PROCESSO DE DESENVOLVIMENTO

1. **Levantamento de Requisitos**

Etapa de compreensão do problema aplicada ao desenvolvimento do software. Os desenvolvedores, juntamente com os clientes, tentam levantar e definir as necessidades dos futuros usuários.

Os requisitos são identificados a partir do domínio do negócio. O levantamento de requisitos compreende um estudo exploratório das necessidades do usuário. O documento de requisitos serve como um termo de consenso (inicial) entre a equipe técnica e o cliente. Abaixo o que deve conter no **DOCUMENTO DE REQUISITOS:**

* 1. Requisitos funcionais: definem as funcionalidades do sistema. Ex.:
     1. O sistema deve permitir o cadastro do cliente.. etc
  2. Requisitos não-funcionais: declaram as características de qualidade que o sistema deve possuir e que estão relacionadas às suas funcionalidade. Ex.:
     1. Confiabilidade: tempo médio de falha;
     2. Desempenho: tempo de resposta esperado para as funcionalidades;
     3. Portabilidade: restrições sobre hardware e software em que o sistema será implantado;
     4. Segurança: limitações sobre a segurança do sistema em relação a acessos não-autorizados;
  3. Restrições: restrições impostas sobre o desenvolvimento do sistema. Definem, por exemplo, a adequação a custos e prazos, plataforma tecnológica e aspectos legais (no nosso caso, sincronizar com emissor de cupom fiscal na venda), limitações sobre a interface com o usuário, componentes de hardwares e softwares a serem adquiridos (a empresa precisa comprar alguma coisa?);

Um ponto importante sobre o documento de requisitos é que ele não deve conter informações sobre *soluções técnicas*, mas sim responder à questão: “O QUE O USUÁRIO NECESSITA DO NOVO SISTEMA?”.

A principal função do levantamento de requisitos é compreender o sistema o máximo possível antes de começar a construí-lo. Ele será ordenado pelo usuário em função do seu grau de prioridade. Lembrando que os requisitos podem ser alterados a qualquer momento.

Por ser essa a parte mais importante, dedicaremos um tempo maior em sua análise.

1. **Casos de Uso**
2. **Análise de Requisitos**

Nesse momento os analistas realizam um estudo detalhado dos requisitos levantados anteriormente. É a partir dessa análise que são elaborados os **modelos** que vão representar o sistema a ser desenvolvido.

Ainda não se leva em consideração o ambiente tecnológico a ser utilizado. Estaremos definindo O QUE o sistema deve fazer... depois que analisamos COMO será feito.

Depois de prontos os modelos, faremos a VALIDAÇÃO (assegurar que as necessidades do cliente foram atendidas) e VERIFICAÇÃO (os modelos estão como os requisitos?).

Em um processo de desenvolvimento Orientado a Objeto, os resultados da análise são modelos que representam as estruturas das classes de objetos componentes do sistema. Além disso, a análise também resulta em modelos que especificam as funcionalidades do sistema de software.

Essa é a fase da PROTOTIPAGEM.

1. **Projeto**

Nesta fase será determinado COMO o sistema funcionará para atender os requisitos, de acordo com os recursos tecnológicos existentes. Neste momento identificamos as restrições cabíveis às análises de requisitos feita na etapa anterior: arquitetura do sistema, padrão de interface gráfica, linguagem de programação, gerenciador de banco de dados, etc. Ou seja, é produzida uma descrição computacional do que o software deve fazer e deve ser coerente com a descrição feita na análise.

O projeto consiste em duas atividades:

* **Projeto de arquitetura:** distribuição das classes. Nessa fase é utilizado o diagrama de UML de implementação. (verificar o que é).
* **Projeto detalhado:** são modeladas as colaborações entre os objetos de cada módulo com o objetivo de realizar as funcionalidades do módulo. Também são realizados o projeto da interface com o usuário e o projeto de banco de dados (acho que esse último não se aplica). Os diagramas da UML utilizados nessa fase do projeto são: diagrama de classes, diagrama de casos de uso, diagrama de interação, diagrama de estados e diagrama de atividades. (verificar)

**OBS.: as fases de projeto e análise muitas vezes se misturam!**

1. **Implementação**

É a parte de codificação. Não faremos.



Esse é aquele padrão para o projeto todo...



Esse seria o padrão SCRUM.. assim a UML vai crescendo com o tempo.. não sei se os modelos também.. podemos decidir.