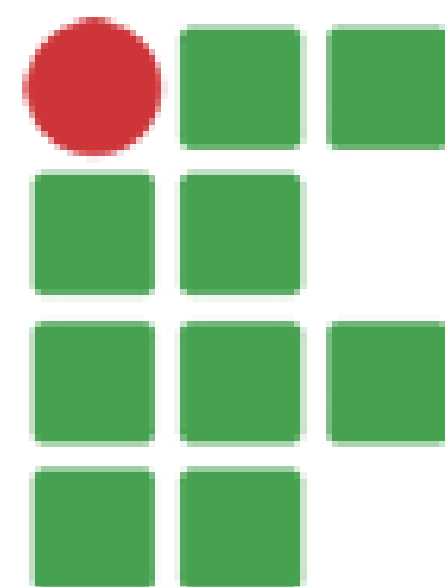




**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Rio de Janeiro

Disciplina

Modelagem de Domínio e
Conceitos de Orientação a Objetos

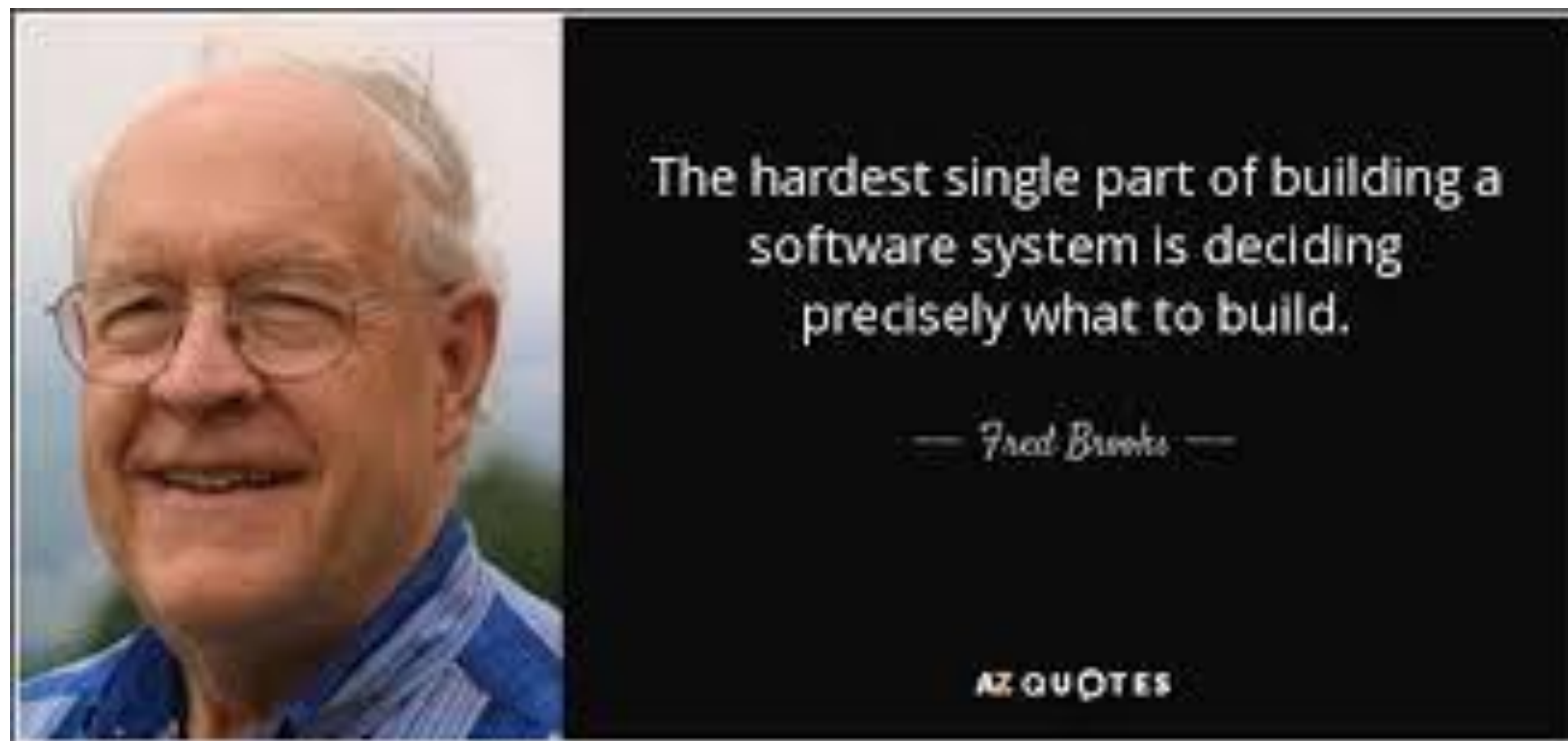
PROF. FERNANDO TAMBERLINI ALVES

- **Engenharia de Software:** processos, métodos e ferramentas de apoio que nos auxiliam no desenvolvimento de software;
- **Arquitetura de Software:** características de como o software será desenvolvido;
- **Processo de Desenvolvimento:** etapas específicas a serem percorridas durante o desenvolvimento de um software (Cascata e Ágil);
- **Engenharia de Requisitos:** é o processo de entender o que o sistema deve fazer (requisitos funcionais) e como ele deve se comportar (requisitos não funcionais).
