

The background features a large, stylized number '5' in blue on the right side. Scattered around the page are numerous 3D cubes in various colors including purple, blue, yellow, green, orange, and pink. Some cubes are solid, while others are hollow. Faint, thin lines in blue, green, and yellow curve across the lower portion of the image.

Ambientes de desenvolvimento de software

- ✓ O que é um software;
- ✓ Tipos de software;
- ✓ Linguagens de programação;
- ✓ Bancos de dados;
- ✓ Tecnologias e ferramentas;
- ✓ Frameworks;
- ✓ Metodologias.

5.1. Introdução

Nesta leitura complementar, veremos como a Programação Orientada a Objetos se apresenta nos ambientes de desenvolvimento de software e quais as tecnologias e profissionais envolvidos nesse processo.

5.2. O que é um software

Software é um conjunto de instruções a serem executadas por um computador com o objetivo de resolver algum problema. Desenvolver um software é elaborar tais instruções de maneira lógica e ordenada, em um formato que o computador consiga interpretar.

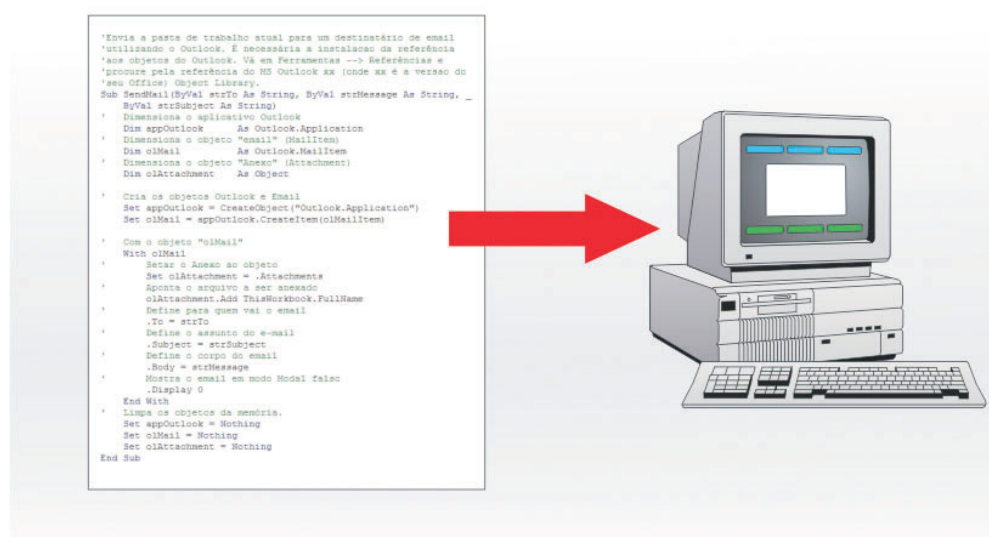


ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

A elaboração de um software passa por cinco fases: **Conceito**, **Design**, **Desenvolvimento**, **Testes** e **Implantação**. A seguir veremos o que acontece em cada uma dessas fases.

