

Metodologias de desenvolvimento de sistemas

Curso: Desenvolvimento web e multimédia

Disciplina: Análise e Desenvolvimento de Software

TRABALHO ELEABORADO:

Nº 24873	Sandro Pereira
Nº 18279	Bruno Silva
Nº XXXXX	
Nº XXXXX	

Índice

Índice de figuras	3
Metodologia Evolutiva	4
Modelo incremental	4
Metodologia Clássica.....	5
Modelo RAD (Rapid Application Development).....	5
Metodologia clássica	7
Modelo de Prototipação.....	7
Metodologia clássica	8
Modelo Cascata	8
Metodologia evolutiva	10
Modelo Espiral.....	10

Índice de figuras

Figura 1 – Modelo incremental	4
Figura 2 – Modelo RAD.....	5
Figura 3 – Modelo de Prototipação	7
Figura 4 – Modelo Cascata	8
Figura 5 – Modelo Espiral	10

Metodologia Evolutiva

Modelo incremental

O modelo incremental combina elementos do modelo cascata com a filosofia da aproximação da prototipagem.

O objetivo é trabalhar junto do cliente para descobrir as suas dores, de forma a melhorar até ser obtido o produto final.

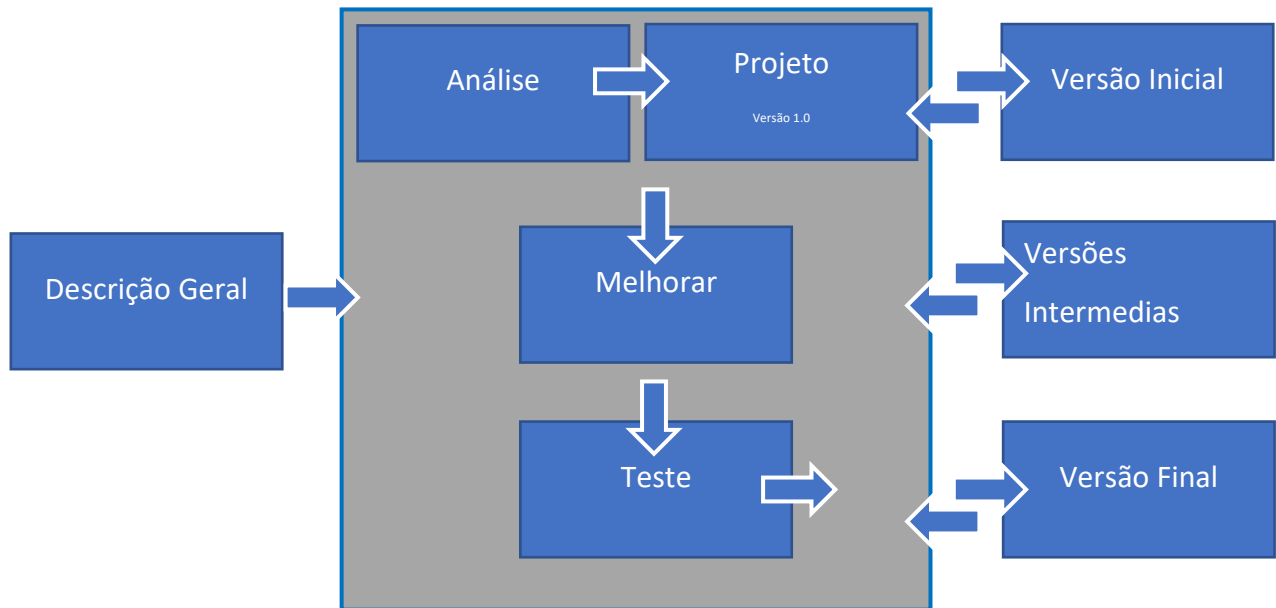


Figura 1 – Modelo incremental

deste modelo.

O objetivo é trabalhar o mais próximo possível do cliente para descobrir as suas necessidades de forma a incrementar até que o produto final seja obtido.

A evolução acontece quando novas características são adicionadas ou sugeridas pela necessidade do cliente.

Desvantagens:

- É um modelo para sistemas pequenos;
- Podem surgir problemas de arquitetura visto não serem conhecidos todos os requisitos no início.