- **Modelagem:** é a representação graficamente a estrutura, o comportamento e os requisitos do sistema antes de sua construção.

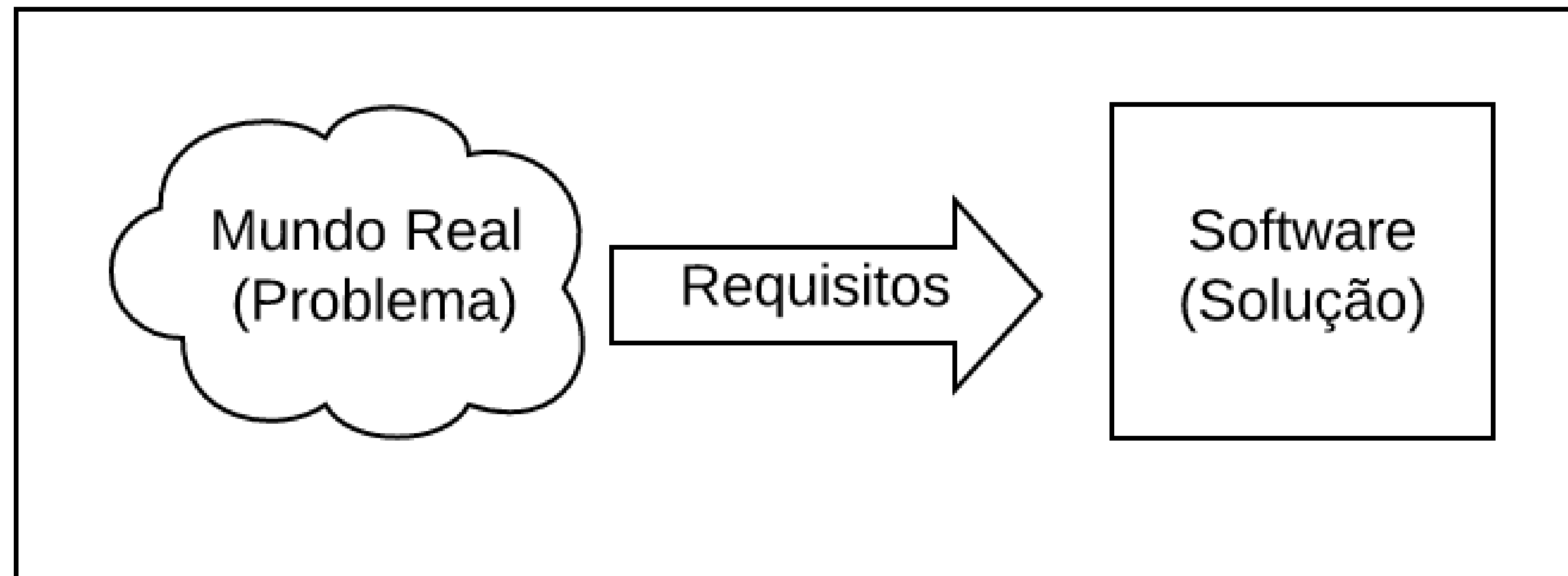
Engenharia de Requisitos

“A parte mais difícil da construção de um software é definir precisamente o que deve ser construído.”

Fred Brooks



Engenharia de Requisitos



Engenharia de Requisitos

Engenharia de Requisitos é uma das etapas mais importantes do processo de desenvolvimento de software. Ela tem como objetivo identificar, documentar, validar e gerenciar as necessidades e expectativas dos stakeholders (usuários, clientes, desenvolvedores, etc.) em relação ao sistema que será desenvolvido.