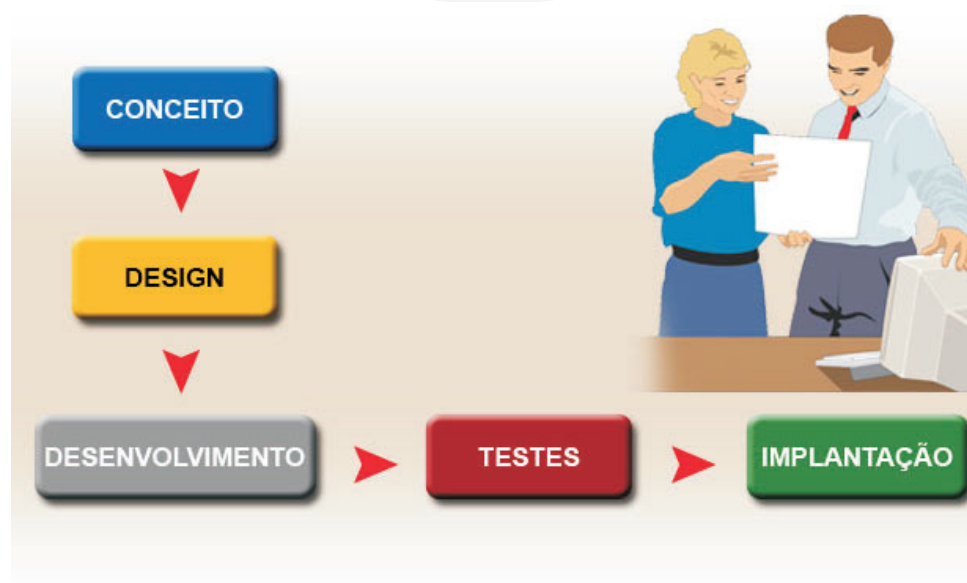


ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

- **Fase 1 – Conceito**

Nesta fase, o problema a ser resolvido é entendido pelos analistas para que seja possível construir a solução da melhor maneira possível. Esta é a fase mais importante do projeto, porque servirá de guia para todo o restante do processamento.

Normalmente, diversas reuniões são agendadas com os analistas para que o cliente explique detalhadamente a sua necessidade. O cliente é quem solicita o software, ou seja, é a pessoa ou empresa que tem uma necessidade específica.

- **Fase 2 – Design**

Nesta fase serão escolhidas as tecnologias que melhor atenderão às necessidades dos clientes, assim como a equipe necessária, o prazo e o custo do projeto. Normalmente, esta etapa é feita com os analistas trabalhando em conjunto com os programadores e técnicos. O resultado final desta fase são diversos diagramas em UML ou outra linguagem de especificação.

- **Fase 3 – Desenvolvimento**

Nesta fase entra o trabalho dos programadores, técnicos, especialistas em bancos de dados (dbas) e designers. É a construção do software, feita de acordo com as especificações definidas na fase 2. Normalmente, o processo é feito em etapas, com cronogramas e datas de entrega de partes funcionais do software.

- **Fase 4 – Testes**

A fase de testes consiste em verificar se as regras definidas na fase 2 atendem às necessidades definidas na fase 1. Muitas vezes, neste ponto o desenvolvimento retorna à fase 2 para redesenhar parte do projeto ou, às vezes, para a fase 1, para um melhor entendimento das regras do negócio.

Outra parte importante desta etapa é a estabilização do software. Cada funcionalidade é testada e pequenos ajustes são feitos até que fique totalmente funcional e livre de erros.

- **Fase 5 – Implantação**

Uma vez testado, o software pode ser implantado no ambiente de produção. Muitas vezes é necessário outro projeto apenas para esta fase, principalmente quando já existe um sistema e os dados necessitam de migração. Às vezes é necessário voltar às fases anteriores para ajustar algo que funcionou bem no ambiente de testes, mas que se mostrou ineficiente no ambiente de produção.

5.2.1. Tipos de software

Um software é composto por diversos tipos de projetos. Às vezes, um único projeto é suficiente e, às vezes, são necessários mais de 100 projetos para criar uma solução que atenda às necessidades de uma empresa.

5.2.1.1. Interface de usuário (User Interface ou UI)

A UI é a parte do software com a qual o usuário final interage. Quando uma pessoa utiliza um caixa eletrônico de um banco, por exemplo, os botões na tela e as mensagens são interfaces de usuário. Quando o usuário confirma uma transferência de saldo, não é a interface que faz a transação, mas outro componente com o qual a interface se comunica.

Existem diversas interfaces:

- Programas Windows, como Excel ou Word;
- Programas Web, como lojas on-line;
- Programas para celular;
- Programas para dispositivos portáteis;
- Programa para o surface da Microsoft.



ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

5.2.1.2. Componentes

Os componentes são a parte do software que faz algum processamento, mas que não interage com o usuário final diretamente. Por exemplo, ao confirmar uma transação em um caixa eletrônico, um componente deve guardar essa informação em um banco de dados. Mas esse componente pode ser, também, acionado pela Internet ou por um celular. Esse componente não está ligado a uma interface em particular, ele apenas recebe uma mensagem e realiza a tarefa de gravação.

Existem diversos tipos de componentes:

- Componentes para gravar em bancos de dados;
- Componentes distribuídos, para serem chamados de outro computador;
- Componentes contendo grupos de controles para criar interfaces de usuários.

