**1 CICLO DE VIDA DO SOFTWARE**

 O Ciclo de Vida do Sistema apresenta os detalhes de como fazer para se construir um software, cada fase dessa construção e as tarefas que cada pessoa da equipe de Engenharia de Software fará, sendo que cada sistema tem sua particularidade.

 Segundo Pressmam (2006, p.38),

Todos os modelos de processo de software podem acomodar atividades genéricas, mais cada um aplica uma ênfase diferente às essas atividades e define um fluxo de trabalho que invoca cada atividade de maneira diferente.

 Sendo assim o modelo de processo é definido de acordo com o produto final definido, fazendo com que as atividades intermediárias sejam feitas de acordo com a natureza do sistema e de sua aplicação. Exemplos desse ciclo de vida tem-se o modelo de vida clássico, o Cascata; o Espiral que mostra uma série de iterações correspondendo a uma volta na espiral definindo daí os resultados obtidos, que fornecem dados para a nova iteração, esse modelo é usado para o desenvolvimento de softwares a curto prazo e o Modelo de Prototipagem evolutiva que trata da construção de várias versões, os protótipos, esses ao longo do processo aperem com um versão melhor do eu a anterior até atingir o produto final desejado.

**1.1 MODELO CASCATA**

O Modelo Cascata é o escolhido para o software que será desenvolvido nesse projeto porque é uma abordagem seqüencial e sistemática que começa a partir da comunicação com a especificação de requisitos, seguindo para a fase de planejamento, depois modelagem, construção e por fim a fase de implantação.

 Essas cinco fases podem ser feitas sem a necessidade de interação entre elas e são caracterizadas da seguinte forma (PRESSMAN, 1995):

1. **Comunicação:** essa primeira fase é um levantamento amplo de requisitos do ambiente em que o software irá fazer parte como os elementos que ele irá fazer a interface, o hardware, as pessoas e o banco de dados. Essa coleta de requisitos é em nível de sistema com pequena quantidade de projeto e análise de alto nível;
2. **Planejamento:** essa fase começa o seu trabalho com o processo de coleta de requisitos intensificada e concentrada no software, o engenheiro de software deve compreender o domínio da informação que o cliente deseja, por isso os requisitos do sistema são documentados e revisados pelo cliente;
3. **Modelagem:** nessa fase o projeto de software o processo de software é múltiplos passos, concentrados em quatro atributos distintos: a arquitetura do software, detalhes do procedimento, estrutura dos dados e a caracterização da interface;
4. **Construção ou Codificação:** a fase em o projeto será traduzido para linguagem de máquina, fazendo também os testes quando for gerado o código, para que cada instrução dada não contenha erros garantindo resultados conforme fora exigido’
5. **Implantação:** é última fase onde será entregue o produto final é onde também será feita as mudanças para se acomodar ao ambiente externo ou porque o cliente exigirá acréscimos funcionais ou de desempenho.



Figura 02 - Modelo Cascata

Fonte: PRESSMAN, 2006.

O Modelo Cascata tem alguns problemas que devem ser explicados, pois ao se definir como um projeto com fluxo seqüencial nem todos seguem a definição que o modelo propõe, pois uma fase só deve começar com o término da outra, fazendo com que membros da equipe de projeto precisem esperar que os outros membros terminem suas tarefas na fase anterior, resultando em um tempo gasto em espera muitas vezes maior do que o tempo produtivo (PRESSMAN, 2006).

**1.2 PROCESSO DE DESENVOLVIMENTO DO SOFTWARE**

O Processo de desenvolvimento do software é a contextualização do projeto e da análise desse software com a ajuda de uma metodologia de desenvolvimento, pois define como sendo um conjunto de atividades que resultam em associações para a produção de software. Visto que nenhum processo pode ser conspirado ideal por serem convergentes mais com atividades fundamentais compartilhadas como: especificação de software; projeto e implementação de software; validação de software e evolução de software (SOMMERVILLE, 2003).

Os processos de desenvolvimento do software mais utilizados são:

1. **RUP (RATIONAL UNIFIED PROCESS):** é um processo formado por ciclos, fases, iterações e marcos que integram-se através da linguagem de modelagem unificada (UML) e também por processos comuns aos usuários finais, programadores, analistas e projetistas de maneira que sela disciplinada, fazendo com que ao final haja um desenvolvimento de software de alta qualidade (RATIONAL, 1998).