Vantagens:

- Por se entregar vários protótipos e os testar no terreno facilmente se identifica os erros;
- Incrementar em várias fases do modelo, não obriga a muitos meios humanos.

Metodologia Clássica

Modelo RAD (Rapid Application Development)

- É um modelo sequencial linear (igual ao modelo cascata) que se rege por ciclos de desenvolvimento extremamente curtos.
- O desenvolvimento rápido é obtido por uma abordagem de construção baseada em componentes distribuídos por equipe.
- Cada tarefa principal é atribuída a uma equipe independente e só final será junta.

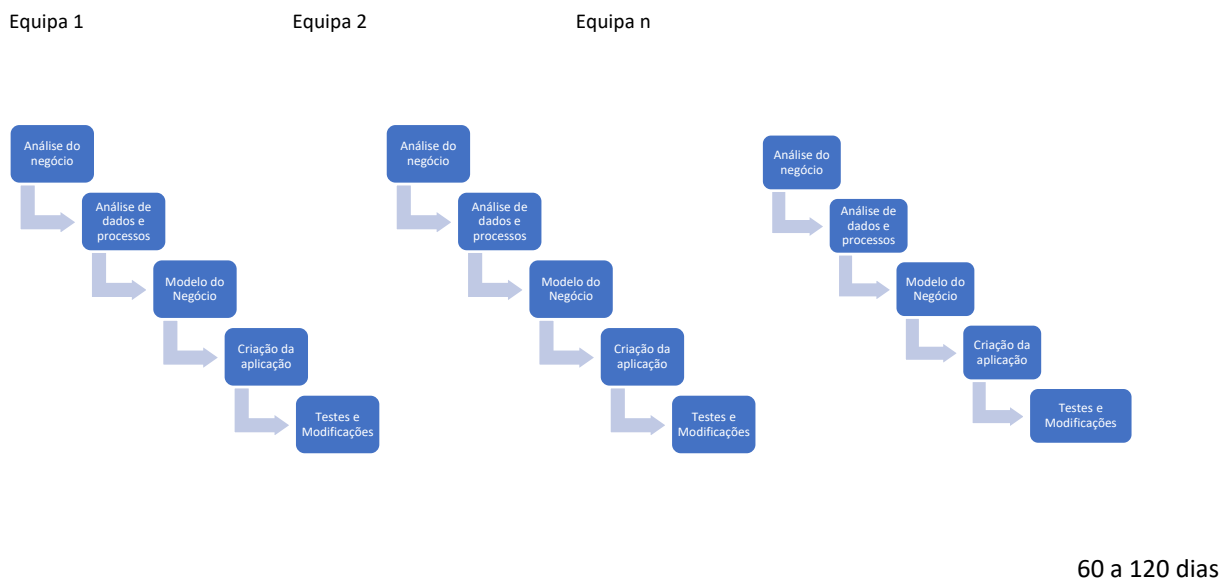


Figura 2 – Modelo RAD

Os cinco passos do modelo:

1. Análise do negócio:

Conhecer/descobrir/recolher as dores do negócio envolvido, bem como dos seus concorrentes.

2. Análise de dados e processos:

Organizar/separar dados recolhidos pelas equipas.

3. Modelo de negócio:

Agilizar/simplificar/automatizar processos.

4. Criar aplicação.

5. Testes, Melhorias, otimizar usabilidade.

Desvantagens:

- São precisos muitos meios humanos;
- Exige que os programadores e o cliente se comprometam a trabalhar de forma dinâmica e em equipe para que os prazos sejam cumpridos.

Vantagens:

- Dividir em várias equipes diminui o tempo e reutiliza os recursos;
- Integração rápida com a introdução dos protótipos muito cedo e por equipe;
- Acompanhamento dos processos de perto.

Metodologia clássica

Modelo de Prototipação

Entender os requisitos do utilizador e, assim, obter uma melhor definição dos requisitos do sistema.

Possibilita que o desenvolvedor crie um modelo (protótipo) do software que deve ser construído e apropriado para quando o cliente, não definiu detalhadamente os requisitos.