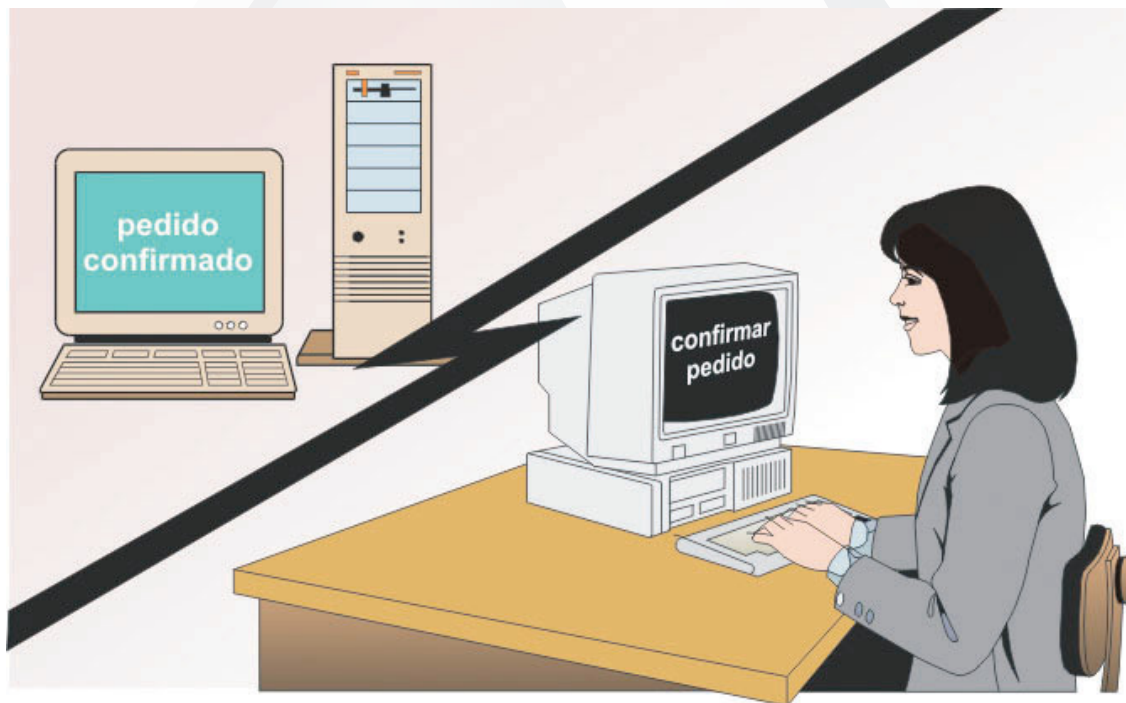


ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

5.2.1.3. Serviços

O Windows utiliza o conceito de serviços, que é um software ativado automaticamente quando o computador é ligado. Esse tipo de software é chamado de servidor. Por exemplo, os softwares antivírus são geralmente instalados como serviços no computador. Assim, o usuário não precisa se lembrar de chamar o programa toda vez que ligar o computador e, além disso, muitos vírus entram em ação antes de o usuário fazer o login na máquina.

5.2.1.4. Web services

Os Web services são componentes instalados em um servidor Web. A vantagem de um Web service é fazer com que um computador que não está na rede local possa chamar serviços de outro computador pela Internet. Em uma empresa, em que normalmente os firewalls bloqueiam qualquer acesso, exceto acesso à Internet, isso é claramente uma vantagem.



ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

5.3. Linguagens de programação

Uma das decisões que um analista deve tomar é a tecnologia que será utilizada para construir um software. Isso envolve escolher o sistema operacional, o ambiente de execução do software, a linguagem de programação, as ferramentas de desenvolvimento, o banco de dados, a metodologia de desenvolvimento e os profissionais necessários.

Há alguns anos, a única preocupação de um analista era a linguagem de programação e todo o resto dependia dessa escolha. A partir do ano 2000, essa abordagem foi mudando, pois cada vez mais uma linguagem pode utilizar componentes feitos por outra e até desenvolvidos em outro sistema operacional. A linguagem de programação passou a ser um dos muitos fatores a serem considerados.

