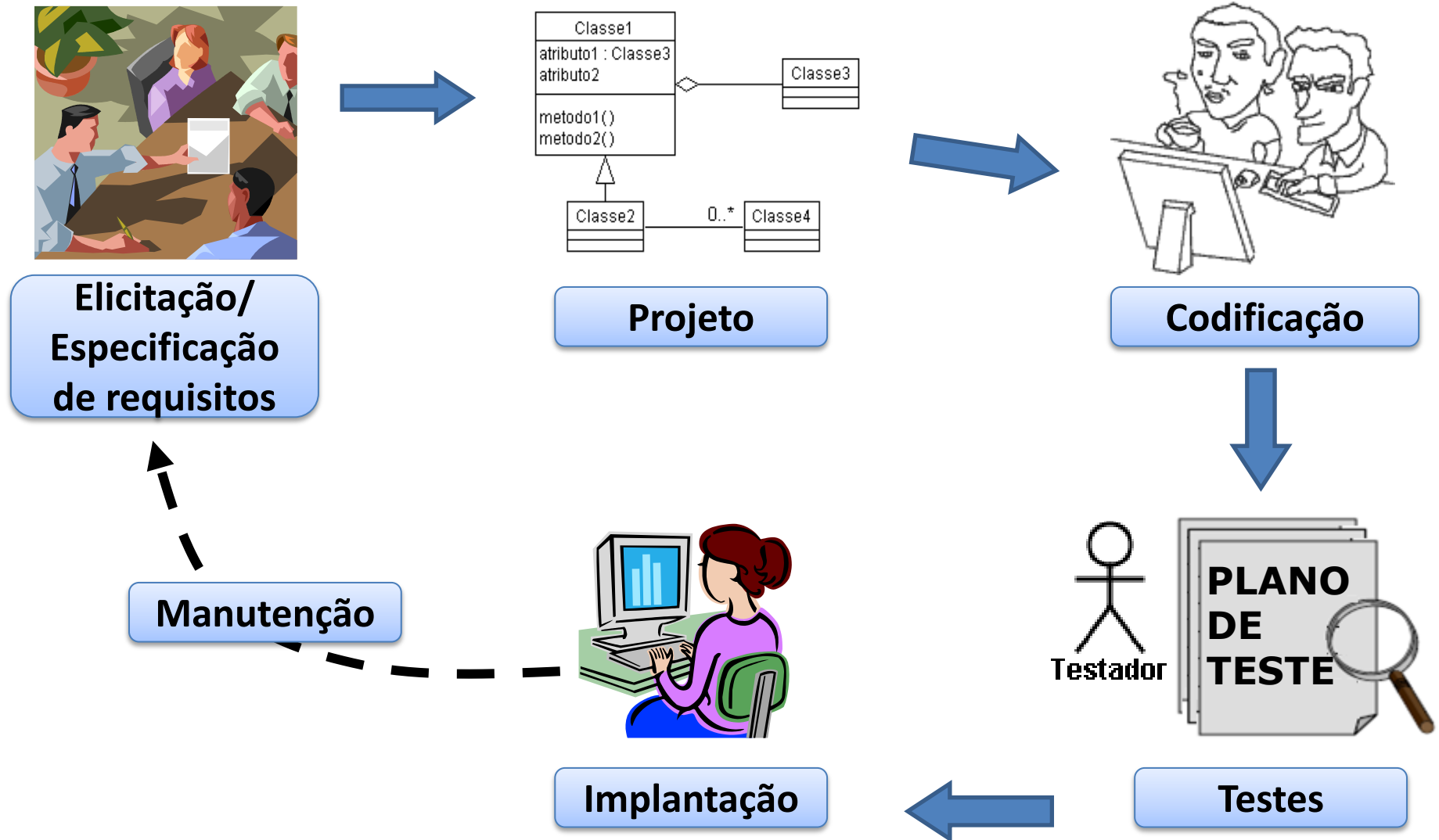


ENGENHARIA DE SOFTWARE I

Ciclo de vida de software

Fases de um projeto de software

→ Ocorrem sempre independente do modelo de processo de software (cascata, iterativo, incremental, etc.)

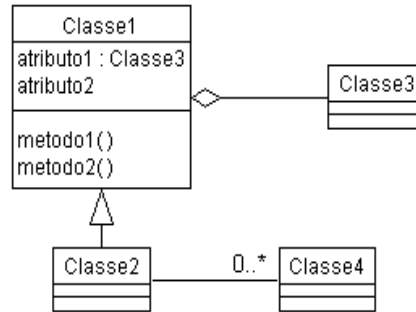


Fases de um projeto de software

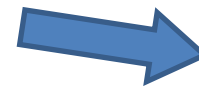
Exemplo: Sistema para gestão de universidades



**Elicitação/
Especificação
de requisitos**



Projeto



Codificação



Testes



Implantação

Manutenção



Ciclo de vida de software

Elicitação / Especificação dos requisitos – também chamado de Análise de requisitos:

- Ocorre antes do início da construção de um sistema
- Deve ser definido quais as funcionalidades deverão ter
- Quais problemas visa resolver para aquele grupo de usuários ou quais oportunidades que aquele sistema irá proporcionar
- Como ele será usado e como será sua interação com os usuários
- **Esta visão externa de seu funcionamento** pode ser obtida através da Elicitação de Requisitos e a definição das funcionalidade é realizada através da Especificação de requisitos.

