

Algoritmo e Estrutura de Dados I

Desenvolvimento de Software

Profa. MSc. Vânia de Oliveira Borges

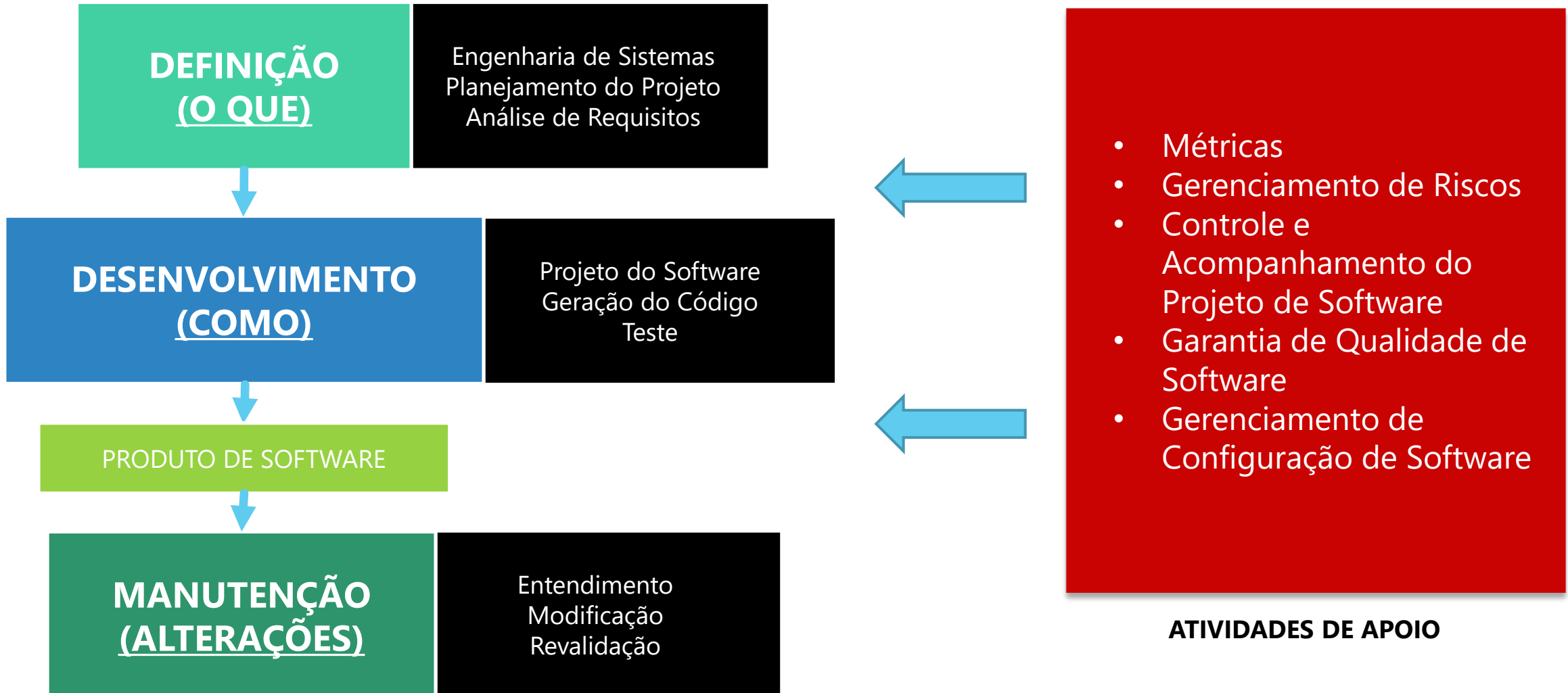
1º semestre de 2024

Software?

