



A <PROGRAMAÇÃO> PRECISA DE TI

Software Development Life Cycle

Laboratório Web

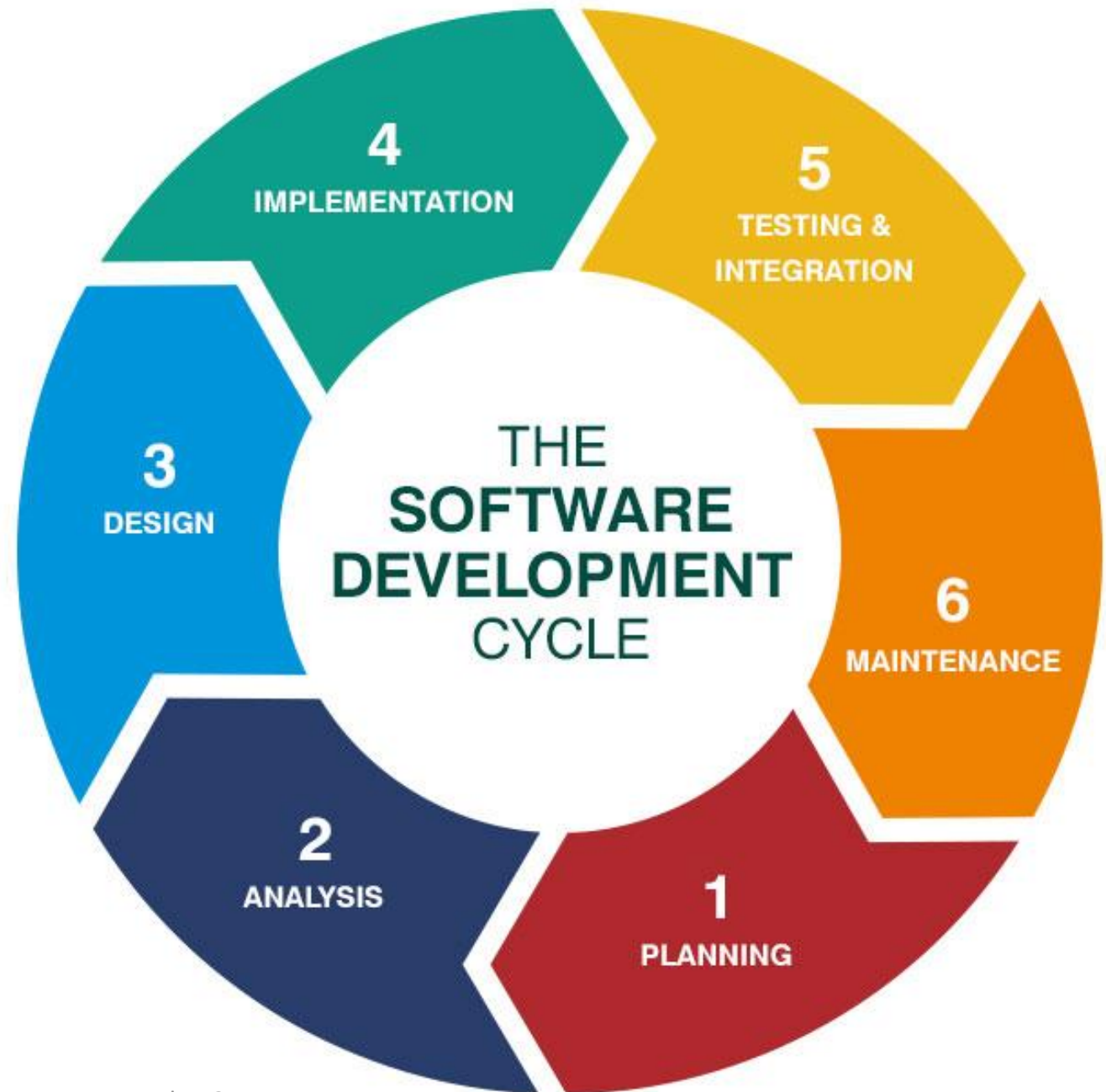
Software Development Life Cycle

- SLDC é uma framework que descreve as atividades realizadas em cada fase de desenvolvimento de um projeto de software
- O SLDC é usado pela indústria de software para desenhar, desenvolver e testar software
- O objetivo é produzir software de qualidade de acordo com as expectativas do cliente, respeitando o prazo de execução e orçamento financeiro