Principais Atividades da Engenharia de Requisitos:

1. Elicitação de requisitos
2. Análise de requisitos
3. Especificação de requisitos
4. Validação de requisitos
5. Gerenciamento de Requisitos

Engenharia de Requisitos

- **Elicitação de Requisitos**

É o processo de descobrir os requisitos através de entrevistas, questionários, observações, workshops, brainstorming, etc. Envolve entender o problema e as necessidades do cliente.

- **Análise de Requisitos**

Os requisitos coletados são analisados para garantir que sejam completos, consistentes, claros e viáveis. Muitas vezes, surgem conflitos entre requisitos que precisam ser resolvidos.

- **Especificação de Requisitos**

É a documentação formal dos requisitos. Pode ser feita em linguagem natural, modelos UML, diagramas, ou linguagens formais. O resultado mais comum é o Documento de Requisitos de Software (DRS ou SRS).

- **Validação de Requisitos**

Garante que os requisitos realmente representam o que o cliente deseja. Técnicas comuns incluem revisões, prototipação, simulações e testes.

- **Gerenciamento de Requisitos**

Consiste em acompanhar as mudanças nos requisitos durante o ciclo de vida do software, controlando versões, rastreabilidade e impacto das alterações.

Tipo de Requisitos

- **Requisitos Funcionais:**

Descrevem o que o sistema deve fazer (ex: "o sistema deve permitir login com senha").

- **Requisitos Não Funcionais:**

Descrevem restrições ou qualidades do sistema (ex: desempenho, segurança, usabilidade)

- **Requisitos de Domínio:**

Regras específicas da área de negócio (ex: "a taxa de juros deve ser recalculada conforme o CDI").

- **Requisito inverso:**

É um tipo de requisito que descreve explicitamente o que o sistema não deve fazer.

Tipo de Requisitos

Requisito de Usuário (User Requirement)

São descrições em linguagem natural (geralmente sem termos técnicos) sobre o que os usuários esperam do sistema. Focam no que o sistema deve fazer do ponto de vista do usuário ou quais objetivos ele precisa alcançar.

- Formato:
Texto descritivo, histórias de usuário (user stories), casos de uso.
- Exemplo:
“O usuário deve ser capaz de recuperar sua senha caso a esqueça.”
- Características:
Compreensível para o cliente e stakeholders não técnicos. Evita detalhes sobre implementação. Ajuda a alinhar expectativas..

Tipo de Requisitos

Requisito de Sistema (System Requirement)

Descreve como o sistema deve se comportar ou ser implementado para atender aos requisitos de usuário. São mais detalhados e técnicos.

- Formato:

Pode ser apresentado como especificações técnicas, diagramas UML, tabelas, requisitos funcionais e não funcionais.

- Exemplo:

“O sistema deve enviar um e-mail de redefinição de senha com um token válido por 24 horas após o usuário solicitar a recuperação de senha.”

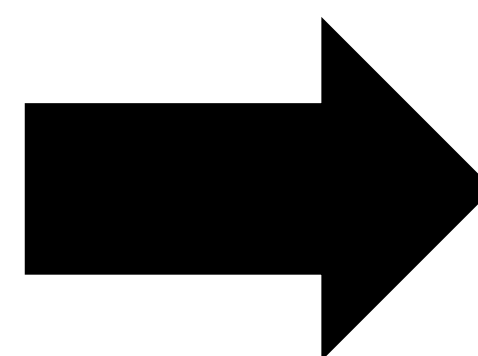
- Características:

Direcionado à equipe técnica (desenvolvedores, arquitetos, testers). Pode incluir restrições de hardware, software, performance, segurança.