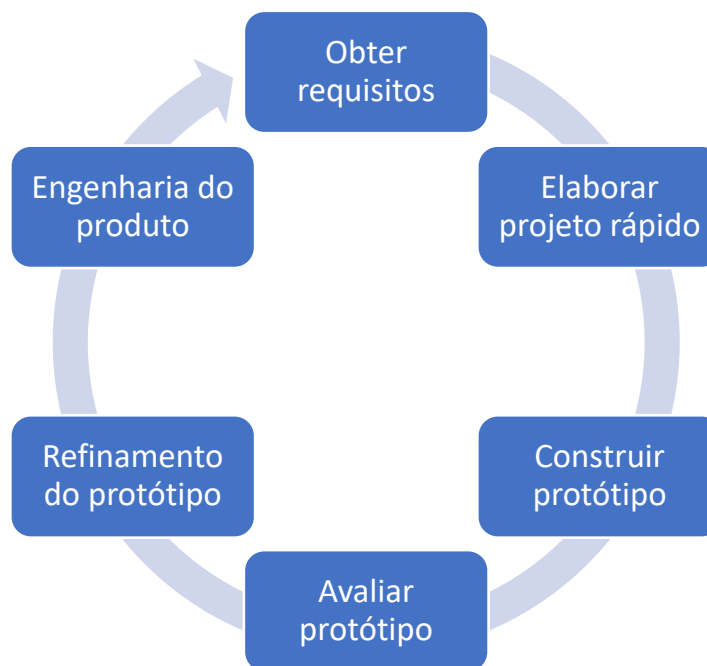


Figura 3 – Modelo de Prototipação

- Obter requisitos
- Elaborar projeto rápido
- Construir protótipo
- Avaliar protótipo
- Refinamento do protótipo
- Engenharia do produto

Metodologia clássica

Modelo Cascata

O modelo cascata é uma abordagem linear de gerenciamento de projetos, que prevê que o projeto só avance a partir do momento que etapa anterior seja concluída, daí provém o nome “cascata”

A utilização do método cascata é indicada em projetos que têm requisitos firmes, ou seja, dos quais não são esperadas alterações.

Primeira etapa “falsa”: Engenharia de Sistemas/ Informação e Modelagem

Modelo cascata é dividido em 6 etapas, sendo que a primeira etapa nem sempre é utilizada. Pois, apenas é requisitada quando o software faz interface com outros elementos como, hardware, banco de dados e pessoas.

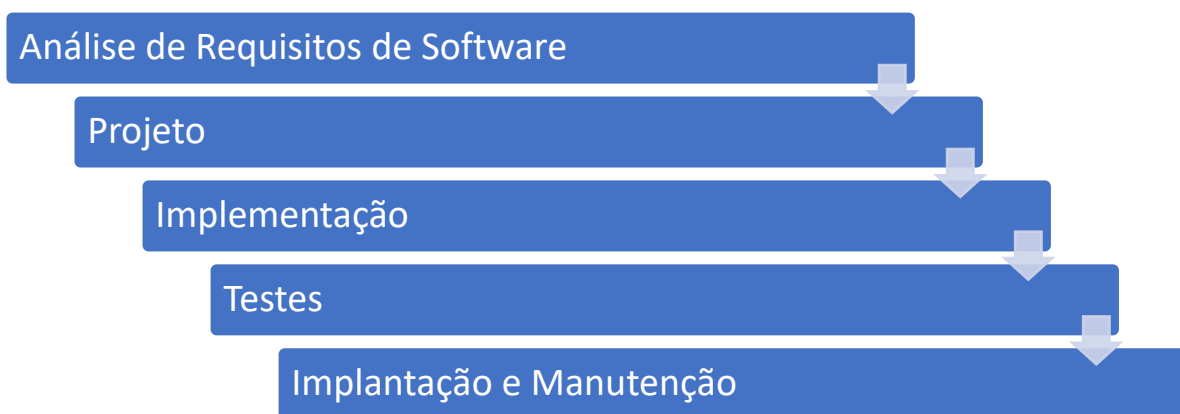


Figura 4 – Modelo Cascata

Primeira etapa “verdadeira”: Análise de Requisitos de Software

Nesta etapa é feita a recolha de requisitos com o cliente, para que desse modo o analista entenda as expectativas que o cliente espera para o seu projeto, e possa definir quais funcionalidades devem ser ou não implementadas no sistema. Há que ter cuidado para que o objetivo do software seja bem interpretado, sendo que se não for as etapas seguintes não podem vir a ser exequíveis.