Software Development Life Cycle

- ISO/IEC 12207 [link](#)
- Introduzida em 1995 , é uma norma que define os processos padrão para desenvolvimento de software
- O objetivo é dividir o processo de desenvolvimento de software em sub-processos mais pequenos, que podem ser paralelos ou sequenciais de forma a melhorar o desenho, implementação e gestão do desenvolvimento

Fases do Software Development Life Cycle



Fases do Software Development Life Cycle

1. Planeamento e análise de requisitos
2. Definição dos requisitos
3. Desenho conceptual do software
4. Implementação/desenvolvimento
5. Testes
6. *Deployment* e manutenção

1. Planeamento e análise de requisitos

1. Esta fase é a mais importante de todo o SDLC
2. É executado pelos elementos séniores da equipa tendo em conta a informação providenciada pelas partes interessadas e especialistas do domínio
3. Planeamento dos requisitos de QA (*Quality Assurance*)
4. Identificação dos riscos associados com a execução do projeto

2. Definição dos requisitos

- Após a análise de requisitos estar concluída o passo seguinte é efetuar a definição dos requisitos
- Consiste em definir e documentar de forma clara os requisitos do software e obter aprovação das partes interessadas (*stakeholders*)
- Isto é efetuado através do documento SRS (Software Requirement Specification)

3. Desenho conceptual do software

- Baseado nos requisitos do especificados no documento SRS são propostas várias arquiteturas para o produto
- Estas propostas são documentadas num documento designado por DDS (Design Document Specification)
- Este documento é analisado por todas as partes interessadas e tendo em conta a arquitetura, riscos, orçamento e restrições temporais a melhor abordagem é escolhida

