

# Resumos de AS

## Teste Teorico

Tiago Almeida

June 18, 2024

## Contents

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Diagramas UML</b>	<b>2</b>
2.1	Activity Diagram . . . . .	2
2.2	Use Case Diagram . . . . .	2
2.3	Class Diagram . . . . .	2
2.4	Sequence Diagram . . . . .	2
2.5	State machine Diagram . . . . .	2
2.6	Package Diagram . . . . .	2
2.7	Component Diagram . . . . .	2
2.8	Deployment Diagram . . . . .	2
<b>3</b>	<b>SDLC</b>	<b>2</b>
<b>4</b>	<b>Processos de software</b>	<b>3</b>
<b>5</b>	<b>Visão geral do OpenUP/Unified Process</b>	<b>5</b>
<b>6</b>	<b>Conclusão</b>	<b>5</b>
<b>7</b>	<b>Glossário</b>	<b>6</b>

## 1 Introdução

Escrever um pequeno overview da matéria que sai para o teste teórico 2 e o que esperar encontrar neste documento

## 2 Diagramas UML

### 2.1 Activity Diagram

### 2.2 Use Case Diagram

### 2.3 Class Diagram

### 2.4 Sequence Diagram

### 2.5 State machine Diagram

### 2.6 Package Diagram

### 2.7 Component Diagram

### 2.8 Deployment Diagram

## 3 SDLC

SDLC (System Development Life Cycle), consiste em 4 fases fundamentais:

#### 1. Planeamento

- Fase onde se discute o motivo da existencia do sistema e como construí-lo;
- O valor do sistema para a organização é identificado

#### 2. Análise (Investigação)

- Fase onde se discute sobre quem irá utilizar o sistema, o que o sistema deve fazer e onde e como será utilizado
- Comunicação com os stakeholders para identificar as necessidades e propor soluções.

#### 3. Desenho (projeto Técnico)

- Fase onde se decide como o sistema irá ser construido, seja em termos de hardware, software, infraestrutura de rede, interface, formulários e relatórios de utilizador, programas especificos, bases de dados e ficheiros que serão necessários.

#### 4. Implementação

- Fase onde o sistema é construido

**Mnemónica: PADI.**

**NOTA:** Existem passos para cada fase que eu não inclui aqui. Se vier a ser importante, voltar ao slide ‘AS-TP05b Processos de software.pdf’ (páginas 5 a 8) e adicionar.

## 4 Processos de software

O SDLC é concretizado usando um **processo de software** sistemático

Um processo de software é um guião para as atividades e tarefas que são necessárias para construir software de qualidade

Existem duas metodologias que se podem seguir ao criar software:

- Linear/estruturada
  - Waterfall
- Evolutiva
  - Prototyping
  - Spiral model
  - Métodos ágeis

Na primeira metodologia, o método **Linear/estruturado**, todas as atividades a serem realizadas são planeadas em avanço e o progresso feito é medido tendo em conta o plano já feito.

O exemplo clássico da metodologia Linear/estruturada é o modelo *Waterfall*

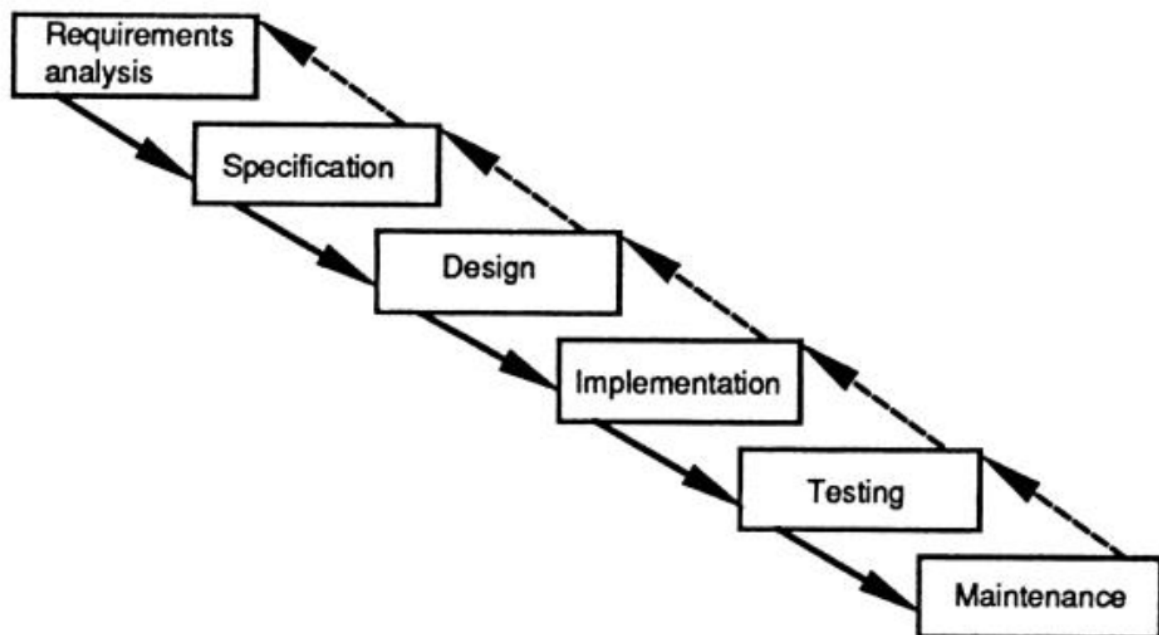


Figure 1: Modelo Waterfall

**Vantagens do modelo Waterfall:**

- Facil de planear: Uma agenda pode ser feita com deadlines para cada fase criada.
- Facil de gerenciar: Cada fase tem um processo específico
- Cada fase é feita uma de cada vez, e apenas se avança para a proxima quando se acabar a anterior
- Modelo funciona bem quando os requerimentos são simples, estáveis e bem entendidos por quem está a trabalhar