Problema dos requisitos de software



Como o cliente explicou



Como o lider de projeto entendeu



Como o analista planejou



Como o programador codificou



O que os beta testers receberam



Como o consultor de negocios descreveu



Valor que o cliente pagou



Como o projeto foi documentado



O que a assistencia tecnica instalou



Como foi suportado

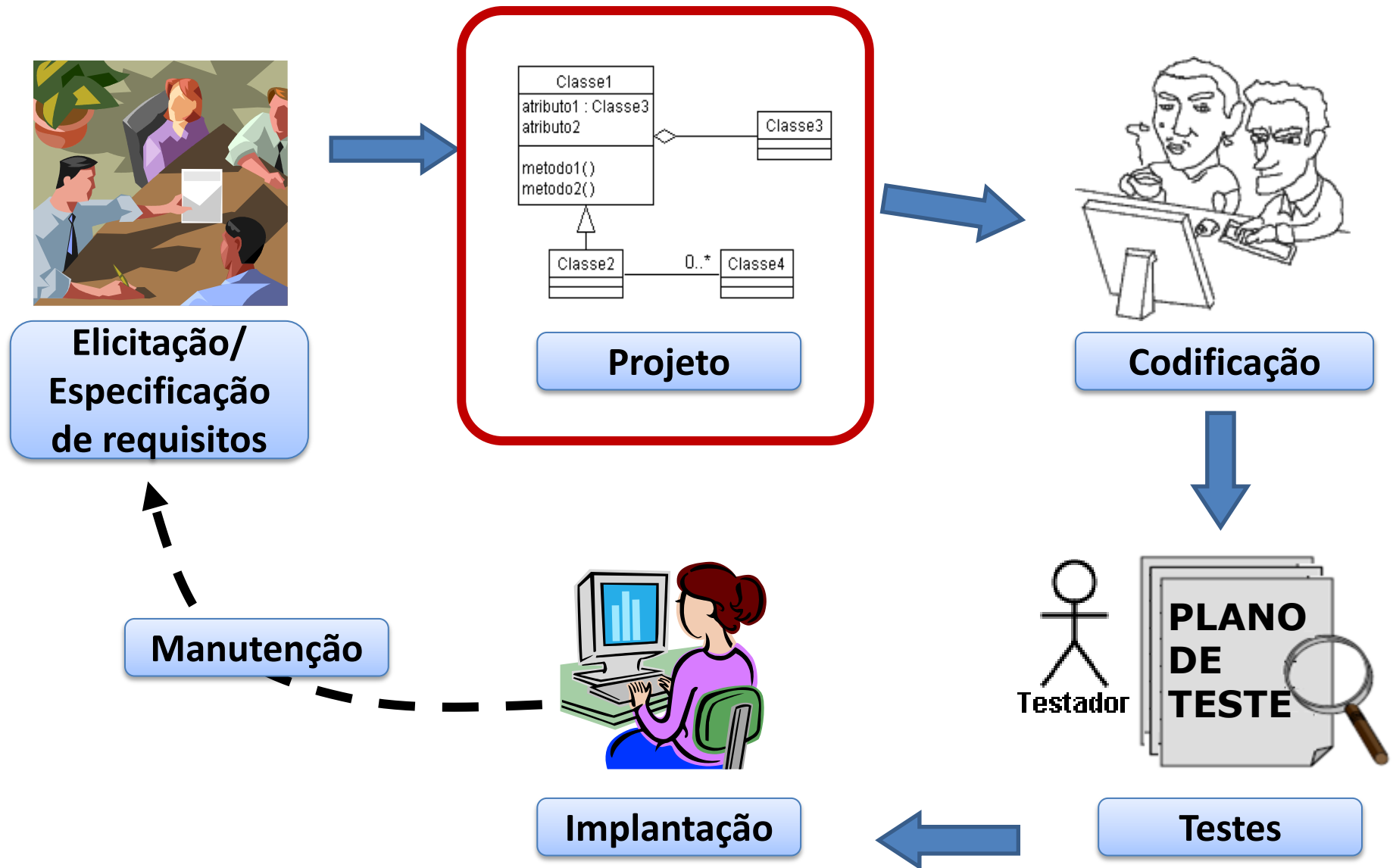


Quando foi entregue



O que o cliente realmente necessitava

Fases de um projeto de software



Ciclo de vida de software

Projeto:

- A **Análise** geralmente transcorre com a suposição de que há uma tecnologia “perfeita” disponível;
- No **Projeto**, sabe-se que o sistema será implementado em uma plataforma de hardware, sob um sistema operacional, usando uma linguagem de programação.
- Ou seja, a **Análise** se interessa pelo que o sistema deve fazer, enquanto o **Projeto** diz respeito a como os requisitos serão implementados.

Ciclo de vida de software

Fase de Projeto



Ciclo de vida de software

Projeto:

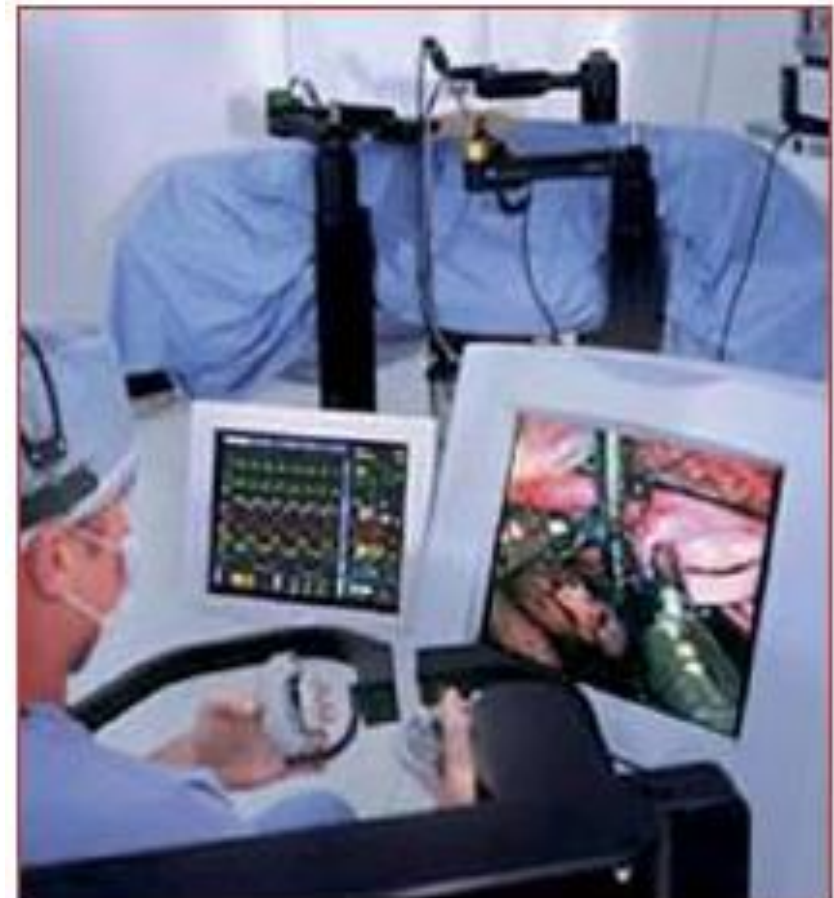
- Na fase de Projeto há a incorporação de requisitos tecnológicos aos componentes modelados na fase de Análise, bem como a definição da plataforma e das ferramentas utilizadas.
- O Projeto é dependente de aspectos como as características da linguagem de programação utilizada, das características da plataforma de implementação e das características da interface com o usuário.

Rede Social



The image shows the Facebook login and registration interface. At the top, there's a blue header with the Facebook logo and login fields for email/phone and password. Below this, a section titled "Abra uma conta" (Create an account) is visible. It includes a subtext "É gratuito e sempre será." (It's free and always will be). The registration form contains fields for "Nome" (Name) and "Sobrenome" (Surname), "Seu e-mail" (Your email), "Insira o e-mail novamente" (Re-enter your email), and "Nova senha" (New password). There are also dropdown menus for "Data de nascimento" (Date of birth) with options for Day, Month, and Year, and radio buttons for "Feminino" (Female) and "Masculino" (Male). A small disclaimer at the bottom states: "Ao clicar em 'Abrir uma conta', você concorda com nossos Termos e que você leu nossa Política de uso de dados, incluindo nossa Use de cookies."

Cirurgia computadorizada



Comércio eletrônico



The screenshot shows the Amazon website interface. At the top, the Amazon logo and 'Try Prime' button are visible. Below the navigation bar, the 'Computer Monitors' category is selected. The main content area displays 'Monitors' with a sub-header 'Shop for monitors from top brands including'. A 'Best Sellers' section features a monitor displaying a child's face. On the left, there are filters for 'Show results for' (Electronics, Computers & Accessories, Monitors) and 'Refine by' (International Shipping, Display Size).

amazon
Try Prime

Computer Monitors

Departments

Your Amazon.com Today's Deals

Computers Laptops Tablets Desktops Monitors Computer

Show results for

- Electronics
- Computers & Accessories
- Monitors

Refine by

International Shipping

☐ Ship to Brazil

Display Size

- ☐ 50 Inches & Above (415)
- ☐ 40 to 49.9 Inches (219)