4. Implementação/desenvolvimento

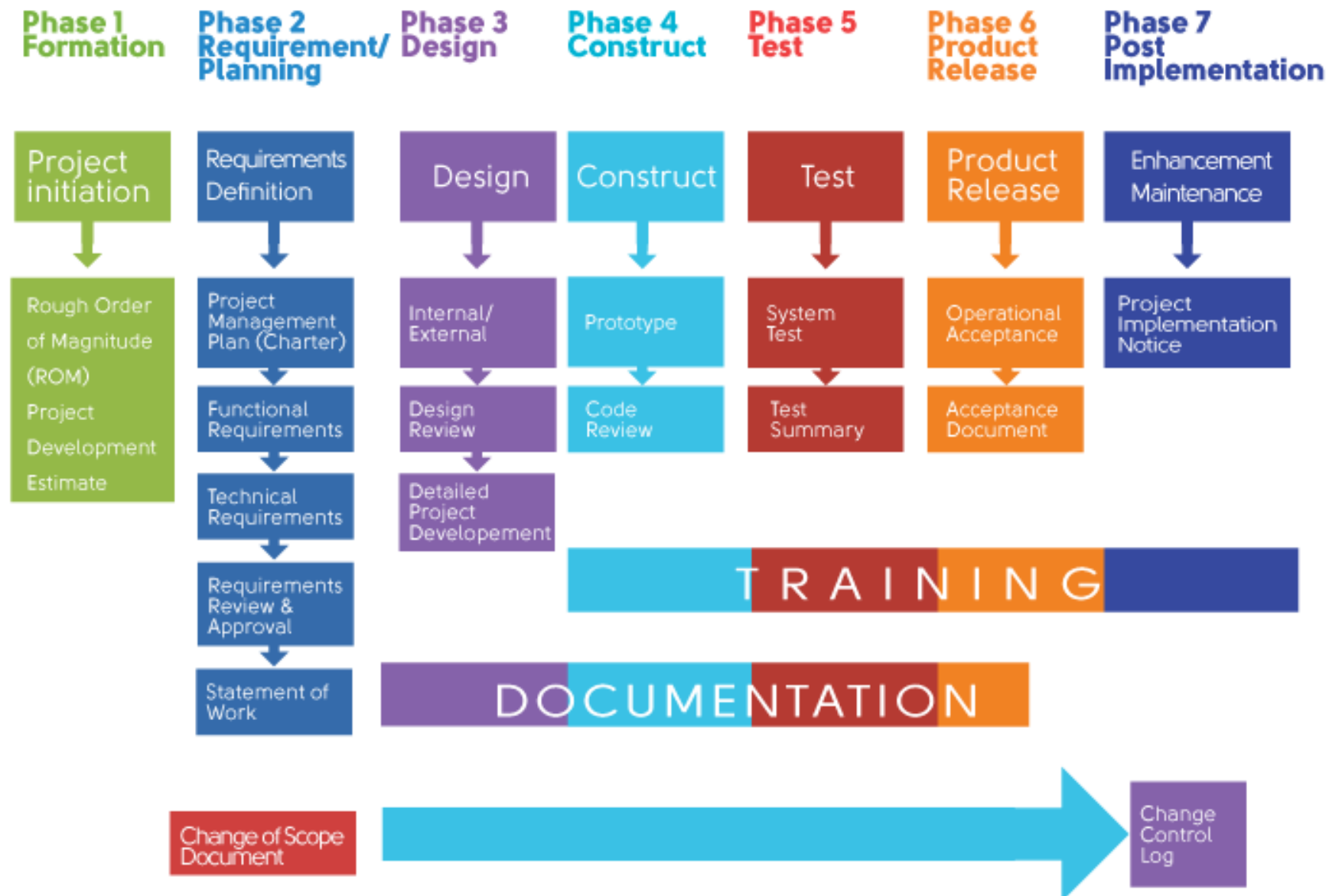
- Nesta fase é onde o atual desenvolvimento é iniciado e o produto implementado
- O software é implementado de acordo com as especificações documentadas no DDS (Design Document Specification)
- Aqui os programadores devem seguir as diretrizes definidas pela organização e ferramentas como compiladores, IDE's, e outras aplicações são utilizadas para implementar o código

