



SÃO  
PAULO  
TECH  
SCHOOL

# Engenharia de Software

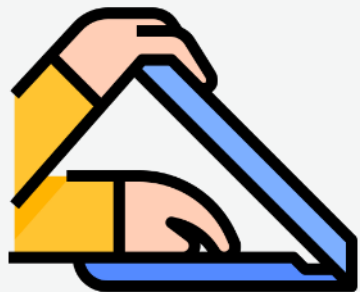
## Qualidade de Software

Aula 10

**Fábio Figueredo**

[fabio.figueredo@sptech.school](mailto:fabio.figueredo@sptech.school)

# Regras básicas da sala de aula



1. **Notebooks Fechados:** Aguarde a liberação do professor;
2. Celulares em modo **silencioso e guardado**, para não tirar sua atenção
  - Se, caso haja uma situação urgente e você precisar **atender ao celular**, peça licença para sair da sala e atenda fora da aula.



3. **Proibido usar Fones de ouvido:** São liberados apenas com autorização do professor.

4. **Foco total no aprendizado**, pois nosso tempo em sala de aula é precioso.

- Venham sempre com o **conteúdo da aula passada em mente** e as atividades realizadas.
- Tenham caderno e caneta;
- **Evitem faltas e procure ir além** daquilo que lhe foi proposto.
- **Capricho, apresentação e profundidade** no assunto serão observados.
- **“frequentar as aulas** e demais atividades curriculares aplicando a **máxima diligência no seu aproveitamento**” (Direitos e deveres dos membros do corpo discente - Manual do aluno, p. 31)



# Regras básicas da sala de aula



As aulas podem e devem ser divertidas! Mas:

- **Devemos respeitar uns aos outros** – cuidado com as brincadeiras.
  - “observar e cumprir o regime escolar e disciplinar e comportar-se, dentro e fora da Faculdade, **de acordo com princípios éticos condizentes**” (Direitos e deveres dos membros do corpo discente – Manual do aluno, p. 31)

# COMPROMISSO



COM VOCÊ:  
**ARRISQUE**, NÃO  
TENHA MEDO DE  
ERRAR



COM OS  
PROFESSORES:  
ORGANIZE A **ROTINA**  
PARA OS ESTUDOS

COM OS COLEGAS:  
**PARTICIPAÇÃO**  
**ATIVA** E PRESENTE



COM O PROJETO:  
**RESPEITO** E  
FLEXIBILIDADE

  
**Respeito**

# Boas práticas no Projeto

Reações **defensivas** não levam  
ao envolvimento verdadeiro!

Transforme cada problema e  
cada dificuldade em uma  
**OPORTUNIDADE** de aprendizado  
e crescimento.

## **EVITE:**

- Justificativas e Desculpas
- Transferir a culpa
- Se conformar com o que sabe
- Se comparar com o outro

# Dica: **Como ter sucesso** (Maiores índices de aprovações)

## Comprometimento

- Não ter faltas e atrasos. Estar presente (*Não fazer 2 coisas ao mesmo tempo*)
- Fazer o combinado cumprindo os prazos

## Atitudes Esperadas:

- **Profissionalismo**: Entender que não é mais ensino médio (*Atitude, comportamento, etc.*)
- **Não estar aqui só pelo** estágio ou pelo diploma
- Não ficar escondido: precisa **experimentar**
- **Trabalhar** em grupo e **participar** na aula
- **Não ser superficial** ou “achar que sabe”
- **Não se enganar** utilizando de “cola”
- Assumir a responsabilidade: Não colocar a culpa em outra coisa. **Não se vitimizar.**

# Nosso Caminho



S3

- Qualidade e Testes
- Processos de Software
- Projeto de Software
- Apresentação PI
- Avaliação Integrada

S2

- ~~Design Inclusivo~~
- ~~Ciclo de Vida de Produto~~
- ~~Teste de Usabilidade~~
- ~~Arquitetura de Software~~

S1

- ~~Introdução a Engenharia de Software~~
- ~~Conceitos de UI e UX~~
- ~~Fatores Humanos~~
- ~~Personas~~
- ~~Design de Interface Básico~~



Palavra-chave dessa Sprint:

# PRAGMATISMO

**prag·má·ti·co**

. adjetivo

1. Relativo à pragmática ou ao pragmatismo.

2. **Que tem motivações relacionadas com a ação ou com a eficiência. =**

**PRÁTICO**

. adjetivo e substantivo masculino

3. Que ou quem revela um sentido prático e sabe ou quer agir com eficácia.





**Frase dessa sprint:**

**Aprender/Ensinar** processos, métodos e ferramentas para construção e manutenção de **softwares profissionais.**

# Tópicos da Aula

- Entregáveis de PI – Sprint 3
- Qualidade pt1

## ENTREGÁVEIS DE PI – SPRINT 3

- **Planilha UAT** – Cenários core do sistema com evidência de execução;
- **Projeto Web aderente aos requisitos de UI e UX;**



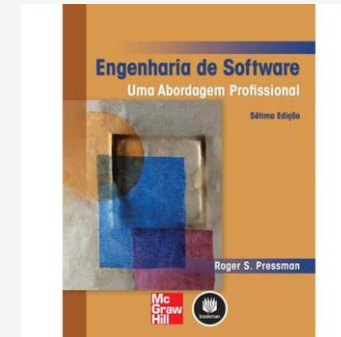
# QUALIDADE



# Bibliografia

Engenharia de Software 8ª Edição / Ian Sommerville

Engenharia de Software 6ª Edição / Roger Pressman



Adicional

Code Complete

SWEBOK

entre outros

The background is a dark, textured surface with intricate, glowing patterns. These patterns consist of numerous thin, curved lines that flow and swirl across the frame, creating a sense of motion and depth. Interspersed among these lines are many small, bright, out-of-focus light points, resembling stars or bokeh from a camera lens. The overall effect is a sophisticated, high-tech aesthetic.

# O que é Qualidade de Software?

# O que é Qualidade?

## No Aurélio

- 1 - Maneira de ser boa ou má de uma coisa.
- 6 - Aquilo que caracteriza uma coisa.
- 10 - Atributo, modalidade, virtude, valor.



## Em Software

- A qualidade engloba os requisitos, especificações e o design do sistema.

**Satisfação do Cliente =  
Produto compatível + Boa qualidade (Produto útil)  
+ Entrega dentro do orçamento e do cronograma**



# Produto Útil

Um produto útil fornece o conteúdo, funções e recursos que o usuário final deseja.

Um **produto útil** sempre **satisfaz** os **requisitos** que foram **explicitamente declarados** pelas partes interessadas.

Além disso, satisfaz um conjunto de **requisitos implícitos** (por exemplo, facilidade de uso, convenções, etc) **que são esperados de todos os softwares de alta qualidade**.

Por exemplo, o campo CEP em um formulário.

# O que é Qualidade para o PMBOK

O gerenciamento da qualidade do projeto inclui os processos e as atividades da organização executora que determinam as políticas de qualidade, os objetivos e as responsabilidades, **de modo que o projeto satisfaça às necessidades para as quais foi empreendido.**



Qual tem mais qualidade?



# Gerenciamento da Qualidade (ISO)

**De acordo com o ISO** (Organização Internacional para Padronização), **qual tem mais qualidade?**

**Segundo a norma ISO 9000** (versão 2000), **a qualidade é o grau em que um conjunto de características inerentes a um produto, processo ou sistema cumpre os requisitos inicialmente estipulados para estes:**

**Qualidade do produto** – As medidas e técnicas de qualidade são específicas do tipo de entregas resultantes do projeto. Pode ser expressa como: desempenho, grau, durabilidade, suporte de processos existentes, defeitos e erros.

**Grau** – É uma categoria atribuída aos produtos ou serviços que têm a mesma utilidade funcional, mas diferentes características técnicas. Ter uma baixa qualidade é ruim, mas ter um baixo grau não é necessariamente ruim.

**Precisão** – Significa que os valores de medições repetidas estão agrupados e têm pouca dispersão. Nem todas as medidas precisas são exatas.