Vejamos algumas linguagens existentes:

Clipper	T-SQL
COBOL	Ruby
Visual Basic	ActionScript
Java	Assembly
C++	Pascal
C#	Python
Visual Basic .NET	PHP
JavaScript	Self
VBScript	OCaml
J#	Fortran

Não existe uma resposta simples para a pergunta: “Qual linguagem escolher para iniciar um software?”. Qualquer linguagem pode ser usada para fazer um bom software, assim como qualquer linguagem pode ser usada para desenvolver um péssimo software. Não é a linguagem que determina o sucesso, mas a mente do profissional que concebeu e desenvolveu o software.

Vejamos algumas considerações sobre as linguagens de programação:

- A linguagem Java é muito difundida e conta com uma comunidade muito grande de programadores e colaboradores. Ela é gratuita, portanto, não há custo com ferramenta de desenvolvimento. O profissional da linguagem Java é um dos mais bem pagos do mercado e existe uma procura muito grande por bons programadores;
- A linguagem C# (C Sharp) é uma iniciativa da Microsoft e foi construída especialmente para a plataforma .NET, que é uma coleção de programas e ferramentas para desenvolver e executar softwares usando a Internet como ambiente. Existe uma grande procura por profissionais desta linguagem no mercado. A Microsoft disponibilizou ferramentas gratuitas para criar programas com esta linguagem, mas a principal ferramenta, o Visual Studio, é um programa pago;
- A linguagem PHP é muito utilizada para criar páginas Web, é de fácil aprendizado e largamente utilizada. As últimas versões incorporaram recursos de orientação a objetos. O mercado para PHP é menor do que para Java e C#, mas existe muita procura por profissionais, principalmente para dar suporte e desenvolver planos de migração para as linguagens mais atuais;
- A linguagem VBScript, utilizada no ambiente de programação para Web Active Server Pages (ASP), foi substituída pela linguagem Visual Basic .NET e pelo ambiente ASP.NET, mas ainda é utilizada em grande parte dos sites. A procura por profissionais é essencialmente para migração de dados;

- Ruby é uma linguagem desenvolvida especialmente para a Web e largamente difundida fora do Brasil. Existem diversas bibliotecas para esta linguagem, sendo mais famosa a chamada Ruby on Rails. Uma das características é a criação rápida de sites.

5.4. Bancos de dados

Uma parte fundamental de uma aplicação é como as informações são gravadas e como são recuperadas. A escolha de um banco de dados é fundamental para o sucesso do software.