Monitors

Shop for monitors from top brands including

Best Sellers



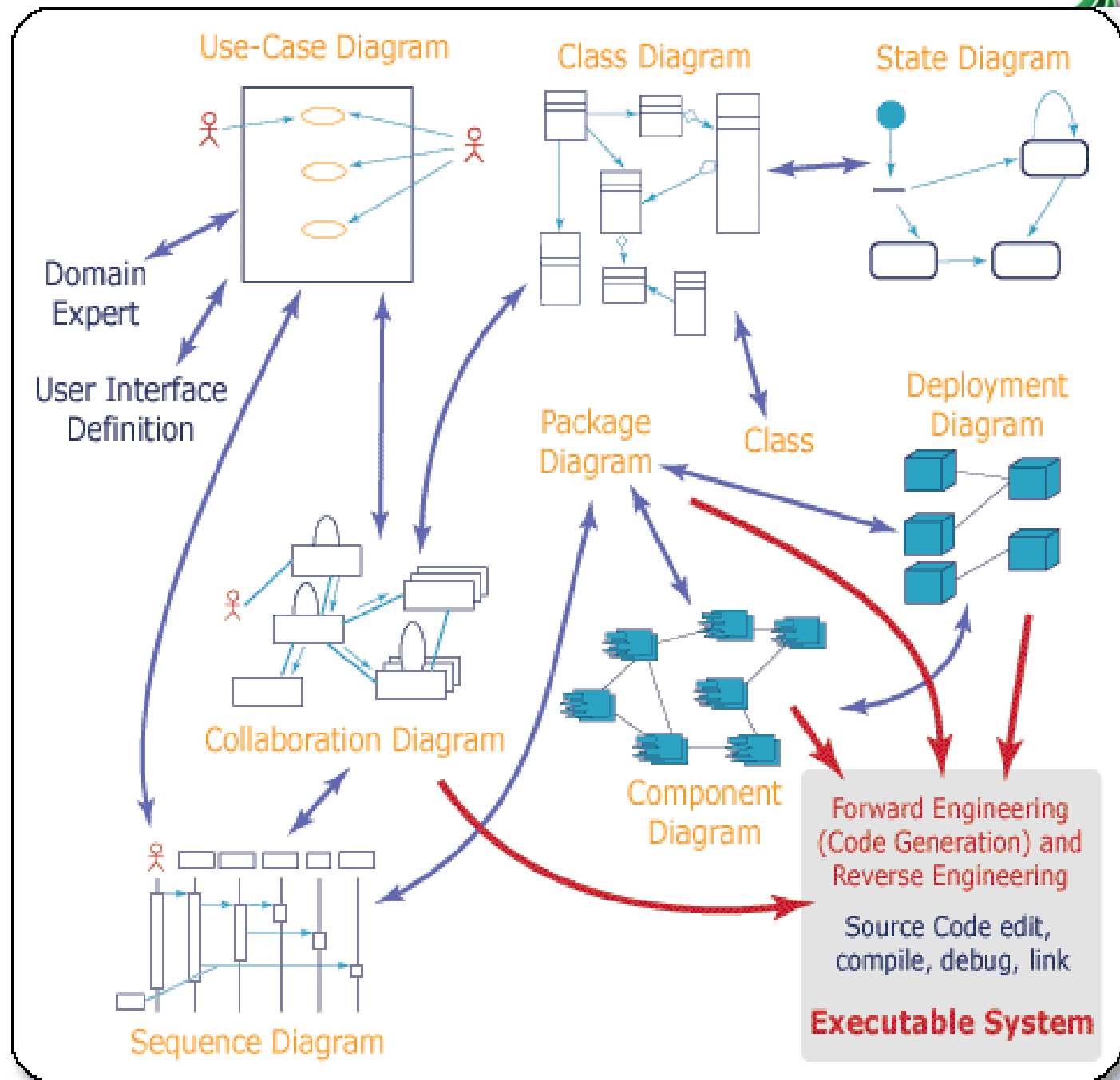
Robô em Marte



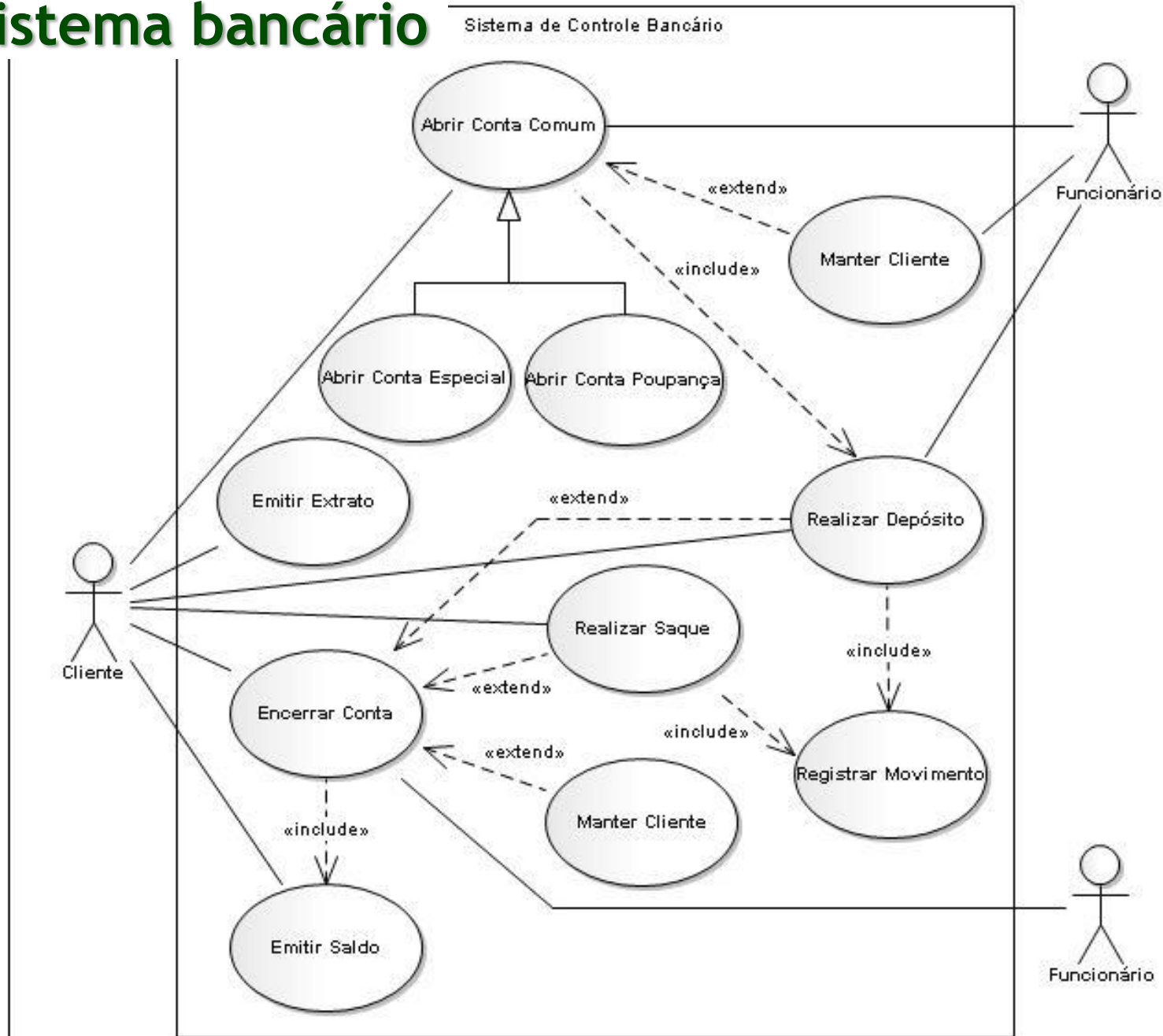
Diagramas da UML



