de software de Vitória da Conquista. O processo de análise abordado por ele pode auxiliar em um estudo mais profundo de mudanças a serem feitas no mercado de software para que o mesmo evolua e se torne mais concorrido. Outra melhoria seria no maior entendimento perante os profissionais que trabalham nas áreas de metodologias, como o MPS-BR, que auxiliem no desenvolvimento de sistemas de forma mais apropriada e condizente com as necessidades impostas pelo mercado atual. E, por fim, uma contribuição importante seria a manutenção das relações entre as empresas privadas e as instituições de ensino, como a UESB, no intuito de ser formada um relação de assistência mútua importante para o crescimento como um todo do pólo tecnológico de Vitória da Conquista.

Como trabalhos futuros, pode ser feita uma análise mais aprofundada do nível de maturidade das empresas, servindo como um projeto base para implantação do MPS-BR ou outro equivalente nas empresas que desejarem.

REFERÊNCIAS

ABES, Associação Brasileiras das Empresas de Software. **Mercado Brasileiro de Software**: *Panorama e Tendências 2009.* Disponível em: < http://www.abes.org.br/arquivos/MercadoBR-2009-ResumoExec.pdf >. Acessado em: 19/05/2010.

AGILE ALLIANCE**. Princípios por trás do Manifesto Ágil.** Disponível em: < http://www.agilemanifesto.org/iso/ptbr/principles.html **>.** Acessado em: 09/01/2011

ÂNGELO, Fernanda K. **CMMI vira requisito para desenvolvedores.** Disponível em: <http://www.tso-cons.com/noticias.htm> Acesso em 19/09/2010

AROUCK, O. **Avaliação de sistemas de informação**: revisão da literatura. Transinformação, v. 13, n. 1, 2001.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. **Normas ABNT sobre documentação**. Rio de Janeiro, 2000.

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR** – **Guia Geral:2009**, maio 2009. Disponível em <www.softex.br>. Acessado em: 01/09/2010

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR - Guia de Avaliação:2009**, maio 2009. Disponível em: <www.softex.br>. Acessado em: 01/09/2010

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR – Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS:2009**, maio 2009. Disponível em: <www.softex.br>. Acessado em: 01/09/2010

AUSTIN, Jim; Portny, Stanley E. **Project Management for Scientists.** Science,

2002. Disponível em: <http://sciencecareers.sciencemag.org/career_development/previous_issues/articles/1750/project_management_for_scientists/> Acessado em: 20/10/2010.

BARBOSA, Francisco Vidal, et. al. **O Brasil Como Importante Player no Mercado Mundial de Software**: *Uma Avaliação Baseada no Modelo de Exportação de Heeks e Nicholson*. 3º Colóquio IFBAE, Grenoble, FR. anais.....2005.

BICHARA, P. A. D., et. al. **HiperionCad**: *Desenvolvimento Orientado a Modelos Aplicado ao Projeto de Redes de Fibra Ótica.* UFBA, Salvador, BA, 2007.

BOEHM, Barry W. **Anchoring the Software Process**. *IEEE Software. 1*996.

BOHEM, B. **The Spiral Model as a Tool for Evolutionary Software Acquisition**. Crosstalk, 2001.

BOHEM, B. ***Balancing Agility and Discipline:*** *A Guide for the Perplexed* .Boston, MA: Addison-Wesley, 2004.

BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML - Guia do Usuário**. 2 ed., Editora Campus. 2005.

BRADAC, M.; PERRY, D.; VOTTA, L. **Prototyping a Process Monitoring Experiment**. IEEE Trans. Software Engineering, v. 20, n. 10, 1994.

CAPABILITY Maturity Model Integration (CMMI). **CMMI for software engineering version 1.1(CMMI-SW, V1.1).** Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Staged Representation**,** Improving processes for better products (CMU/SEI-2002-TR-029/ESC-TR-2002-029), Aug., 2002a.