Segunda etapa: Projeto

Nesta segunda fase é desempenhado um planeamento que segue as seguintes etapas:

- Utilização de um cronograma para organizar as atividades recursos e prazos de um projeto;

- Definir tarefas com suporte dos requisitos;
- Realizar uma estimativa para a finalização de cada etapa;
- Montar uma equipa de desenvolvimento;
- Realizar a programação da interface e arquitetura do sistema;

Terceira etapa: Implementação

É nesta etapa que começa os primeiros passos da equipa de desenvolvimento na codificação do software de acordo com os requisitos e especificações do mesmo.

Quarta etapa: Testes

Após o desfecho da equipa de desenvolvimento no trabalho de implementação é a altura de realizar os testes para verificar se os objetivos foram cumpridos ou se houve algum erro.

Quinta etapa: Implantação e Manutenção

Seguidamente aos testes e à correção de erros estarem realizados é necessário o sistema ser implantado para que o cliente veja o resultado. Se este resultado não for de acordo com as expectativas do cliente é necessário realizar uma mudança e assim sendo, o software deve passar por uma manutenção.

Metodologia evolutiva

Modelo Espiral

O Modelo Espiral está dividido em 4 setores:

1. Estipular objetivos, alternativas e restrições;
2. Avaliar alternativas, identificar, resolver riscos (criar protótipos);
3. Desenvolver, verificar;
4. Planejar.

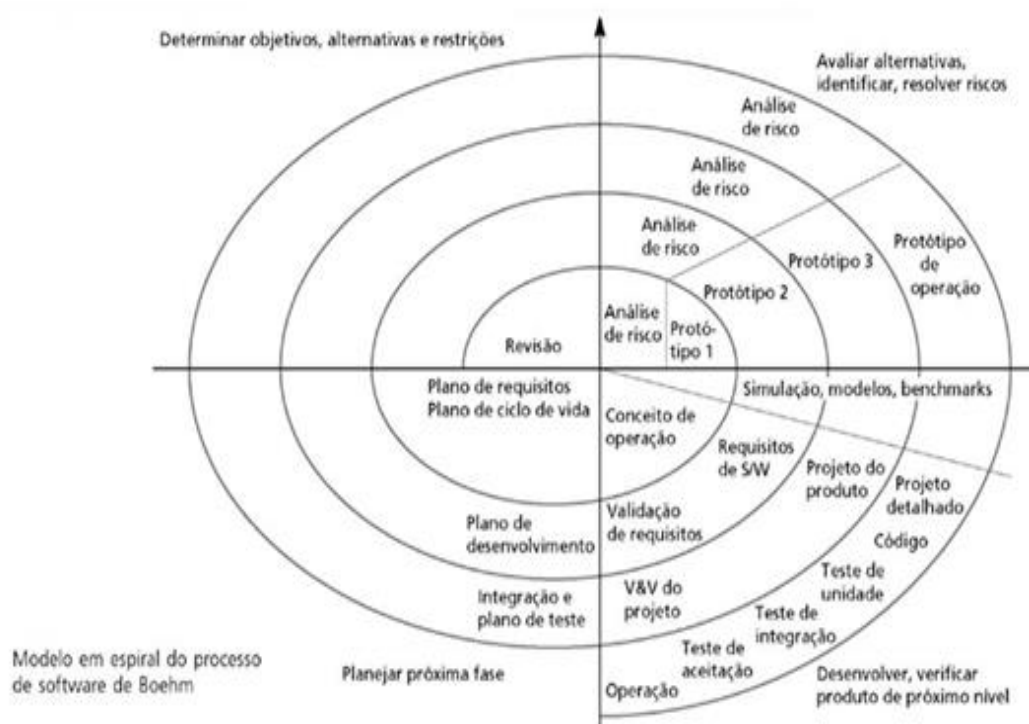


Figura 5 – Modelo Espiral

- Cada vez que chegamos ao fim de um loop de tarefas, será incrementado um nível de loop, este modelo cria assim uma evolução continua insaciável;
- Define objetivos específicos para cada fase do projeto;
- Identifica problemas nos processos;
- Reconhece riscos do projeto e propõe estratégias alternativas;
- Usa a prototipagem para redução de riscos em qualquer fase do loop.