

Ciclo de vida de Software

A fase de projeto: do "o que fazer" para o "como fazer"

A fase de **projeto** é a etapa em que as ideias e requisitos levantados na fase de análise se transformam em um blueprint, ou um mapa detalhado, para a construção do software. É o momento de definir a arquitetura, a estrutura e os componentes do sistema, traduzindo as necessidades de negócio em uma especificação técnica que a equipe de desenvolvimento possa seguir. Se a fase de análise respondeu ao "O que?", a fase de projeto foca no "**Como?**".

Stakeholders envolvidos

Nesta fase, a equipe técnica assume um papel mais proeminente, mas a colaboração com outras partes interessadas continua sendo vital. Os principais stakeholders são:

- **Arquiteto de Software:** Geralmente, um desenvolvedor sênior ou líder técnico. Ele é o principal responsável por projetar a arquitetura do sistema e tomar as decisões técnicas de alto nível.
 - **Equipe de Desenvolvimento:** Os programadores e engenheiros de software que irão construir o produto. Eles participam do processo de projeto, oferecendo feedback sobre a viabilidade das soluções propostas e estimando o esforço necessário.
 - **Gerente de Projeto:** Continua a supervisionar o andamento do projeto, garantindo que o plano de design se alinhe com o cronograma e o orçamento.
 - **Analista de Sistemas/Negócios:** Mantém a comunicação entre a equipe técnica e os stakeholders do negócio, garantindo que o projeto técnico ainda atenda aos requisitos originais.
-

Tarefas e trabalhos executados

A fase de projeto é onde o design do software é detalhado, desde a sua arquitetura geral até os componentes individuais. As principais atividades incluem:

1. **Design da Arquitetura do Software:** Esta é a atividade mais crucial. A equipe decide a estrutura geral do sistema. Por exemplo:
 - **Arquitetura Monolítica vs. Microsserviços:** O sistema será um único e grande aplicativo ou um conjunto de serviços menores e independentes?
 - **Padrões de Design:** Que padrões de design (como MVC - Model-View-Controller) serão usados para organizar o código?

- **Tecnologias:** Que linguagens de programação, frameworks e bancos de dados serão utilizados?
2. **Design de Banco de Dados:** O banco de dados é o coração do sistema. Nesta etapa, a equipe projeta as tabelas, as relações entre elas e define como os dados serão armazenados e recuperados de forma eficiente. Um modelo de dados (como um **Diagrama de Entidade-Relacionamento**) é criado para visualizar essa estrutura.
3. **Design da Interface do Usuário (UI) e da Experiência do Usuário (UX):** Enquanto a fase de análise pode ter gerado protótipos de baixa fidelidade, aqui o design é aprofundado. A equipe de UI/UX cria wireframes, mockups e protótipos interativos de alta fidelidade para definir a aparência e o fluxo de navegação do software.
4. **Desenho de Componentes e Módulos:** O sistema é dividido em componentes menores e mais gerenciáveis. Para cada componente, a equipe documenta suas responsabilidades, como ele se comunica com outros componentes e quais são os requisitos técnicos para sua implementação.
5. **Plano de Testes:** Antes de a codificação começar, um plano de testes é elaborado. Ele detalha os tipos de testes a serem realizados (**unidade, integração, sistema, aceitação**), quem será responsável por eles e quais ferramentas serão usadas.
-

Técnicas e métodos

Para a execução das tarefas, uma variedade de técnicas e documentos são produzidos, cada um com um propósito específico:

- **Diagramas UML (Unified Modeling Language):** São amplamente utilizados para visualizar e documentar o projeto. Alguns dos mais comuns são:
- **Diagrama de Classe:** Mostra as classes do sistema, seus atributos e métodos, e como elas se relacionam.
- **Diagrama de Sequência:** Descreve a interação entre objetos e a ordem em que as mensagens são enviadas.
- **Diagrama de Componentes:** Representa como os componentes de software se organizam e se conectam.
- **Wireframes e Mockups:** Criados por designers para visualizar a interface do usuário. Wireframes são esboços simples que mostram a estrutura, enquanto mockups são designs mais detalhados com cores, tipografia e imagens.
- **Modelagem de Dados:** O **Diagrama de Entidade-Relacionamento (DER)** é uma ferramenta fundamental para modelar o banco de dados. Ele define as entidades (objetos do mundo real), seus atributos e as relações entre elas.

- **Design Patterns:** A equipe utiliza padrões de design de software que são soluções comprovadas para problemas recorrentes. Isso economiza tempo, melhora a qualidade do código e facilita a manutenção.

A fase de projeto transforma a visão do produto em um plano concreto e executável. Um projeto bem-sucedido permite que a fase de implementação prossiga de forma fluida, com a equipe de desenvolvimento tendo clareza sobre o que e como deve construir. A falta de um bom projeto pode levar a problemas de arquitetura, retrabalho e um sistema difícil de dar manutenção no futuro.

Saídas Esperadas da Fase

Ao final da fase de Projeto, a equipe deve possuir:

- Documento de design de software (SDS).
 - Protótipos de interface (wireframes, mockups).
 - Modelagem do banco de dados pronta.
 - Diagramas arquiteturais (componentes, classes, implantação).
 - Definição de APIs e contratos de comunicação.
 - Padrões de qualidade, segurança e codificação.
-

Importância da Fase

- **Evita retrabalho** na programação, já que os detalhes são definidos antes da implementação.
- **Reduz custos:** mudanças no design são mais baratas que correções em código pronto.
- **Facilita manutenção:** sistemas com boa arquitetura são mais fáceis de evoluir.
- **Melhora a comunicação** entre equipe técnica e stakeholders.