CAMPELO, G.. **A utilização de Métricas na Gerência de Projetos de Software**: *uma abordagem focada no CMM nível 2*. Dissertação de Mestrado. Universidade Federal de Pernambuco – Centro de Informática. Recife, Pernambuco. Brasil. 2002.

CARVALHO, Maria do Socorro. **Mapeando a ISO 9001 para o CMMI**. Dissertação de conclusão de graduação em Ciência da Computação, Faculdade Lourenço Filho, Fortaleza, Ceará. 2007

CARVALHO,R. **Mercado de Software cresce 22,3% em 2007**. Disponível em: <http://www.metaanalise.com.br/inteligenciademercado/inteligencia/pesquisas/mercado-de-software-cresceu-22-3-em-2007.html>. Acessado em: 23/08/2010.

COHEN, D., Lindvall, M., & Costa, P. **An introduction to agile methods**. In *Advances in Computers* New York: Elsevier Science. 2004

CORRÊA, C. K. F.; MURTA, L. G. P.; WERNER, C. M. L. **Uma Abordagem para o Controle de Evolução de Software no Desenvolvimento Orientado a Modelos.** UFRJ, Rio de Janeiro, RJ, 2007.

COUTO, Ana Brasil. **CMMI**: *integração dos modelos de capacitação e maturidade de sistemas.* Rio de Janeiro: Ciência Moderna, 2007.

CS3 Consulting Service, 2002. Disponível em: <http://cs3inc.com/DSDM.html>. Acessado em: 01/09/2010

CURTIS, B. et al. **People Capability Maturity Model (CMU/SEI-95-MM-002)**. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon Universtity, 1995.

D´ÁVILA, Márcio. **PMBOK e Gerenciamento de Projetos – atualizado**. 2010. Disponível em: <http://blog.mhavila.com.br/2010/03/18/pmbok-e-gerenciamento-de-projetos-atualizado/>. Acessado em: 12/10/2010.

DOWSON, M.; NEJMEH, B.; RIDDLE, W. **Fundamental software Process Concepts**. In: EUROPEAN WORKSHOP ON *SOFTWARE* PROCESS MODELLING, 1., 1991, Milan, Italy. Proceedings... [S.l.]: AICA Press, May 1991.

ENANI, L. M.; HUZITA, E. H. M.; TAIT, T. F. C. **A project management model to a distributed software engineering environment**. In Proceedings of ICEIS 2006 – International Conference on Information Systems, Cyprus, 2006.

GIRAFFA, Lúcia; MARCZAK, Sabrina; PRIKLADNICKI, Rafael. **PDS-E**: *Em direção a um processo para desenvolvimento de Software Educacional*. XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo, RS, 2005.

HUMPHREY, Watts S. "TSP – Coaching Development Teams". Addison-Wesley. 2006.

ISO 9000. **ISO 9001:2008 -** *Quality management systems – Requirement.* Disponível em: **http://www.iso.org/iso/iso_catalogue/management_and_leadership_standards/quality_management/iso_9001_2008.htm>** *Acessado em: 01/10/2010*

ISO/IEC - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBRISO/IEC9126-1 **Engenharia de software - Qualidade de produto - Parte 1: Modelo de qualidade**. 2003.

JUNIOR, Edson Oliveira Lessa; LESSA, Rafael Orivaldo. **Modelos de Processo de Engenharia de Software**. UNISUL, Palhoça, SC, 2008.

KERZNER, Harold. **Gestão de Projetos:** *As melhores práticas*. São Paulo: Bookman, 2001.

KRUCHTEN, P. **The Rational Unified Process**. An Introduction. 2nd Edção, Addison-Wesley, 2000.

LESSA, Rafael Orivaldo; JUNIOR, Edson Orivaldo Lessa. **Modelos de**

Processos de Engenharia de Software. UNISUL, Palhoça, SC. 2008.

MACBREEN,P. [sem título] .Boston, MA:Addison-Wesley, 2003.