### Desvantagens do modelo Waterfall:

- Não é flexível: Dificil de se adaptar caso acha alguma mudança
- Para projetos de longo prazo, o modelo criado no inicio pode ficar desatualizado e não ser mais ideal
- Projetos reais raramente conseguem seguir um modelo 100% sem ter de fazer mudanças
- Uma vez que uma versão que trabalhe só é feita no final, o cliente vai ter de esperar até o final do projeto para ver resultados
- O custo de corrigir um erro vai crescendo conforme o projeto avança

Na segunda metodologia, o metodo **Evolutivo**, o planeamento é feito de forma incremental ao longo do projeto de forma a ser possivel a adaptação às necessidades dos clientes.

Existem alguns modelos da metodologia Evolutiva apresentada:

#### 1. Prototyping

- Consiste em criar um prototipo desde o inicio que os stakeholders devem concordar que serve para definir os requerimentos.
- O prototipo é descartado e o sistema real é começado a ser criado com olho na qualidade.

#### 2. Spiral Model

- Neste modelo, o projeto passa pelas diferentes fases da evolução repetidas vezes, incluindo a fase de *Deployment*.
- Nas iterações iniciais, o projeto liberado ao publico pode ser só um modelo ou protótipo. Nas iterações mais avançadas as versão liberadas vão ficando cada vez mais completas.

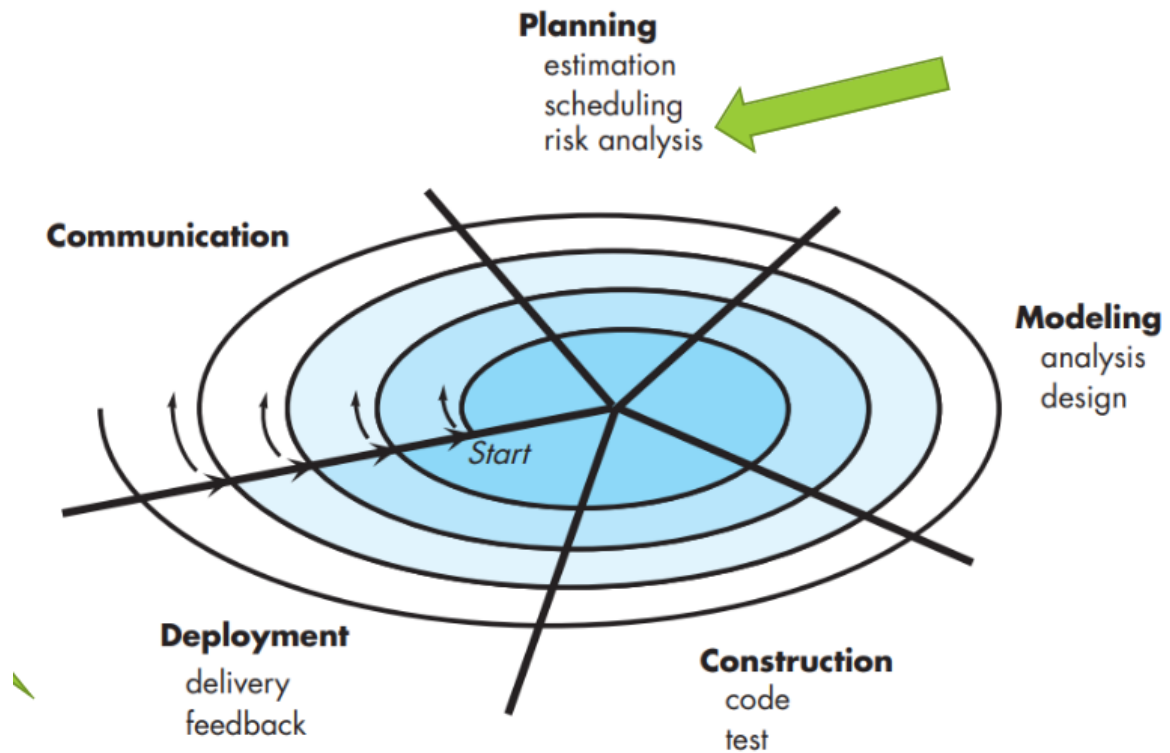


Figure 2: Modelo Waterfall

### 3. Métodos ageis

•

**Nota:** Nos processos de desenvolvimento apresentados acima, o SDLC está sempre presente.

## 5 Visão geral do OpenUP/Unified Process

## 6 Conclusão

Algumas conclusões e considerações que se deve ter após ter acabado o estudo

## 7 Glossário

Aqui está a secção de glossário. Cada termo usado repetidamente no documento está listado aqui com sua definição.

- buh