Conteúdo da
disciplina de
**Engenharia de
Software II**



Exemplo: Sistema bancário



Exemplo sistema bancário

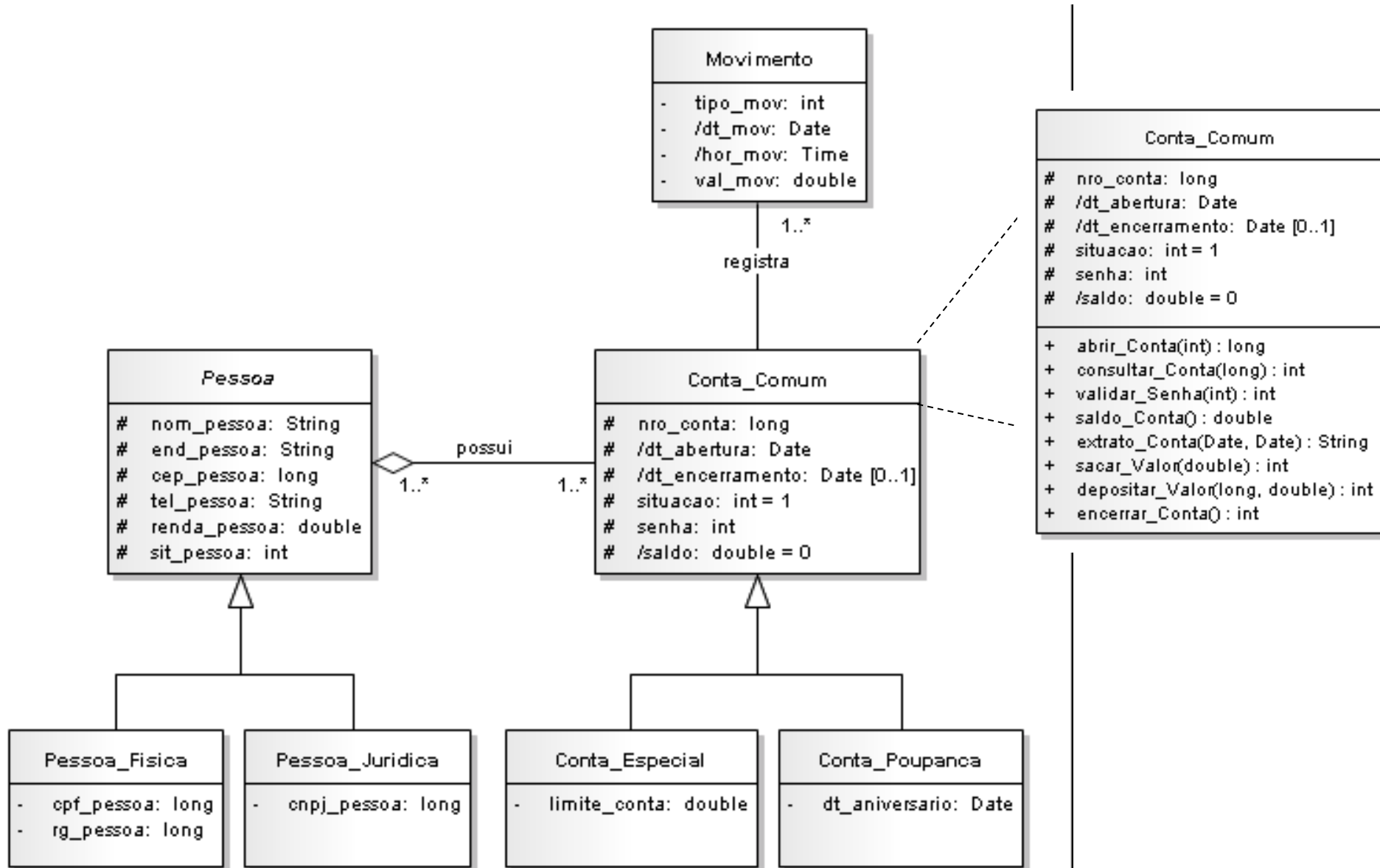


Diagrama de pacotes