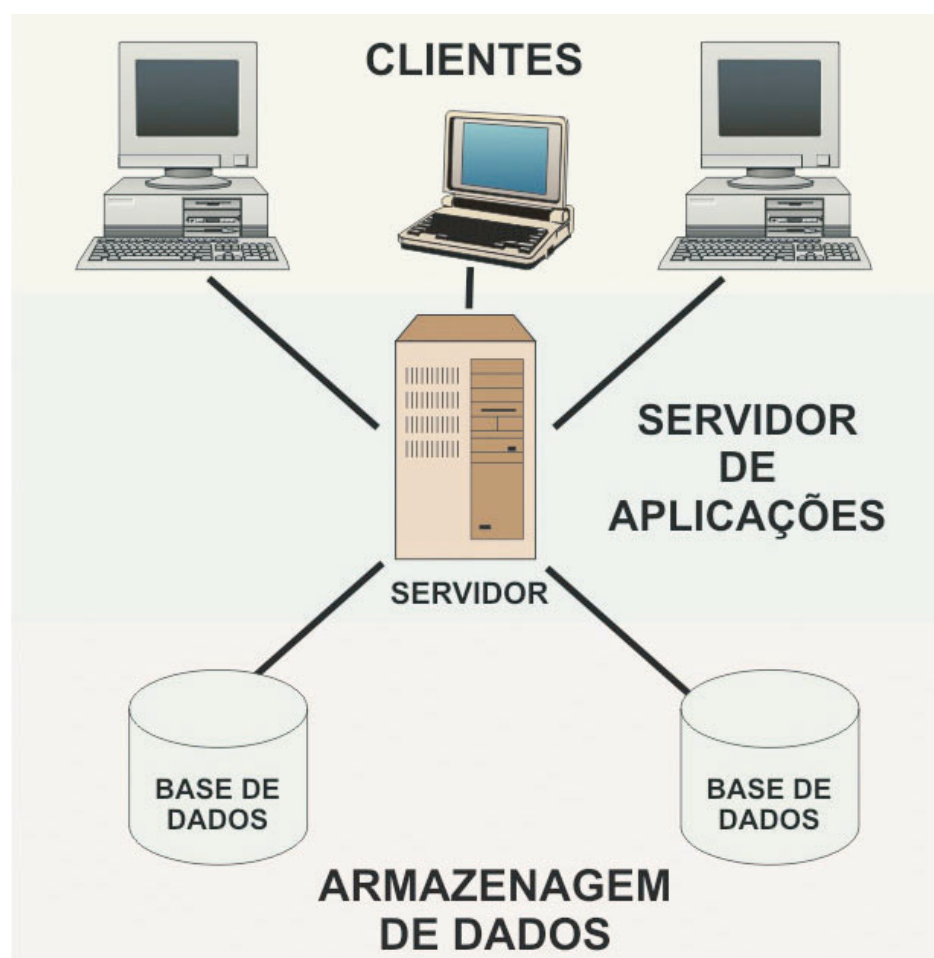


ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

Existem dois tipos de bancos de dados: de servidor e de arquivo.

- **Banco de dados de arquivo**

Neste tipo de banco de dados, as informações são gravadas diretamente em um ou mais arquivos, e estes são abertos diretamente pelo sistema operacional. Este tipo de banco é ideal para aplicações isoladas, que não utilizam a rede. Ao compartilhar o arquivo em rede, corre-se o risco de corromper o arquivo caso haja uma falta de energia no momento de uma gravação ou mesmo se houver um problema com a placa de rede.

Podemos citar como exemplos de bancos de dados de arquivo: Access, DBF, Excel, Paradox.

- **Banco de dados de servidor**

Neste tipo de banco de dados, um serviço instalado em um computador (chamado servidor) acessa os dados e cuida da integridade dos mesmos. Os aplicativos e usuários acessam o servidor e não os arquivos diretamente. Este tipo de banco de dados é ideal para aplicações multiusuários e para grande volume de dados.

Podemos citar como exemplos de banco de dados de servidor: SQL Server, Oracle, MySQL, PostgreSQL, DB2, Firebird.

A linguagem padrão para o acesso a um banco de dados é a SQL (Structured Query Language). Cada banco pode introduzir variações e funções específicas, mas o padrão é praticamente o mesmo para todos os bancos.

5.5. Tecnologias e ferramentas

Existem muitas ferramentas disponíveis para ajudar um profissional a desenvolver qualquer parte de um software. As ferramentas que reúnem em um único lugar todos os recursos necessários para criar um software são chamadas IDE ou Ambiente de Desenvolvimento Integrado.

5.5.1. Java

O ambiente Java conta com aplicativos que facilitam o design de aplicações. As ferramentas mais conhecidas são:

- NetBeans;
- Eclipse;
- jEdit;
- JBuilder;
- JDeveloper.

5.5.2. Plataforma .NET

A plataforma .NET (Linguagens VB.NET e C#) conta com muitas ferramentas para criar softwares desde a concepção até a implantação final. Algumas dessas ferramentas são:

- Visual Studio;
- Web Developer;
- C# Express;
- VB Express.

5.6. Frameworks

Frameworks são conjuntos de códigos compilados prontos para serem usados em um aplicativo. O uso de um framework pode acelerar o processo de desenvolvimento de um software. A decisão de usar ou não um framework é geralmente polêmica dentro de um ambiente corporativo. De um lado, o framework pode tornar mais ágil o desenvolvimento, mas por outro pode limitar seriamente a construção de algo novo.

Alguns frameworks famosos são:

- **Hibernate**

O Hibernate é um framework que pode ser usado tanto em Java quanto em .NET (C#/VB). O objetivo é transformar os objetos em instruções SQL para automatizar o processo de gravação e leitura de objetos. Esse tipo de operação é chamada de Mapeamento Objeto-Relacional.

- **AJAX.NET**

Ajax.NET é uma biblioteca para utilizar Ajax em aplicações Web. O Ajax é uma tecnologia que combina JavaScript com XML para fazer as páginas Web responderem mais rapidamente.

- **Struts**

Struts é um framework para realizar o controle de operações com objetos com validação, gravação e recuperação de dados. O Struts foi desenvolvido em Java.



ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

5.7. Metodologias de desenvolvimento

Construção de software é uma ciência recente e em fase de estruturação. Várias metodologias de desenvolvimento têm aparecido e desaparecido com o passar dos anos. Na verdade, cada empresa acaba criando uma metodologia própria de desenvolvimento com o passar do tempo, baseada em suas necessidades, disponibilidade de mão de obra, prazos, custos e casos de sucesso e fracasso.

Algumas metodologias famosas são:

- **Programação estruturada**

É uma forma de programa em sequência, priorizando as ações a serem tomadas. Muito popular nos anos 1960, foi sucedida pela Programação orientada a objetos.



ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

- **RUP**

O Rational Unified Process é uma metodologia orientada a objetos cujo ponto principal é a notação UML para definir os procedimentos. É geralmente usado para grandes projetos, quando existe um analista ou uma equipe de analistas que determina minuciosamente o que deve ser feito.

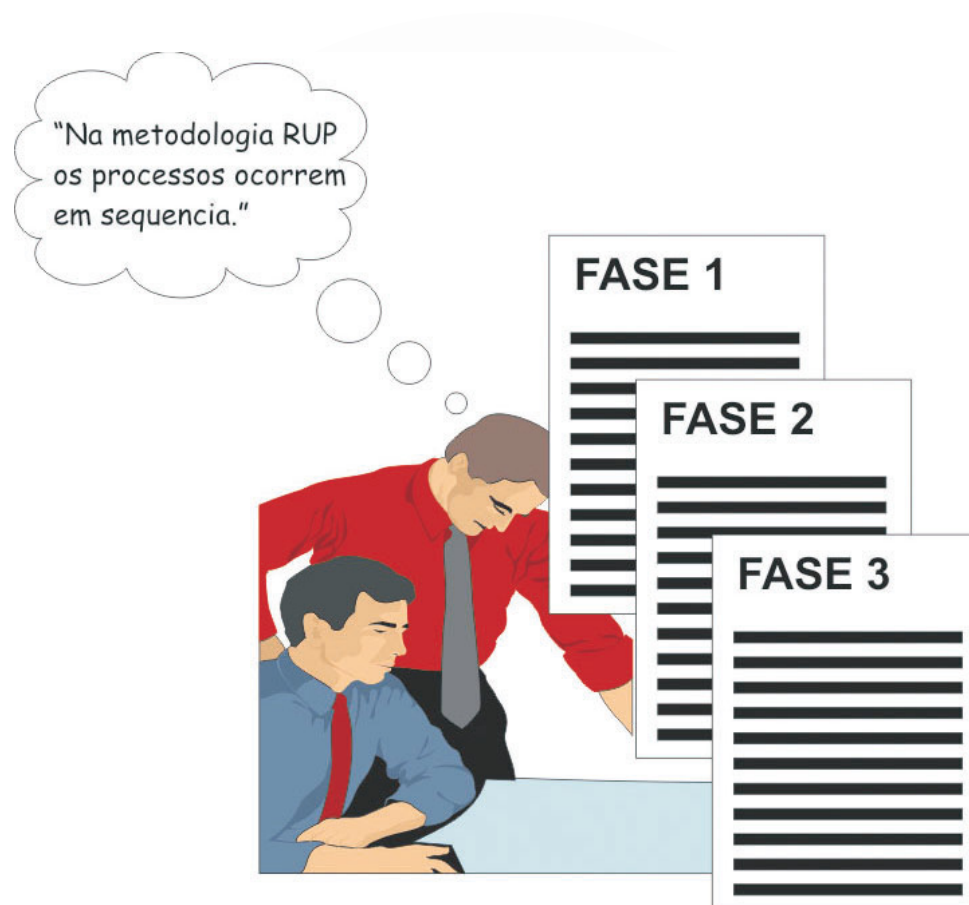


ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

- **XP**

O Extreme Programming é uma metodologia geralmente usada por pequenas equipes de programadores. O ponto principal é a interação da equipe com o cliente e a entrega semanal de unidades de softwares prontas para uso. O XP incorpora nas suas premissas que o objetivo do software é mutável e o cliente pode mudar de ideia no meio do processo. As mudanças são bem-vindas e fazem parte do processo de criação.



ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

- **Scrum**

O método Scrum de desenvolvimento é utilizado em algumas empresas de construção de veículos no Japão e consiste na entrega de pequenos programas funcionais em intervalos regulares e reuniões diárias sobre o andamento.

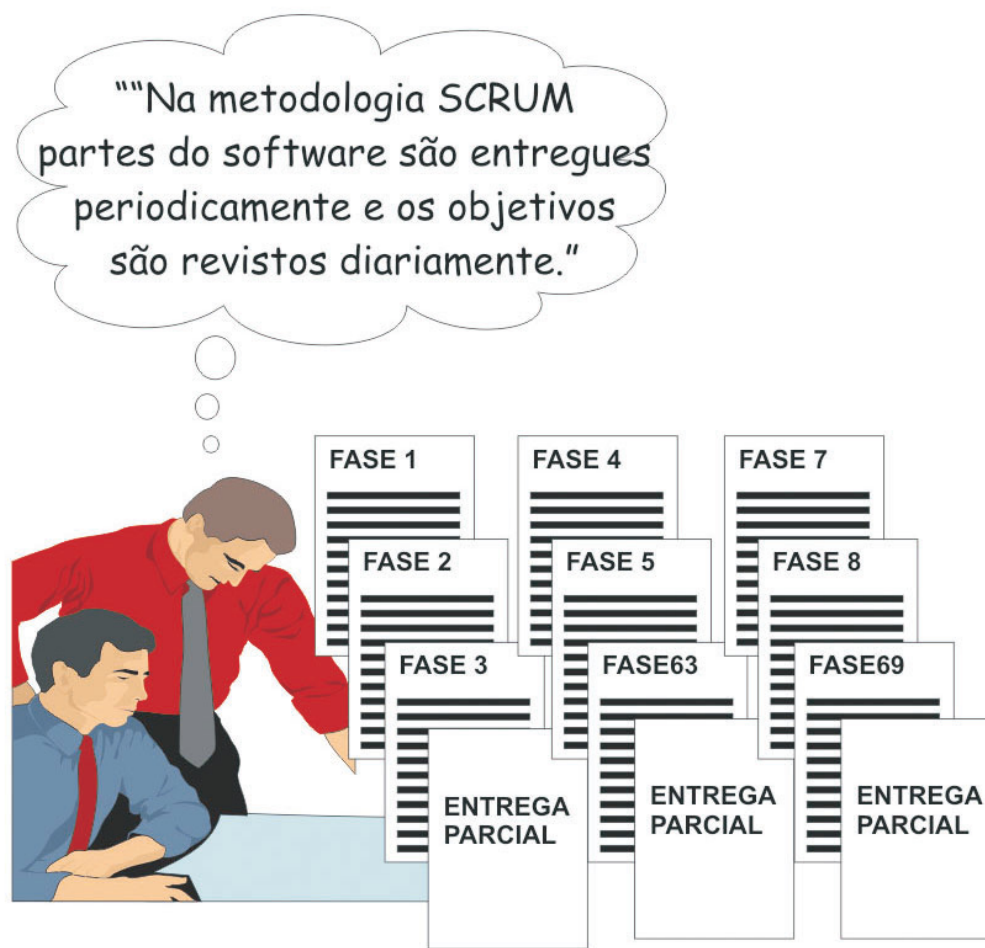


ILUSTRAÇÃO: EDUARDO JOSÉ DE SOUZA ENGELMANN

5.8. Resumo

Para desenvolver um software com sucesso, é necessário analisar vários aspectos que fazem parte desse ambiente. Escolher a linguagem, a tecnologia, a equipe e a metodologia corretas é um ponto crucial para o sucesso.

A seguir, temos um exemplo de um ambiente de desenvolvimento:

PROJETO LOJA ON LINE

1. Ambiente: ASP.NET
2. Linguagem: C#
3. Banco de dados: SQL Server
4. Ferramentas: Visual Studio, Photoshop e Flash
5. Framework: AJAX.NET
6. Profissionais: 1 analista, 2 programadores, 1 Web designer

PROJETO ORCAMENTO

1. Ambiente: Linux com ambiente gráfico
2. Linguagem: Java
3. Banco de dados: MySQL
4. Ferramentas: NetBeans
5. Profissionais: 1 programador