

## PROTOTIPAÇÃO

Um protótipo é uma **versão inicial** de um sistema de software.

Usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções.

O desenvolvimento rápido e iterativo do protótipo é **essencial para que os custos sejam controlados** e os stakeholders do sistema possam experimentá-lo no início do processo de software.

Um protótipo de software pode ser usado em um processo de desenvolvimento de software para **ajudar a antecipar as mudanças que podem ser requisitadas:**

- No processo de engenharia de requisitos, um protótipo **pode ajudar na elicitação e validação de requisitos de sistema.**
- No processo de projeto de sistema, um protótipo pode ser usado para estudar soluções específicas do software e para apoiar o projeto de interface de usuário.

Os **usuários podem obter novas ideias para requisitos** e encontrar pontos fortes e fracos do software, podem então, propor novos requisitos do sistema.

Enquanto o sistema está em projeto, um protótipo do sistema pode ser usado para a realização de experimentos de projeto visando à verificação da viabilidade da proposta.

Além disso, o desenvolvimento do protótipo **pode revelar erros e omissões nos requisitos propostos.**

A função descrita em uma especificação pode parecer útil e bem definida.

## MANUTENÇÃO

### Definição

É o processo geral de mudança em um sistema depois que ele é liberado para uso.

As mudanças são implementadas por meio da modificação de componentes do sistema existente.

Quando necessário, adicionar novos componentes.

### Características

As alterações feitas no software podem ser:

- Correção de erros de codificação;
- Correção de erros de projeto;
- Corrigir erros de especificação; ou
- Acomodar novos requisitos.

### Tipos de Manutenção

**Existem três tipos de manutenção** de software:

- Correção de defeitos 17% do custo total.
- Adaptação ambiental 18% do custo total.
- Adição de funcionalidade 65% do custo total.

Erros de codificação são relativamente baratos para serem corrigidos.

Erros de projeto são mais caros, pois podem implicar reescrever vários componentes de programa.

Erros de requisitos são os mais caros para se corrigir devido ao extenso reprojeto de sistema que pode ser necessário.

- *Adaptação Ambiental*

Esse tipo de manutenção **é necessário quando algum aspecto do ambiente do sistema sofre mudança.**

Como o hardware;

A plataforma do sistema operacional; ou

Outro software de apoio.

- *Adição de Funcionalidade*

Esse tipo de manutenção é necessário quando os requisitos de sistema mudam em resposta às mudanças organizacionais ou de negócios.

Adicionar uma nova funcionalidade após a liberação é caro porque é necessário tempo para aprender o sistema e analisar o impacto das alterações propostas.

## QUALIDADE DE SOFTWARE

Um produto que apresenta um grau de satisfação das necessidades dos clientes sob todos os aspectos.

Em que momento temos que incorporar a qualidade?

Necessita estar presente em todo o processo de desenvolvimento!

### Características

Funcionalidade – atender requisitos do cliente

Confiabilidade – É imune a falhas?

Usabilidade – É fácil de usar?

Eficiência – É rápido e “enxuto”=essencial?

Manutenibilidade – É fácil de manutenção?

Portabilidade – É fácil de usar em outro ambiente?

## TESTE DE SOFTWARE

Atividade de executar um programa e verificar se o seu comportamento é o esperado.

Objetivo: Revelar defeitos.

### Erro, Defeito e Falha

**Erro** (error): é fruto da ação humana, que produz um resultado incorreto.

**Defeito** (fault): também conhecido como bug, é o resultado de um erro no código, funcionamento no sistema.

**Falha** (failure): é resultado da execução de um defeito no código.

### Níveis de Teste

- **Teste de Aceitação:** Verifica se o sistema está em conformidade com os requisitos esperados pelo cliente. Realizado pelo cliente ou pelo testador. O sistema é utilizado para capacitação dos usuários de forma que eles validem todos os requisitos do sistema. Realizado de forma manual ou automática.
- **Teste de Sistema:** Verifica se o sistema está em conformidade com a especificação de requisitos. Realizado pelo testador, o qual tem acesso apenas a interface do sistema. Realizado de forma manual ou automática.
- **Teste de Integração:** Verifica se ao juntar vários componentes do sistema, se eles se comunicam corretamente. Realizado pelos desenvolvedores ou analistas de sistema para testar um módulo do sistema. Realizado de forma automática.
- **Teste de Unidade:** Teste realizado em uma unidade ou componente para verificar sua corretude. Realizado de forma automática.
  - **Caixa-Branca(estrutural):** Teste de Unidade; Teste de Integração
  - **Caixa-Preta(funcional):** Testes Funcionais; Testes de Aceitação; Testes Exploratórios