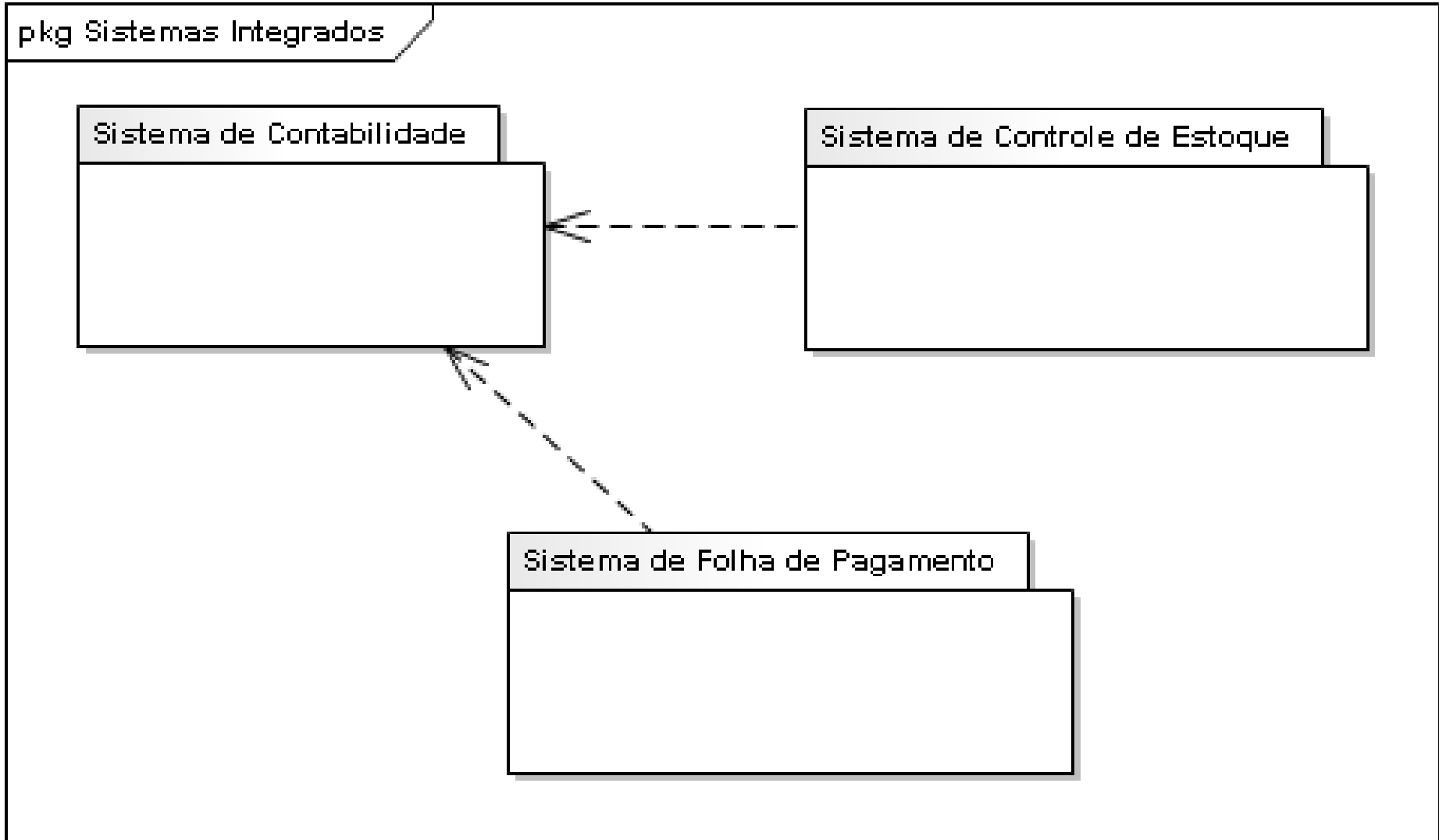


Diagrama de sequencia: Abrir conta

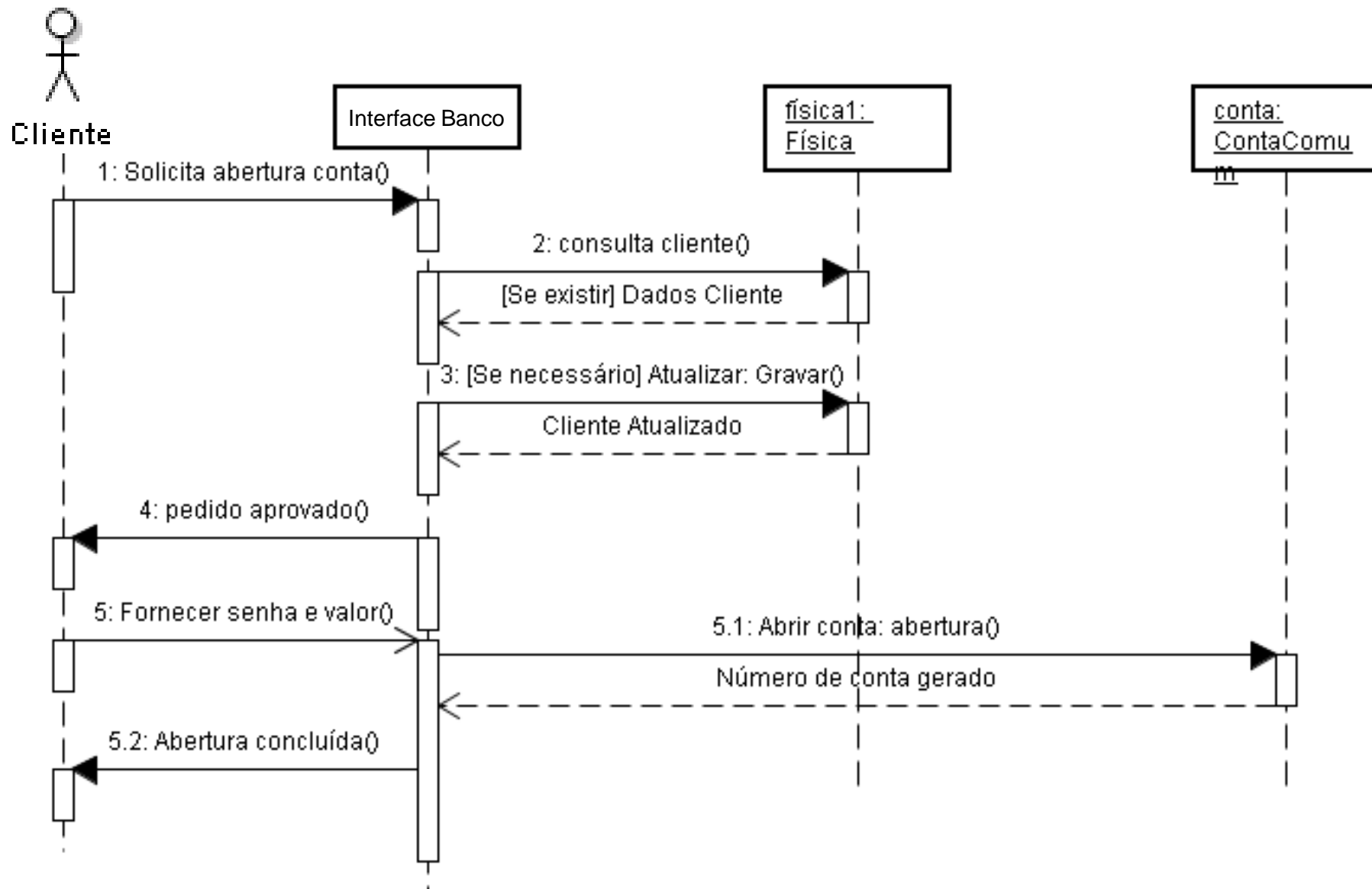
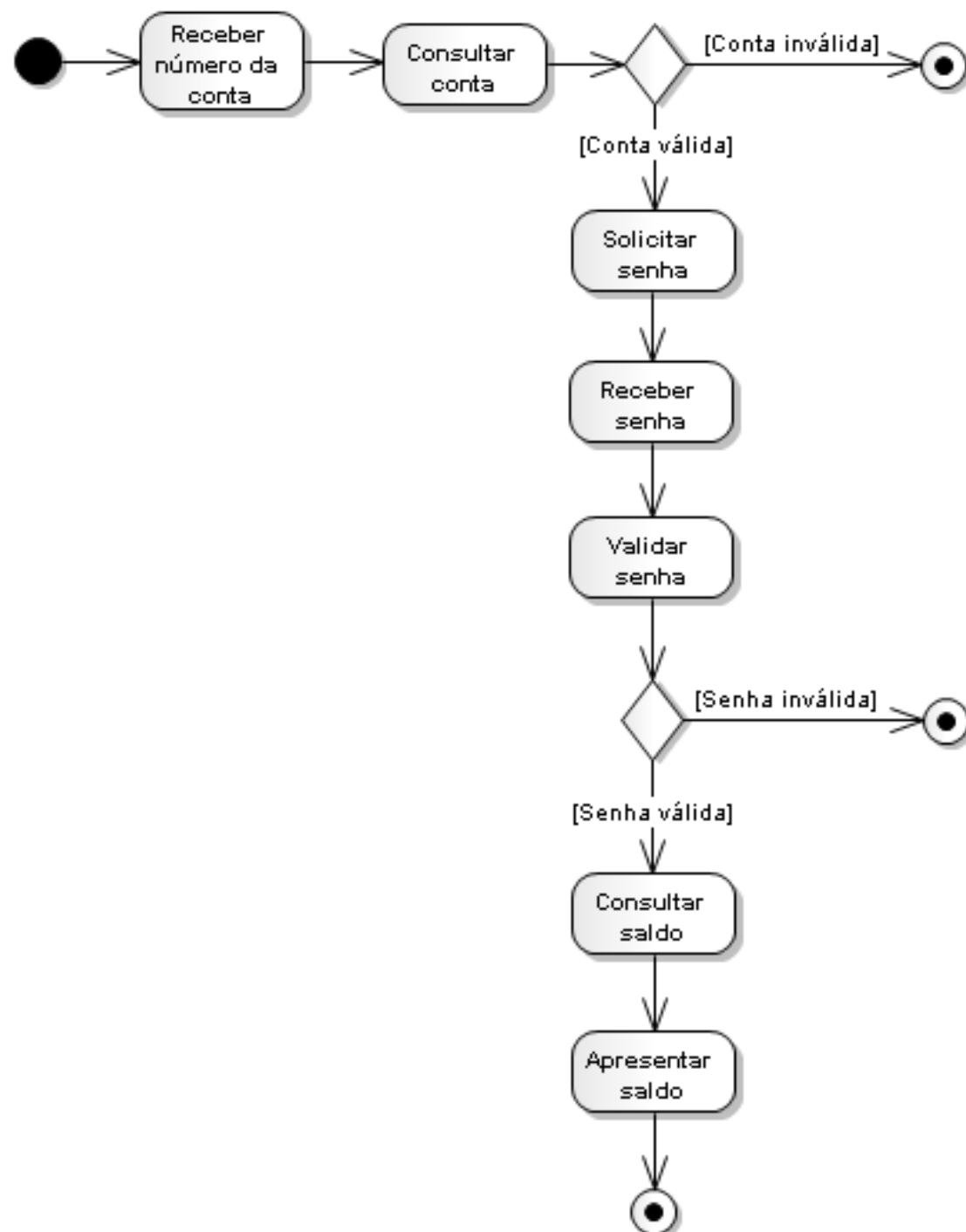
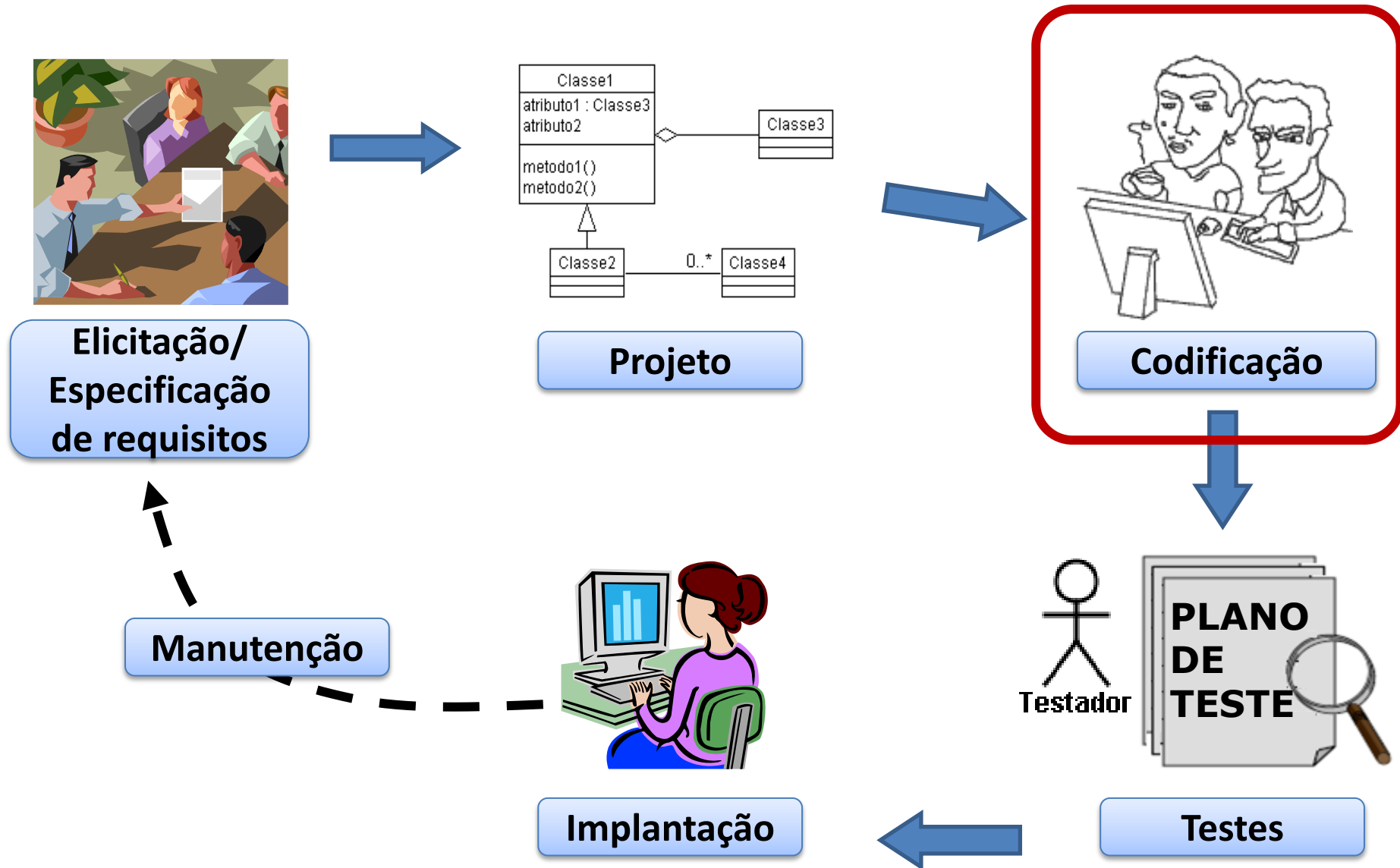


