

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](#_Toc117835630)

[Metodologia Evolutiva 4](#_Toc117835631)

[Modelo incremental 4](#_Toc117835632)

[Metodologia Clássica 5](#_Toc117835633)

[Modelo RAD (Rapid Application Development) 5](#_Toc117835634)

[Metodologia clássica 7](#_Toc117835635)

[Modelo de Prototipação 7](#_Toc117835636)

[Metodologia clássica 8](#_Toc117835637)

[Modelo Cascata 8](#_Toc117835638)

[Metodologia evolutiva 10](#_Toc117835639)

[Modelo Espiral 10](#_Toc117835640)

# Índice de figuras

[Figura 1 – Modelo incremental 5](file:///C:\Users\ppcfe\Downloads\trabalho_sexta-8.docx#_Toc117801101)

[Figura 2 – Modelo RAD 6](#_Toc117801102)

[Figura 3 – Modelo de Prototipação 8](#_Toc117801103)

[Figura 4 – Modelo Cascata 9](#_Toc117801104)

[Figura 5 – Modelo Espiral 11](#_Toc117801105)

# 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 – Modelo incremental

Descrição Geral

Melhorar

Teste

Análise

Projeto

Versão 1.0

Versão Inicial

Versões Intermedias

Versão Final

v

v

A versão inicial é frequentemente o núcleo do produto sendo assim a parte mais importante 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 fazes 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 equipa.

- Cada tarefa principal é atribuída a uma equipa independente e só final será junta.

Equipa 1 Equipa 2 Equipa n

60 a 120 dias

Figura – Modelo RAD

**Os cinco passos do modelo:**

1. Análise do negócio:

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

1. Análise de dados e processos:

Organizar/separar dados recolhidos pelas equipas.

1. Modelo de negócio:

Agilizar/simplificar/automatizar processos.

1. Criar aplicação.
2. 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 equipa para que os prazos sejam compridos.

**Vantagens**:

* Dividir em várias equipas diminui o tempo e reutiliza os recursos;
* Integração rápida com a introdução dos protótipos muito cedo e por equipa;
* 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 – 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, dai 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 divido 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 – 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 compridos 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. Planear.

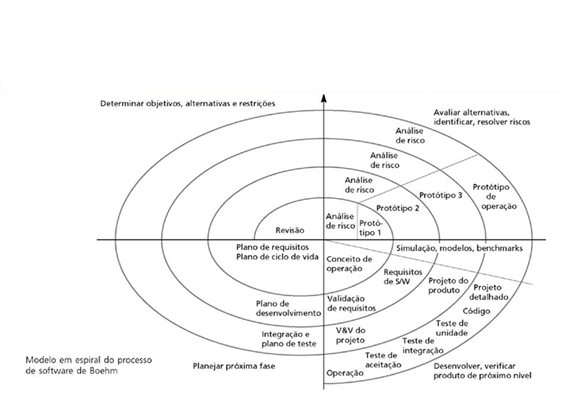


Figura – Modelo Espiral

* 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.