# INTRODUÇÃO

Com este trabalho podemos ter uma noção maior dos tipos de modelos de ciclo de vida , percebendo todo o desenvolvimento e preparação, juntamente com o nível de dificuldade e atenção que cada modelo merece, não esquecendo da agilidade que o mercado de hoje exige, podemos conhecer também um pouco dos chamados processos ágeis e a junção de todas os conceitos e metodologias para um resultado esperado dentro do processo unificado rational (RUP).

# Modelos de ciclo de vida

Os modelos de ciclo de vida são divididos em quatro tipos, são eles: cascata, espiral, prototipação e codifica-remenda.

## CASCATA

Este é o modelo considerado mais antigo, criado em aproximadamente 1970, tem como marco as fases seqüenciais e sistemáticas, ou seja, como um vídeo game, só pode-se passar para uma fase posterior tendo completado a fase anterior, caso contrário, “*game over”* , deve-se começar todas as fases novamente.

## Prototipação

É um processo que visa entender os objetivos e necessidades do usuário antes de desenvolver um projeto, sendo assim a prototipaçao divide-se em algumas partes:

### Obtenção de requisitos:

Esta etapa visa conseguir o maior número de informações possíveis sobre o que o cliente deseja e a sua necessidade, analisando também algumas possíveis definições adicionais que podem melhorar o projeto.

### Projeto Rápido:

É uma espécie de projeto onde somente é feito as entradas e saídas do programa, as quais são mais visíveis ao usuário.

### Construção do protótipo:

Nesta fase a realização do projeto é realizada.

### Avaliação do protótipo:

Aqui o cliente juntamente com o desenvolvedor avaliam o protótipo.

### Refinamento do Protótipo:

Esta fase funciona como uma peneira onde o cliente e o desenvolvedor escolhem o que continuará ou não no software.

### Construção do produto:

Nesta fase é a essencial, onde o protótipo ( fase de testes) termina e o projeto é construído baseando-se apenas no que o desenvolvedor e o cliente acharam conveniente.

## Espiral

Espiral é um tipo de ciclo de vida onde usa-se pequenos ciclos, de uma certa maneira a forma seqüencial é usada, mas ao mesmo tempo, para cada etapa existe um teste em que a fase é analisada, caso não haja sucesso no protótipo dessa fase pode-se deixá-la inacabada e seguir o projeto adiante e mais tarde voltar a fase inacabada, pulando assim de um ciclo para outro sem alterar o produto do projeto.

## Codifica- remenda

É quando o desenvolvedor cria seu protótipo, faz seus testes, cria seu projeto e apresenta ao cliente, se o mesmo analisar sugerir ou exigir algo o desenvolvedor somente acrescenta as exigências no programa e assim vai fazendo até seu cliente mostrar-se satisfeito.

# processos ágeis

Cada vez mais o mercado procura o melhor trabalho desenvolvido no menor espaço de tempo e com recursos limitados. Assim surgiram os processos ágeis que são um conjunto de metodologias que geram uma base de conceitos para a criação de um projeto de software. Exemplo de processo ágil é Extreme programming (XP).

## Extreme programming

Uma das metodologias ágeis usadas por pequenas e medias empresas, onde existem constantemente mudanças, alterações e ajustes no seu desenvolvimento.

Possui 04 valores fundamentais: comunicação,simplicidade, feedback e coragem.

Por meio destes quatro valores é possível realizar constantemente vários feedbacks, assim analisando e gerando resultados simples e implantando mudanças sempre a procura do melhor e mais satisfatório resultado.

# rup ( processo unificado rational)

O processo unificado visa distribuir em etapas e auxiliar no desenvolvimento de sistemas orientados a objetos.

O RUP acaba se tornando semelhante contudo para cada etapa tem que se alcançar um resultado. Podemos chamar de uma “prova real”, como se fosse em uma conta de matemática, sempre achamos um resultado e para sabermos se está certo, fazemos então a prova real, gerando assim um projeto ( “ uma conta”) sem falhas, com o resultado esperado. Para se exata o RUP divide-se em 4 fases e 9 disciplinas. São elas:

- Concepção;

- Elaboração;

- Construção;

- Transição;

- Modelagem de negócios;

- Requisitos;

-Análise;

- Implementação;

- Teste;

- Implantação;

- Gerência de configuração e mudança;

- Gerência de projetos;

- Ambiente.

# CONCLUSÃO

Por meio deste pude analisar e vivenciar mais as etapas, testes e desenvolvimento por completo de um software, visando a utilidade e exigência do cliente no quesito qualidade do produto, tempo de produção e técnicas de desenvolvimento utilizadas por um profissional de analise de sistemas .

###### REFERÊNCIAS

TANAKA, Simone Sawasaki. Analise de Sistemas I. São Paulo: Pearson Eduacationa, 2009.