Diagrama de atividades: Emitir sado



UML → conteúdo da disciplina de Engenharia de Software II

Fases de um projeto de software

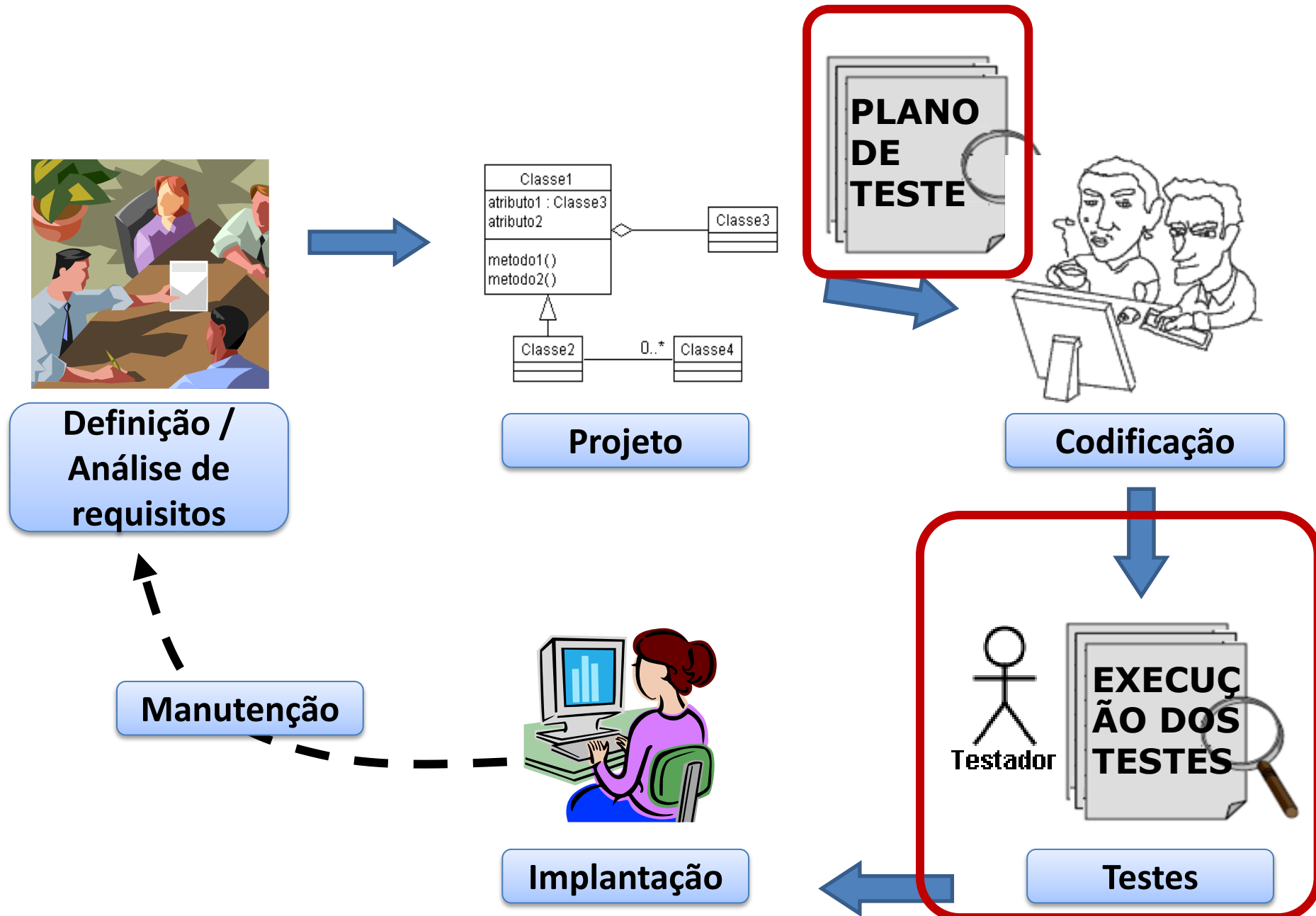


Ciclo de vida de software

Codificação:

- A Codificação deve ser vista como uma **extensão ao processo de projetar**.
- Deve ser direta, quase mecânica, uma vez que as **decisões difíceis devem ter sido tomadas durante o projeto**.
- A Codificação deve ser uma tradução das decisões de projeto em uma linguagem específica.

Fases de um projeto de software

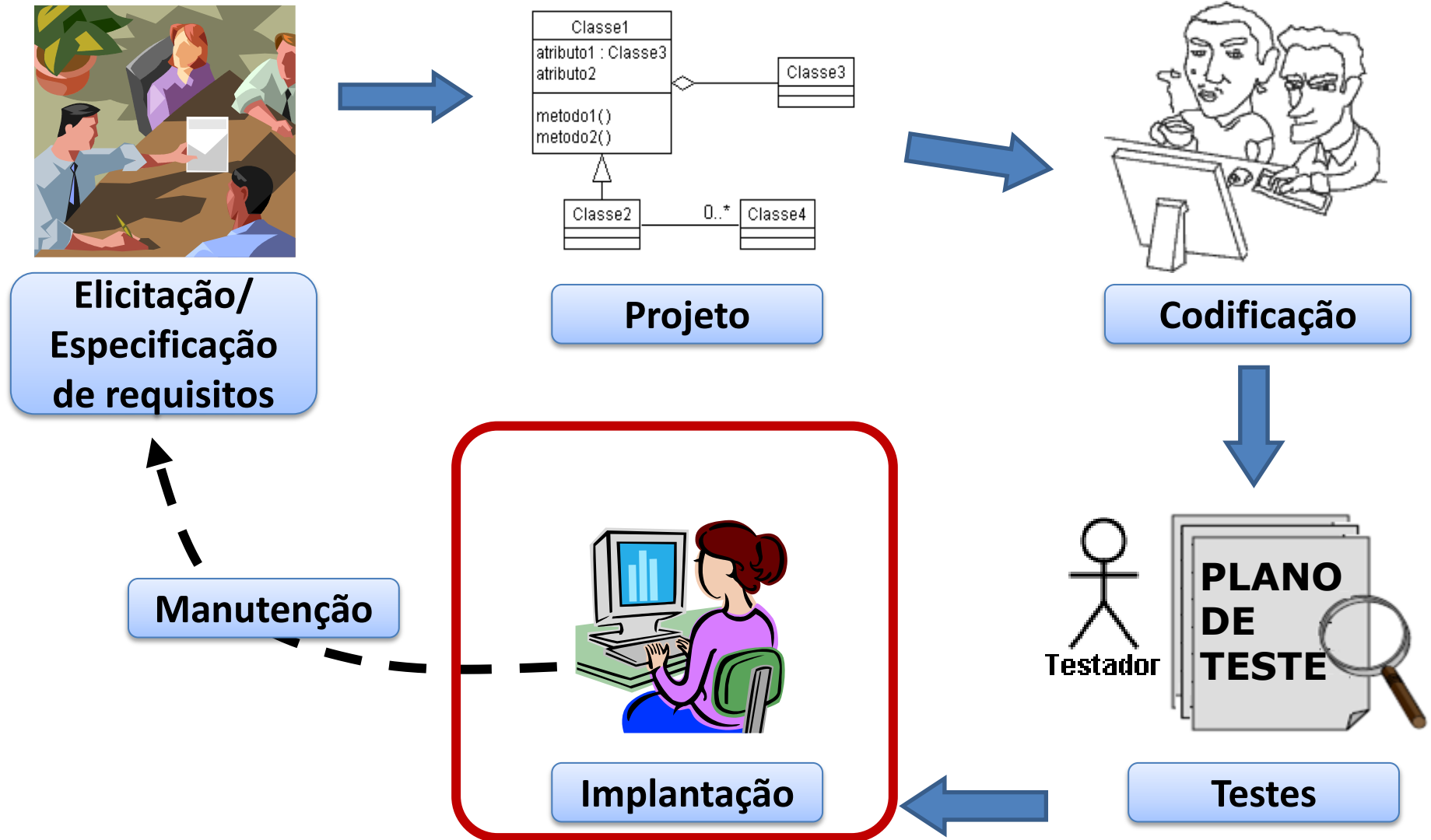


Ciclo de vida de software

Testes:

- Teste de software é uma atividade de garantia da qualidade.
- O principal objetivo é analisar a qualidade do software em execução, verificando se este atende às necessidades do cliente.
- Os principais tipos de teste são: teste de unidade, teste de integração, teste de sistema e teste de aceitação.