5. Testes

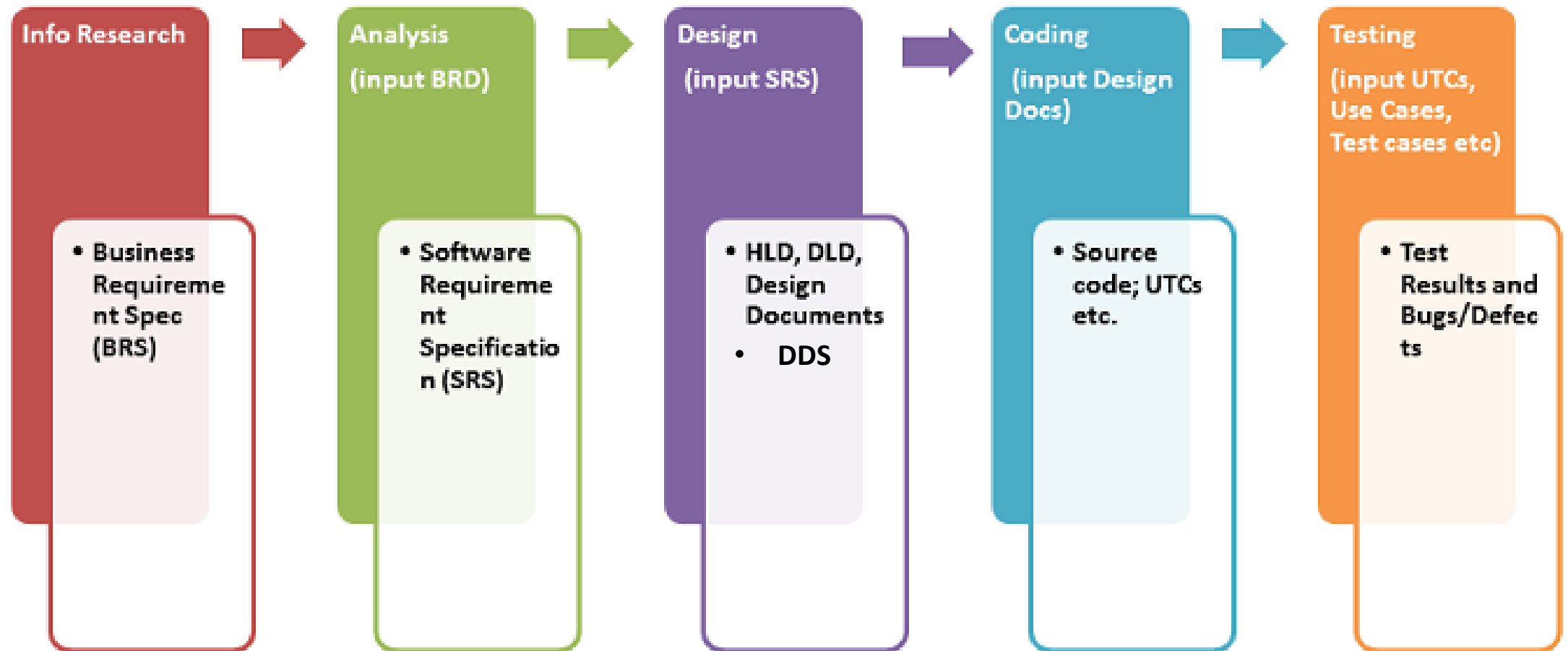
- Atualmente esta fase está envolvida em quase todas as outras fases do SDLC
- Originalmente esta fase referia-se apenas aos testes realizados ao software onde os erros encontrados são:
 - Reportados, seguidos, corrigidos e testados novamente
- Esta fase decorre até que o software alcance os padrões de qualidade documentados no SRS

6. Deployment e manutenção

- A partir do momento em que todas as funcionalidades foram implementadas e testadas (sem existência de erros) o produto pode ser *deployed*/lançado
- O produto será colocado num ambiente de produção para ser utilizado pelos utilizadores finais
- Após o *deployment* é iniciada a fase de manutenção para garantir o pleno funcionamento do produto



Fases e documentação



Modelos SDLC

- Ao longo do tempo surgiram diversos modelos de SDLC
- Estes modelos providenciam:
 - padrões para planejamento, estimativa e calendarização
 - padrões para terminologias, atividades e entregáveis
 - mecanismos de controlo e acompanhamento dos projetos
 - aumento da visibilidade da execução e progresso do projeto para todas as partes interessadas

Vantagens de utilizar o modelo apropriado

- Maior velocidade de desenvolvimento
- Maior qualidade na entrega do software
- Controlo e acompanhamento
- Introduz dinâmicas positivas entre cliente e fornecedor
- Mitigação de riscos
- Gestão estruturada com funções bem identificadas/definidas