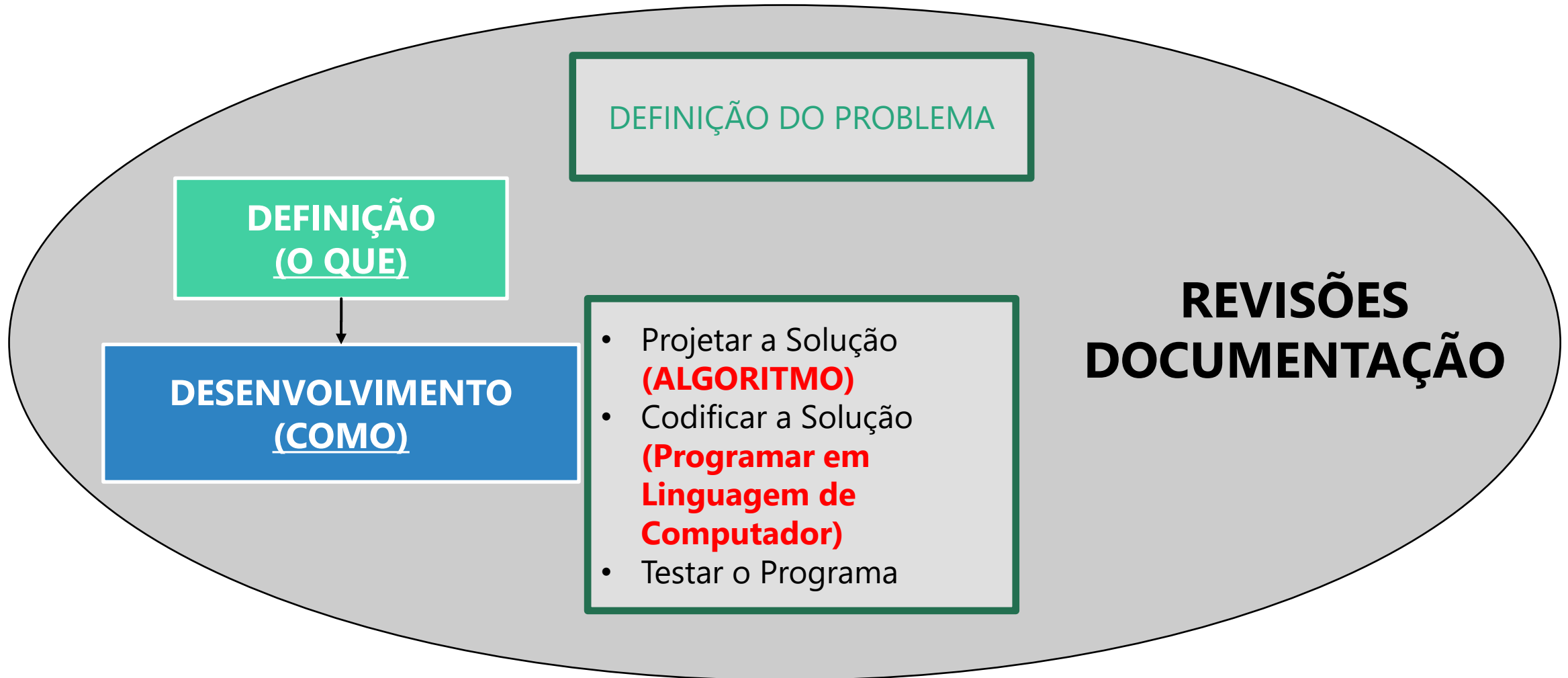
Software

- ▶ É uma sequência de instruções a serem seguidas e/ou executadas, na manipulação, redirecionamento ou modificação de um dado/informação ou acontecimento.

Desenvolvimento de software – 3 fases genéricas



Desenvolvimento de Software



Linguagem de Programação



Linguagem de Programação

- ▶ Forma de comunicação com o computador;
- ▶ Para que um computador eletrônico funcione, necessita ser programado;
- ▶ O processo de programação é uma “conversa” controlada entre um ser humano e o computador;
- ▶ O processo de comunicação se faz com o uso de uma linguagem de programação que o computador entenda;
- ▶ É possível comunicar-se com um computador utilizando linguagem de programação de baixo e alto nível;

Linguagem de Programação

LINGUAGEM DE
BAIXO NÍVEL

LINGUAGEM DE
ALTO NÍVEL

Linguagens de BAIXO NÍVEL

▶ Linguagem de Máquina:

- ▶ Uma CPU somente pode compreender instruções que sejam expressas em termos de sua LINGUAGEM DE MÁQUINA;
- ▶ Um programa escrito em linguagem de máquina consiste de uma série de números binários e é muito difícil de ser entendido pelas pessoas;
- ▶ Exemplo: Cada instrução é constituída de 2 partes:

Código da Operação	Operando
001	01010

Linguagens de BAIXO NÍVEL

► Linguagem de Montagem:

- Possui a mesma estrutura e conjunto de instruções que a linguagem de máquina, porém permite que o programador utilize nomes (chamados mnemônicos) e símbolos em lugar de números.