Fases de um projeto de software



Ciclo de vida de software

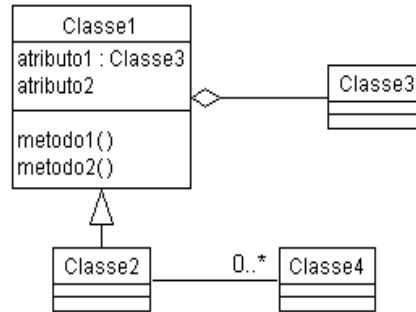
Implantação → Suporte e treinamento:

- Utilização por parte do usuário
- As pessoas são resistentes à mudança.
- Então, como uma parte da fase de desenvolvimento, é muito importante o treinamento e suporte aos usuários após a implantação do sistema.

Fases de um projeto de software



**Elicitação/
Especificação
de requisitos**



Projeto



Codificação



Testes



Implantação

Manutenção



Ciclo de vida de software

Manutenção:

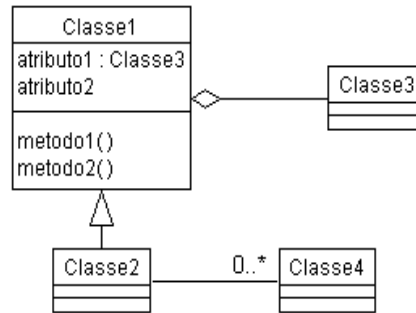
- A manutenção e melhoria de SW trata com a descoberta de novos problemas e requisitos.
- Pode tomar mais tempo que o gasto no desenvolvimento inicial do software.
- Requer um significativo esforço por parte de um Engenheiro de Software.
- A maior parte da manutenção é para ampliar os sistemas para novas funcionalidades, o que pode ser considerado um novo trabalho.



Fases de um projeto de software



**Elicitação/
Especificação
de requisitos**



Projeto



Codificação



Testes



Implantação

Manutenção



GERÊNCIA DE PROJETOS



Modelos de processo de software

Leitura complementar

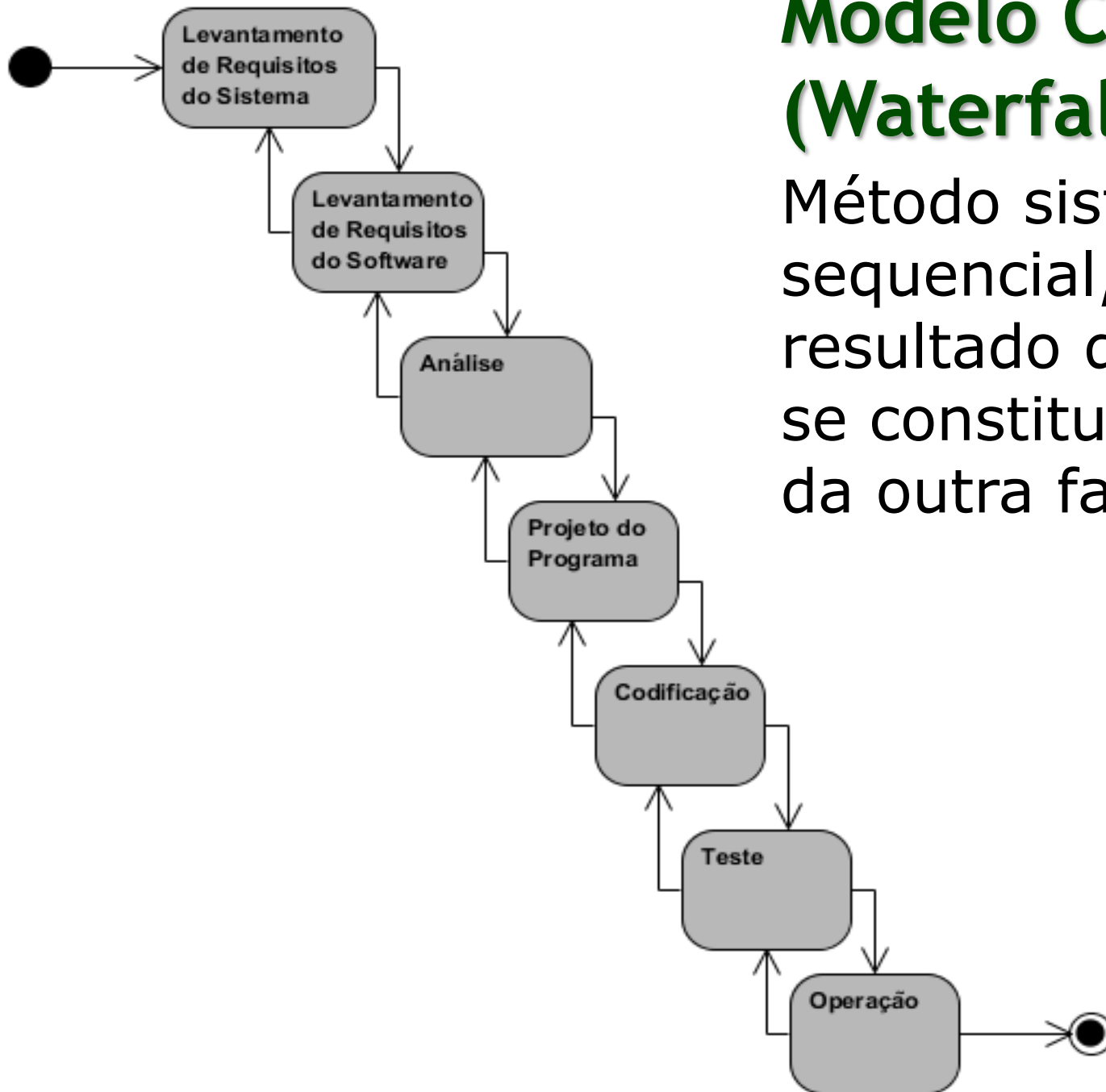
- Livro: Engenharia de Software: uma abordagem profissional - 7ª edição (2011)
Autor: Pressman
- 2.3.1. O modelo cascata
- 2.3.2. Modelos de processo incremental
- 2.3.3. Modelos de processo evolucionário (modelo espiral)

Principais modelos

- Modelo Cascata
- Modelo Incremental
- Modelo Espiral
- existem outros, mas estes são os mais utilizados

Modelo Cascata (Waterfall)

Método sistemático e sequencial, em que o resultado de uma fase se constitui na entrada da outra fase.



Modelo cascata (Waterfall)

- Modelo mais antigo e o mais amplamente usado na engenharia de software, modelado em função do ciclo da engenharia convencional.
- Modelo dirigido a planos. Fases de especificação e desenvolvimento separadas e distintas.

Modelo Cascata

Problemas do modelo Cascata:

- dificuldade de acomodação de mudanças depois que o processo já foi iniciado
 - Por isso esse modelo só é **apropriado quando os requisitos são bem entendidos** e as mudanças durante o processo de projeto serão limitadas
 - Poucos sistemas de negócio possuem requisitos estáveis.