O RUP segundo SILVA E VIDEIRA (2001) é caracterizado por ter boas práticas de desenvolvimento, tendo como principal características:

* É uma metodologia iterativa e incremental;
* Usa casos de uso (use cases);
* É adepto a modelagem visual;
* O seu desenvolvimento de software é mostrado em arquiteturas de software e em arquitetura de componentes,
* Tem UML como linguagem de modelagem.
1. **XP (EXTREME PROGRAMMING):** é uma metodologia onde se é desenvolvida um software ágil que atenda os clientes e suas necessidades com qualidade. Essa metodologia atende projetos com requisitos que mudam com frequência e que utilizem desenvolvimento orientado a objetos e incremental com equipes de até 12 desenvolvedores. O XP busca sempre, a cada dia de trabalho, o máximo de sua equipe em relação ao desejo do cliente, para que em pouco tempo o cliente tenha um produto que possa ser utilizado para aprender e reavaliar o mesmo e verificar se estar de acordo com o que foi o desejado (BECK, 2004).

O XP é organizado em valores e práticas que asseguram que o investimento do cliente tenha um retorno alto. Os valores definem como a equipe deve agir e também as principais prioridades dessa metodologia que são os seguintes: Feedback onde o cliente verifica o desenvolvimento do produto estabelecendo prioridades e dizendo o que é importante para se ter um produto ao seu contento; Comunicação para que aconteça um feedback entre o cliente e o desenvolvedor com sucesso, o XP propões que haja uma comunicação entre essas partes direta e eficaz, com o contato, se possível, pessoalmente para evitar especulações e mal entendidos e que as dúvidas sejam tiradas imediatamente; Simplicidade esse valor determina que o software seja desenvolvido e implementado da forma mais simples possível para que o cliente tenha seu pedido atendido e por fim o valor da Coragem que por ser um processo de desenvolvimento novo que contrariam o modelo tradicional, os desenvolvedores deverão ter coragem de desenvolver um software de forma incremental, fazer um sistema simples, permitir que o cliente participe e defina as prioridades entre outras (ASTELS E NOVAK,2002).

1. **PRAXIS** **(PROCESSO PARA APLICATIVOS EXTENSÍVEIS INTERATIVOS) :** é um processo para fins didáticos. Que segundo PÁDUA (2005), o PRAXIS busca suportar projetos realizados individualmente ou por pequenas equipes e que tenham a duração de seis meses a um ano, e que seja utilizados em projetos de conclusão de curso de graduação ou similares. O PRAXIS e composto pelos os requisitos, análise, desenho, testes e implementação, gestão de requisitos, gestão de projetos, garantia de qualidade e gestão de configurações, é baseada na orientação a a objetos utilizando a UML.
2. **ICONIX:** metodologia, prática, pura e simples e ainda muito poderosa, não tão burocrática, pois não gera tanta documentação e nem tão simples pois não deixa a desejar na Análise de Design (MAIA, 2005). Essa metodologia será mais bem definida na próxima seção por a escolhida para esse projeto.

**REFERÊNCIAS**

CAMARGO, Álvaro Antônio Bueno De. KHOURI, Lourdes Halim El e GIAROLA, Paulo César. **O** **Uso de Sistemas Colaborativos na Gestão de Projetos:** Fatores Relevantes para o Sucesso. Trabalho de Conclusão de Curso. Fundação Instituto de Administração – FIA. 2005.

FERREIRA, Felipe.**Linguagem PHP vantagens e aplicações, 2005. Disponível em <**<http://www.artigonal.com/internet-artigos/linguagem-php-vantagens-e-aplicacoes-1788643.html>**>> Acesso em 12 de outubro de 2010.**

FILHO, Wilson de Pádua. **Engenharia de Software** – Fundamentos, Métodos e Padrões. Rio de janeiro: LTC, 2001.

LAUDON, K. C. & LAUDON, J. P. **Sistemas de Informação**. Rio de Janeiro: LTC, 1999.

LAUDON, K. C., LAUDON, J. P.**Sistemas de Informação Gerencias:** Administrando a empresa digital. 5. ed. São Paulo: Prentice-Hall, 2004

MAIA, José Anízio. **Construindo softwares com qualidade e rapidez usando ICONIX**, 2005. Disponível em <<<http://www.guj.com.br>>> Acesso em 25 de maio de 2010.