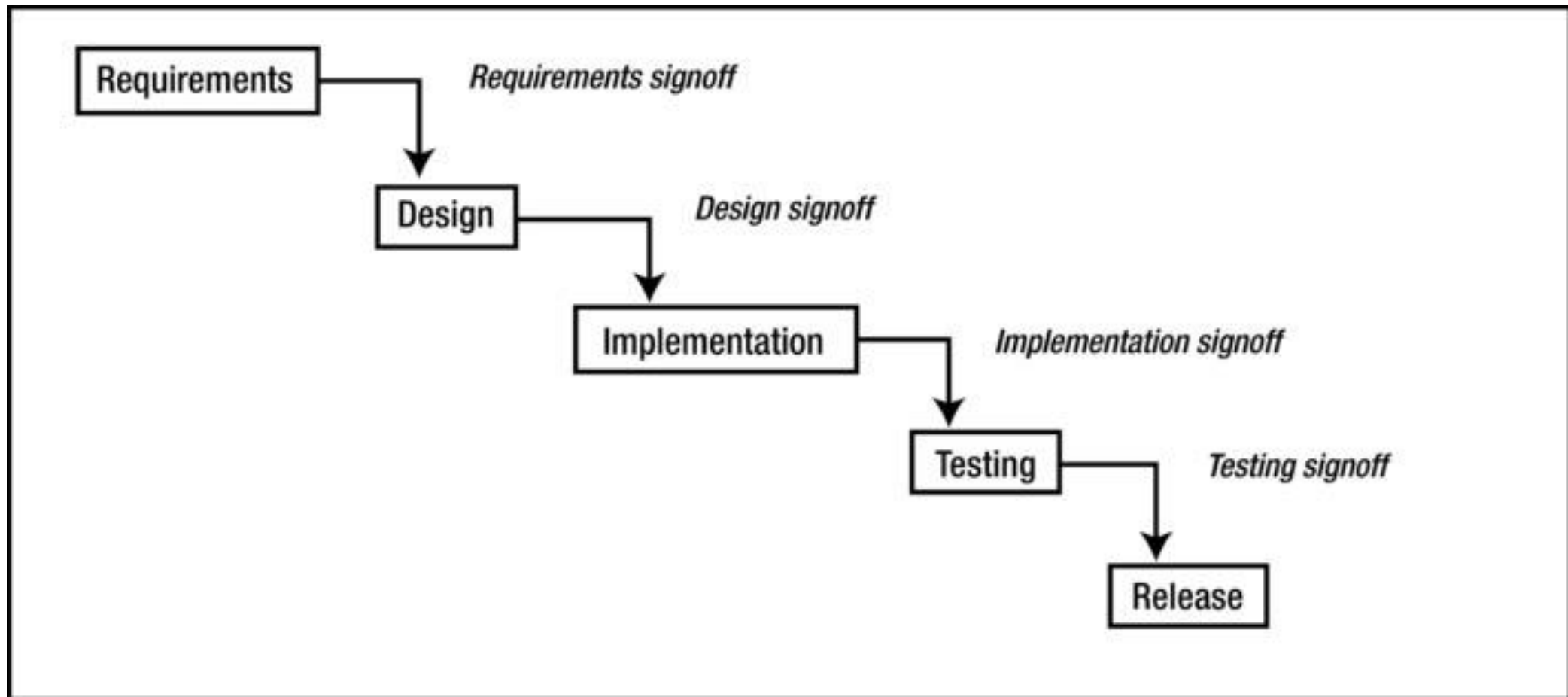
Modelos SDLC

- Modelo Big Bang
- **Modelo Waterfall**
- Modelo Iterativo
- Modelo Spiral
- **Modelo Agile**

Modelo Waterfall

- Modelo SDLC mais conhecido (e mais antigo)
- Baseado em processos de construção, manufatura e engenharia
- Segue um conjunto de passos estritamente sequenciais
- Adequado para projetos onde os requisitos são bem conhecidos e bem definidos

Modelo Waterfall



Modelo Waterfall - Vantagens

- Fácil de entender, logo de utilizar
- Providencia estrutura a elementos da equipa júniores
- Milestones são perceptíveis e bem definidos
- Ideal para gestão e controlo (planear e acompanhar equipa/desenvolvimento)
- Focado mais em qualidade do que custos ou calendarização

Modelo Waterfall - Desvantagens

- Todos os requisitos tem que ser definidos à priori
- Os entregáveis para cada fase são considerados imutáveis
- Não existe uma iteração das suas fases
- Integração apenas ocorre no fim do projeto
- O cliente só tem acesso ao desenvolvimento/progressos na fase final

Modelo Waterfall – Quando utilizar?

- Quando **todos** os requisitos são conhecidos e bem definidos
- A definição do produto/software é estável
- Todas as tecnologias envolvidas são conhecidas
- Para implementação de uma nova versão de um produto existente
- Migração de um produto existente para uma nova plataforma

qualificar

TAL

X

</>

A <PROGRAMAÇÃO