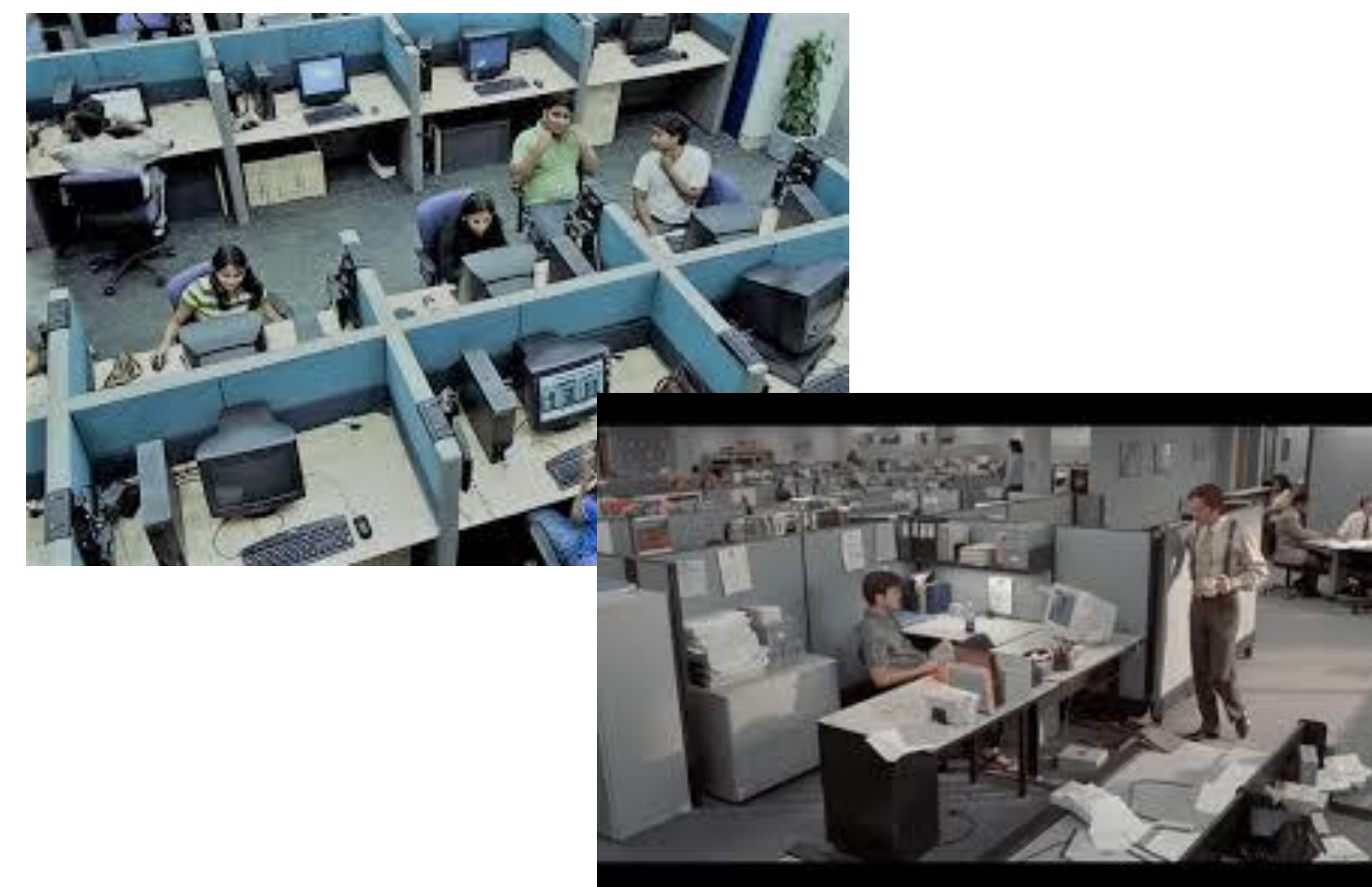
Elicitação dos Requisitos

Antes...

Analistas



Programadores



Elicitação dos Requisitos

Hoje...

Product
Owner
(PO)



Devs

Levantamento de Requisitos

Histórias de Usuário (User Stories)

Pequenas descrições das funcionalidades sob a perspectiva do usuário, focando no "quem", "o quê" e "por quê".

- Formato Clássico:

Pequenas descrições das funcionalidades sob a perspectiva do usuário, focando no "quem", "o quê" e "por quê".

- Exemplo:

Como [tipo de usuário]
Quero [ação ou funcionalidade]
Para que [benefício ou motivo].

- Características:

Linguagem simples, sem muitos detalhes técnicos. Usadas em metodologias ágeis (Scrum, Kanban).

Casos de Uso (Use Cases)

Descrevem os passos e interações entre o usuário (ator) e o sistema para alcançar um objetivo específico.

- Formato
 - Nome do caso de uso; Atores envolvidos. Fluxo principal (cenário de sucesso) e Fluxos alternativos ou de exceção
- Formato
 - Slide seguinte
- Características:
 - Mais detalhados que user stories. Muito usados em processos tradicionais (cascata) ou híbridos. Podem ser representados graficamente com diagramas UML.