MIT, Massachussets International Techonology. **Slicing the Knowledge-based Economy in Brazil, China and India:** *a tale of 3 software industries.* Massachussets, 2003.

NETO, O. N. S. **Análise Comparativa das Metodologias de Desenvolvimento de Softwares Tradicionais e Ágeis**. Belém, 2004. Disponível em: <http://www.cci.unama.br/margalho/portaltcc/tcc2004/oscar.PDF>. Acessado em: 08/10/2010.

OLIVEIRA, S. R. B; ROCHA, T. A.; VASCONCELOS, A. M. L. **Adequação de Processos para Fábricas de Software.** Anais do Simpósio Internacional de Melhoria de Processos de Software – SIMPROS 2004, São Paulo, 2004.

OLIVEIRA, Karine Coelho de Amorim. **Análise da Aplicabilidade do CMMI em Fábricas de Software orientadas a Produto no contexto de Gerenciamento de Requisitos**. Monografia da disciplina Tópicos Avançados em Engenharia de Software II - Universidade Federal de Pernambuco Centro de Ciências Exatas e da Natureza - Departamento de Informática - Mestrado em Ciências da Computação – 2007.

OMG. **Model Driven Architecture (MDA)**. Object Management Group (OMG), Document ormsc/01-07-03, July 2003. Disponível em: <http://www.omg.org/docs/ormsc/01-07-03.pdf>. Acessado em: 01/10/2010.

PINHEIRO, Andréa. **O Brasíl é o sétimo maior mercado de software(2003)**. Disponível em: <http://webinsider.uol.com.br/2003/04/27/brasil-e-o-setimo-maior-mercado-de-software/> . Acesso em 19/05/2010.

PIRES, Hindeburgo Francisco. **Reestruturação e Alta-Tecnologia no Brasil:** As *indústrias de Informática de São Paulo*. São Paulo: USP, 1995. Disponível em: <http://www.cibergeo.org/tese>. Acessado em: 15/05/2010.

PREFEITURA DE VITÓRIA DA CONQUISTA. [sem título]. Vitória da Conquista, BA, 2009.

PMI BRASIL. **Sobre o PMI**. 2010. Disponível em: <http://www.pmi.org.br/>. Acessado em: 21/09/2010

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. Tradução de Rosângela Delloso Penteado. São Paulo: McGraw-Hill Interamericana do Brasil, 2006.

PROJECT MANAGEMENT INSTITUTE**. Government Ext. To A Guide To The Project Management Body Of Knowledge (PMBOK® Guide) — 2000 Ed.** PMI**,** 2003

ROYCE, W.W. **Managing the development of large software systems:** *concepts and techniques*. Proc. IEEE Westcon, Los Angeles,CA. 1970.

SCHWARE, R. **Software industry entry strategies for developing countries:** *awalking on two legs proposition*. World Development, v.20, n.2, p.143-164,1992.

SCHWARTZ, Jonathan I. Construction of software. In: **Practical Strategies for Developing Large Systems**. Menlo Park: Ed. Addison-Wesley, 1975.

SEI – Carnegie Mellon Software Engineering Institute. **Capability Maturity Model Integration**, v. 1.1. 2002. Disponível em: <http://www.sei.cmu.edu/publications/documents/02.resports/02tr012.html>. Acesso em 05/11/2010.

SOLEY, R. **Model Driven Architecture**, OMG document omg/00-05-05, Object Management Group, 2000.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Pearson Addison-Wesley, 2007.

SOTILLE, Mauro. **Gerenciamento de Projetos na Engenharia de Software**. São Paulo: Pmtech. 2006.

STANDISH GROUP . ***The CHAOS Report (1994)***. Report of the Standish Group. Disponível em : <http://www.standishgroup.com/sample_research/chaos_1994_1.php> Acessado em: 15/08/2010.

STANLEIGH, Michael. **Combining the ISSO 10006 and PMBOK to ensure successful projects**. Traduzido por Profa. Dra. Kátia P. Thomaz. 2006.