► ASSEMBLY;

► Exemplo:

Código da Operação	Operando	Significado
LD	A	load A
MPI	5	multiplica 5

Linguagem de Programação

► Linguagem de Alto Nível:

- São linguagens de programação nas quais se pode escrever programas em uma notação próxima à maneira natural de expressar o problema que se deseja resolver;
- Possuem uma estrutura e palavras-chave que são mais próximas da linguagem humana. Tornando os programas mais fáceis de serem lidos e escritos;
- C, JAVA, FORTRAN, COBOL, VB;
- Exemplo: $\text{RESULT} = D - ((A + B) / C);$

Tradutores



Tradutores

▶ MONTADOR

▶ Efetua a tradução de linguagem de montagem (Assembly) para a linguagem de máquina, da seguinte forma:

- ▶ obtém a próxima instrução do Assembly;
- ▶ traduz para as instruções correspondentes em linguagem de máquina;
- ▶ executa as instruções em linguagem de máquina;
- ▶ repete o passo 1 até o fim do programa.



Tradutores

► INTERPRETADOR

- Efetua a tradução a de uma linguagem de alto nível para linguagem de máquina da seguinte forma:
 - obtém próxima instrução do código-fonte em linguagem de alto nível;
 - traduz para as instruções correspondentes em linguagem de máquina;
 - executa as instruções em linguagem de máquina;
 - repete o passo 1 até o fim do programa;
- Traduz e executa cada linha por vez.



Tradutores

► COMPILADOR

- Efetua a tradução de todo o código-fonte em linguagem de alto nível para as instruções correspondentes em linguagem de máquina, gerando o código-objeto do programa. Em seguida é necessário o uso de um outro programa (Link-Editor) que é responsável pela junção de diversos códigos-objeto em um único programa executável;
- Executa o programa após todas as instruções serem traduzidas;



Paradigmas de Programação

- ▶ Metodologia que cada linguagem usa para realizar a comunicação com a máquina;
- ▶ Fornece e determina a visão que o programador possui sobre a estruturação e execução do programa;
- ▶ Estilo de Programação.

Paradigmas de Programação - Estruturado

▶ PROGRAMAÇÃO ESTRUTURADA

- ▶ Programas reduzidos a 3 estruturas;
- ▶ Criação de estruturas simples;
- ▶ Utilizando sub-rotinas e funções;



Estruturas de sequência:

Onde uma tarefa é executada após a outra, linearmente.



Estruturas de decisão:

Onde, a partir de um teste lógico, determinado trecho de código é executado, ou não.



Estruturas de iteração:

Onde, a partir de um teste lógico, determinado trecho de código é repetido por um número finito de vezes

Paradigmas de Programação – Orientado a Objetos

- ▶ É um paradigma que tem como principal característica a manipulação de objetos que interagem entre si;
- ▶ Objetos são gerados a partir de classes;
- ▶ Classes são estruturas que agrupam características (atributos) e funcionalidades (métodos) comuns a um grupo de objetos;
- ▶ Os objetos comunicam entre si;
- ▶ Ex.: Java, C++, C#;



Como o cliente explicou



Como o lider de projeto entendeu



Como o analista planejou



Como o programador codificou



O que os beta testers receberam



Como o consultor de negocios descreveu



Valor que o cliente pagou



Como o projeto foi documentado



O que a assistencia tecnica instalou



Como foi suportado



Quando foi entregue



O que o cliente realmente necessitava

Cliente não sabe o que quer...

- ▶ Cliente **NÃO** sabe o que quer ou nem sempre consegue expressar o que pensa;
- ▶ E em muitos casos, explicam totalmente o contrário do que realmente queriam explicar;
- ▶ Motivos:
 - ▶ Falta de conhecimento de tendências
 - ▶ Falta de conhecimento técnico
 - ▶ **Conhecimento de mundos diferentes...**

Falha de comunicação...

- ▶ Você entendeu realmente o que seu cliente precisa? Como você validou isso?
- ▶ E seu cliente, consegue entender quando você diz que as mensagens do sistema serão exibidas através de uma Janela Modal?



Falha de comunicação...



Mudanças de requisitos...

Seu Manuel, precisamos
fazer uma pequena
mudança!



Estimativas fantasiosas...

- ▶ Gasto cerca de **20 minutos** de **casa ao trabalho**...



Estimativas fantasiosas...

► **Será?**



Conclusão

- ▶ Não é complicado programar, é complicado desenvolver uma solução que automatize as tarefas do usuário final;
- ▶ É preciso planejar
 - ▶ Criar procedimentos;
 - ▶ Investir tempo em construir modelos e processos;
 - ▶ Validar modelos e processos;