Levantamento de Requisitos

Casos de Uso (Use Cases) - Exemplo

Transferir Valores entre Contas

Ator: Cliente do Banco

Fluxo normal:

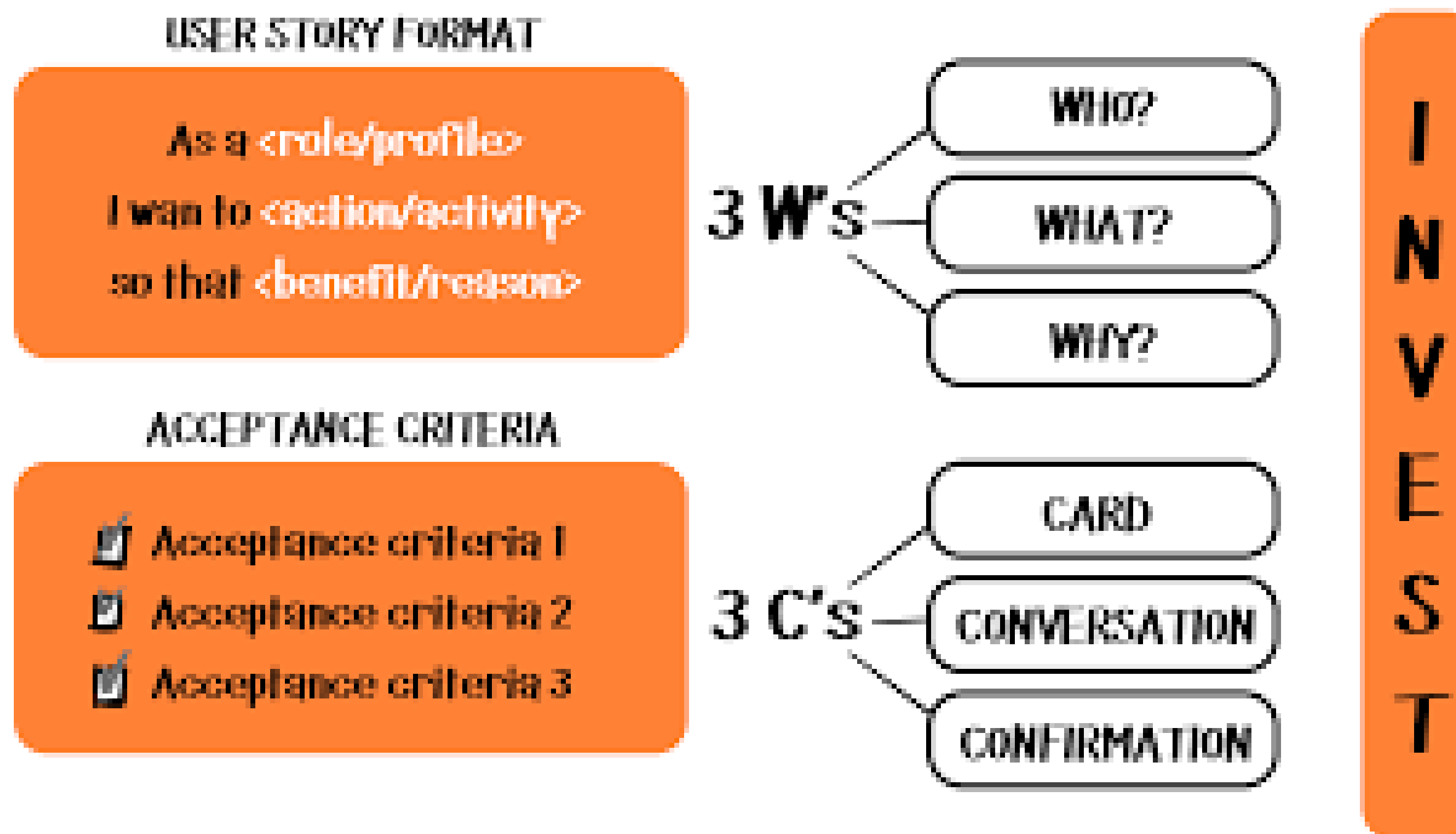
- 1 - Autenticar Cliente
- 2 - Cliente informa agência e conta de destino da transferência
- 3 - Cliente informa valor que deseja transferir
- 4 - Cliente informa a data em que pretende realizar a operação
- 5 - Sistema efetua transferência
- 6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