SWEBOK**. Guide to the Software Engineering Body of Knowledge.** 2004 Version. A project of the IEEE Computer Society Professional Practices Committee. Disponível em: <http://www.swebok.org>. Acessado em: 01/10/2010.

VARGAS, R. V. **Gerenciamento de Projetos:** *estabelecendo diferenciais competitivos*. Rio de Janeiro: Brasport, 2005.

VARGAS, Ricardo Viana.**Gerenciamento de Projetos com o MS Project 98, Estratégia, Planejamento e Controle**. Rio de Janeiro: Brasport, 1998.

VELOSO, Francisco, et. al. **Slicing the Knowledge-Based Economy in Brazil, China and India:** *a tale of 3 software industries*. MIT, 2003. Disponível em: <http://www.softex.br/media/MIT_final_ing.pdf>. Acesso em 29/09/2010.

TAKANO, Edna Tomie. **UM MODELO DE GERENCIAMENTO DE PROCESSO DE *SOFTWARE* PARA O AMBIENTE DiSEN.** Dissertação apresentada para Pós-Graduação em Ciência da Computação, Universidade Estadual de Maringá, Paraná. 2006.

ANEXO 1

QUESTIONÁRIO AVALIATIVO DO NÍVEL DE MATURIDADE

Este formulário visa avaliar o nível de maturidade de empresas de software da cidade de Vitória da Conquista, segunda as normas do MPS-BR.
O questionário consta de 15 questões objetivas.

PARTE I – GERÊNCIA DE PROJETO

1. Como você avalia o processo de desenvolvimento de software?

   [ ] Gerenciado
   [ ] Gerenciado parcialmente
   [ ] Não gerenciado

2. O escopo do projeto é bem definido?

   [ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos
   [ ] Sim, somente 1 projeto [ ] Não

3. As tarefas e produtos necessários para implementação dos projetos são dimensionados, utilizando ferramentas como a Análise de Ponto Por Função?

   [ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos
   [ ] Sim, somente 1 projeto [ ] Não

4. Os projetos seguem algum modelo de ciclo de vida ?

   [ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos
   [ ] Sim, somente 1 projeto [ ] Não

5. Os custos e esforços das tarefas que compõe os projetos são estimados com base em dados históricos ou referências técnicas?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos

[ ] Sim, somente 1 projeto [ ] Não

6. O orçamento e o cronograma dos projetos, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos

[ ] Sim, somente 1 projeto [ ] Não

7. Existe planejamento para os recursos humanos utilizados nos projetos como, por exemplo, a análise do perfil necessário para executar tal projeto ?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos

[ ] Sim, somente 1 projeto [ ] Não

8. O ambiente de trabalho e os recursos existentes nele são planejados ?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos

[ ] Sim, somente 1 projeto [ ] Não

9. Existem mecanismos de segurança para os dados existentes nos projetos?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos

[ ] Sim, somente 1 projeto [ ] Não

10. As pessoas interessadas nos projetos tem envolvimento constante. Se sim, este envolvimento é gerenciado?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos

[ ] Sim, somente 1 projeto [ ] Não

11. Ao fim de cada marco dos projetos são realizadas revisões ?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos

[ ] Sim, somente 1 projeto [ ] Não

12. Existe algum planejamento de ações de correção dos projetos ?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos
[ ] Sim, somente 1 projeto [ ] Não

PARTE II – GERÊNCIA DE REQUISITOS

13. Os requisitos dos projetos são entendidos, avaliados e aceitos por todos os interessados ?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos
[ ] Sim, somente 1 projeto [ ] Não

14. A ligação entre o plano de trabalho e os requisitos é revisada para não criar inconsistência?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos
[ ] Sim, somente 1 projeto [ ] Não

15. Quando há mudanças nos requisitos, estas são gerenciadas ?

[ ] Sim, em todos os projetos [ ] Sim, na maioria dos projetos
[ ] Sim, somente 1 projeto [ ] Não