Modelo Cascata

Problemas do modelo Cascata:

- Requisitos devem ser estabelecidos de maneira completa correta e clara no início de um projeto
- Difícil avaliar o progresso verdadeiro do projeto durante as primeiras fases
- Uma versão executável do software só fica disponível numa etapa avançada do desenvolvimento
- Ao final do projeto, é necessário um grande esforço de integração e testes
- Gera grande quantidade de documentação

Modelo Cascata

- Este modelo tem sido muito criticado pelos defensores dos métodos ágeis

Modelo incremental e iterativo

Delivery 1



Delivery 2



Delivery 3



Incremental plan



Iterative plan

Modelo incremental e iterativo

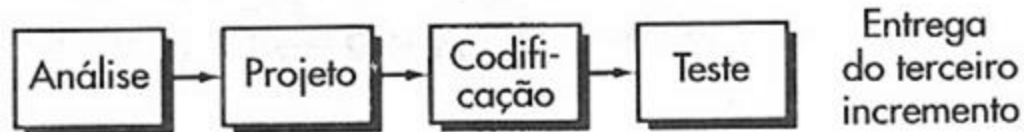
Incremento 1



Incremento 2



Incremento 3



Incremento 4



Modelo Incremental

- Ao invés de entregar o sistema em uma única entrega, o desenvolvimento e a entrega são distribuídos em incrementos, nos quais cada incremento entrega parte da funcionalidade necessária.
- O processo se repete até que um produto completo seja produzido.
- Os requisitos do usuário são priorizados e os requisitos de mais alta prioridade são incluídos nos primeiros incrementos
- Assim que o desenvolvimento de um incremento é iniciado os requisitos são congelados, mas os requisitos dos incrementos posteriores podem continuar a evoluir
- Necessidade de entrega de um produto funcional em pouco tempo
- A cada incremento é produzida uma versão operacional do software.
- Abordagem normalmente usada em métodos ágeis

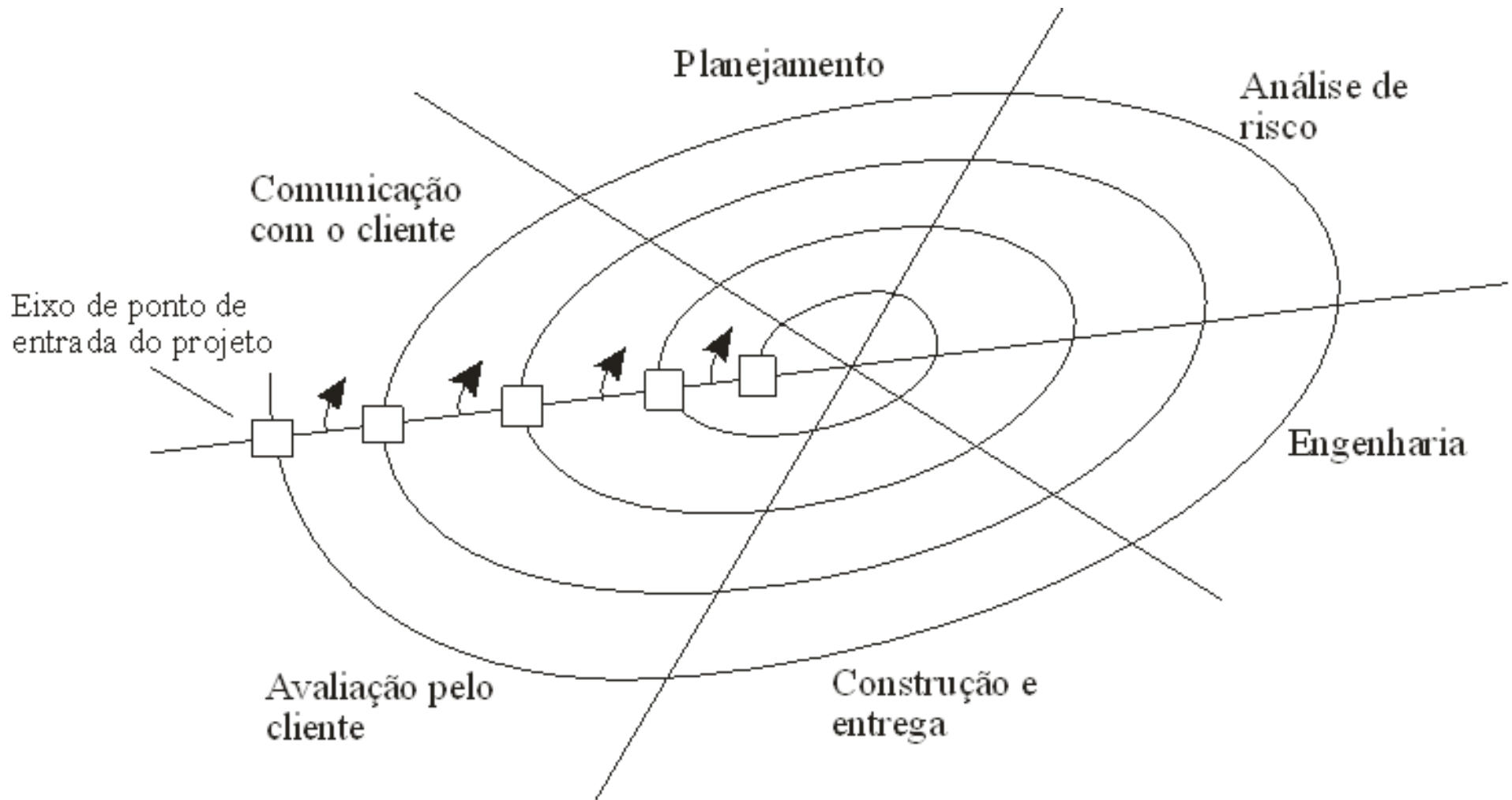
Benefícios do desenvolvimento incremental

- **O custo para acomodar mudanças nos requisitos do cliente é reduzido:**
 - A quantidade de análise e documentação que precisa ser feita é bem menor do que a necessária no modelo cascata.
- **É mais fácil obter feedback do cliente sobre o trabalho de desenvolvimento que tem sido feito:**
 - Os clientes podem comentar demonstrações do software e ver o quanto foi implementado.
- **Possibilidade de mais rapidez na entrega e implantação de software útil para o cliente:**
 - Os clientes podem usar e obter ganhos do software mais cedo do que é possível no processo cascata.
- **Menor risco de falha geral do projeto**

Problemas do desenvolvimento incremental

- **A maioria dos sistemas requer um conjunto de funções básicas que são usadas por diferentes partes do sistema:**
 - Como os requisitos não são definidos em detalhes até que um incremento seja implementado, pode ser difícil identificar funções comuns que são necessárias a todos os incrementos.
- **A estrutura do sistema tende a degradar conforme novos incrementos são adicionados:**
 - A menos que tempo e dinheiro sejam gastos na reconstrução para melhorar o software, as mudanças regulares tendem a corromper a estrutura do sistema. A incorporação posterior de mudanças no software se torna progressivamente mais difícil e cara.
- **O gerenciamento de custo, cronograma e configuração é mais complexo**
- **Se os requisitos são instáveis ou não descritos de forma completa quanto se esperava, alguns incrementos podem precisar ser retirados de uso ou causar retrabalhados**

Modelo Espiral



Modelo Espiral

- Modelo fortemente orientado à **análise e redução de riscos**.
- O processo é representado como uma espiral, onde cada *loop* representa uma fase do processo.
- O projeto é subdividido em subprojetos, cada qual abordando um ou mais elementos de alto risco, até que todos os riscos identificados possam ser tratados
- O conceito de risco é definido de maneira abrangente e pode envolver desde requisitos mal compreendidos até problemas tecnológicos, incluindo desempenho, arquitetura, questões de negócio, etc.
- Depois que os principais riscos foram mitigados, o processo segue de forma normal.

Modelo Espiral

- O Processo Espiral é similar ao Incremental mas
 - Cada ciclo produz algo a ser avaliado
 - não necessariamente código
 - Gerência de Riscos embutida no processo
 - Ao final de cada loop é perguntado **“Devemos continuar?”**

**Profissionais envolvidos
nos projetos de software**

Comparação entre as profissões

Profissão	Atividades desenvolvidas
Engenheiro de Software	Especifica os processos de planejamento, gerência e produção a serem implementados. Ele acompanha e avalia o desenvolvimento de todos os projetos de software para verificar se o processo estabelecido é seguido, e realiza mudanças no processo quando necessário.
Analista de sistema	É o profissional responsável pela compreensão do problema relacionado ao sistema que deve ser desenvolvido, ou seja, pelo levantamento dos requisitos e sua efetiva modelagem.
Arquiteto de software	Deve levantar em consideração as especificações do analista e propor a melhor tecnologia para produzir o sistema. É responsável pelo design das camadas ou partições do sistema (componentes, pacotes, camadas)

Comparação entre as profissões

Profissão	Atividades desenvolvidas
Programador	Irá desenvolver (programar) o software conforme especificações do projeto.
Analista de testes	Ele é responsável pelo projeto e elaboração dos casos de teste a serem aplicados ao sistema
Testador	O testador é responsável pela realização efetiva dos testes. Ele deve registrar os resultados dos testes e no caso de componentes que não passem nos testes deve informar os respectivos responsáveis para providências de correção.
Gerente de projetos	Ele é o responsável por planejar as atividades e alocar os recursos físicos e humanos, bem como acompanhar o projeto garantindo que prazos e orçamentos sejam cumpridos e tomando decisões de correção de rumo quando necessário.