Extensões:

- 2a - Se conta e agência incorretas, solicitar nova conta e agência
- 3a - Se valor acima do saldo atual, solicitar novo valor
- 4a - Data informada deve ser a data atual ou no máximo um ano a frente
- 5a - Se data informada é a data atual, transferir imediatamente
- 5b - Se data informada é uma data futura, agendar transferência

História de Usuários

- 3W's, 3C's e INVEST



História de Usuários – 3W's

- **Formato Padrão – 3W's**
 - WHO - Como [tipo de usuário]
 - WHAT - Quero [o que deseja fazer]
 - WHY - Para que [benefício ou razão]

- **Exemplo**
 - Como cliente de uma loja virtual
 - Quero poder adicionar produtos a minha lista de favoritos
 - Para que eu consiga encontrá-los facilmente depois para compra-los.

História de Usuários – 3C's

- **Partes da História do Usuário – 3C's**
 - Card - Cartão
 - Conversation - Conversa
 - Confirmation - Confirmação

C	O que é	Função
CARD	O texto da User Story	Apresenta o que será feito
CONVERSTION	Conversa entre o time e o cliente	Detalha, negocia e esclarece a história
CONFIRMATION	Critérios de Aceitação	Define o que significa como “pronto”

História de Usuários – Invest

Letra	Significado	Descrição
I	Independent	Deve ser independente de outras histórias.
N	Negotiable	Deve ser aberta para discussão, não um contrato fechado.
V	Valuable	Deve gerar valor para o usuário ou cliente.
E	Estimable	Deve ser possível estimar o esforço para implementá-la.
S	Small	Deve ser pequena o suficiente para ser implementada em um sprint.
T	Testable	Deve ser possível validar se foi feita corretamente (testável).

- Cartão: "Fechar uma compra"
- Conversas: PO explica os meios de pagamento; as formas de entrega, formas de parcelamento, etc
- Confirmação:
 - Testar compra à vista e compra parcelado
 - Testar com cartões de crédito A, B e C
 - Testar com modos de entrega X e Y

Sistema Biblioteca

- 3 Tipos de Usuário:
 - Aluno
 - Professor
 - Bibliotecário

- Histórias de Alunos:
 - Como aluno, eu gostaria de realizar empréstimos de livros
 - Como aluno, eu gostaria de devolver um livro que tomei emprestado
 - Como aluno, eu gostaria de renovar empréstimos de livros
 - Como aluno, eu gostaria de pesquisar por livros
 - Como aluno, eu gostaria de reservar livros que estão emprestados
 - Como aluno, eu gostaria de receber e-mails com novas aquisições

- Histórias de Professores:
 - Como professor, eu gostaria de realizar empréstimos de maior duração
 - Como professor, eu gostaria de sugerir a compra de livros
 - Como professor, eu gostaria de doar livros para a biblioteca
 - Como professor, eu gostaria de devolver livros em outras bibliotecas

- Histórias de Bibliotecário
 - Como bibliotecário, eu gostaria de cadastrar novos usuários
 - Como bibliotecário, eu gostaria de cadastrar novos livros
 - Como bibliotecário, eu gostaria de dar baixa em livros estragados
 - Como bibliotecário, eu gostaria de obter estatísticas sobre o acervo
 - Como bibliotecário, eu gostaria que o sistema envie e-mails de cobrança para alunos com empréstimos atrasados
 - Como bibliotecário eu gostaria que o sistema aplicasse multas quando da devolução de empréstimos atrasados

1. Escreva três histórias de usuário para o serviço de PIX de um banco
2. Suponha que você trabalha em sistema de ensino a distância
 - a. Proponha uma história épica para esse sistema
 - b. Quebre essa história em histórias menores.
3. Seja um sistema de comércio eletrônico.



Seja a seguinte história de usuário desse sistema:

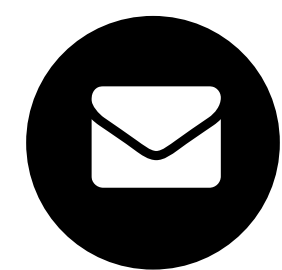
“Como cliente, eu gostaria de listar todas as minhas compras passadas”

Escreva os testes ou critérios de aceitação para essa história.

Dúvidas?



Dúvidas?



fernando.alves@ifrj.edu.br



ftamberlini.dev.br

Referências: VALENTE, M. T (2020). Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade, Editora: Independente.