**Exatidão** – Significa que o valor medido está bem próximo do valor correto.

# Geração de Valor para o Negócio

**A empresa ganha com software de alta qualidade, pois este requer menos esforço de manutenção, menos correções de bugs e menor suporte ao cliente.**

**O usuário também ganha** porque o software de alta qualidade fornece um recurso útil de maneira a agilizar os processos de negócios.

O resultado final é:

- maior receita de produto de software,
- melhor rentabilidade quando uma aplicação suporta um processo de negócio, e / ou
- melhor disponibilidade de informações que são cruciais para o negócio.

**O objetivo do Scrum Master / Gerente de Projetos é gerar valor para o negócio.**

# Atributos de Qualidade

## 1. Usabilidade

- Fatores Humanos
- Estética
- Documentação

## 2. Confiabilidade

- Frequência e severidade das falhas (\* MTBF, \* MTTR)

## 3. Desempenho

- Velocidade de Processamento
- Escalabilidade
- Disponibilidade

## 4. Facilidade de Suporte

- Extensibilidade – Capacidade de receber novas funcionalidades
- Compatibilidade
- Reparabilidade

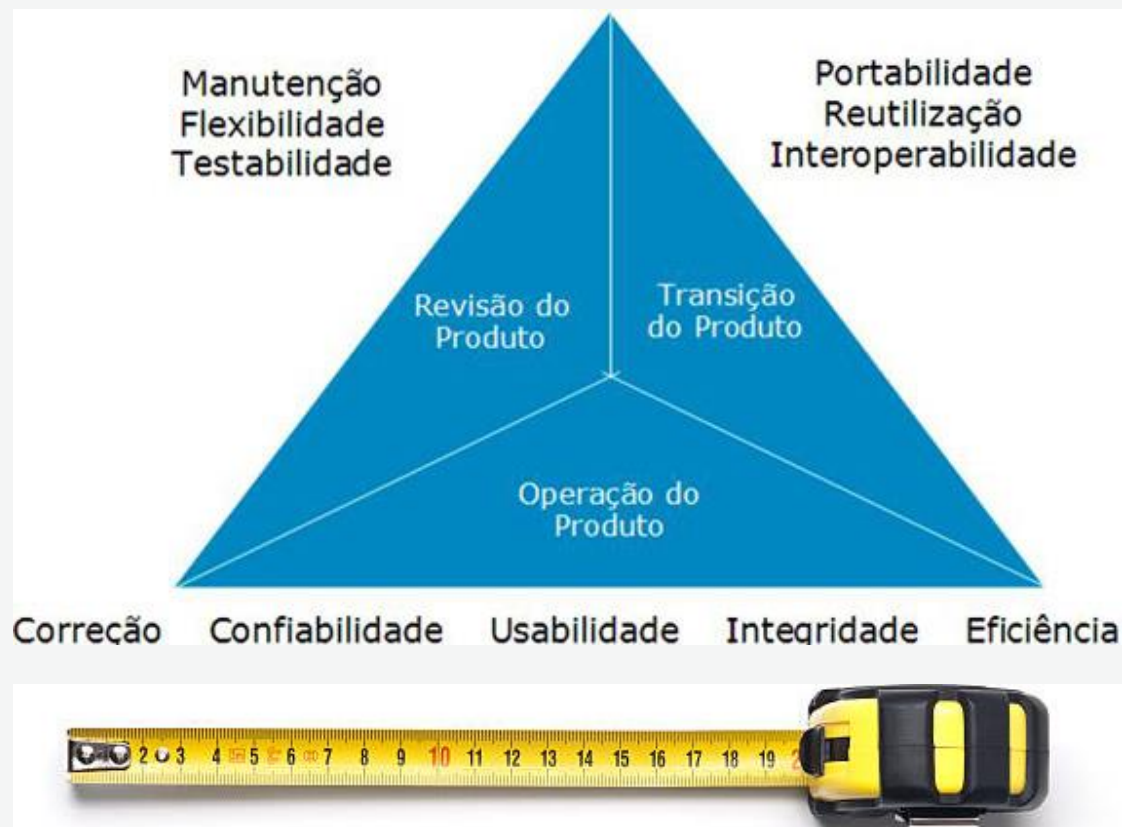


*Os Fatores da Qualidade de McCall.*

\* **MTBF** – Tempo médio entre falhas / **MTTR** – Tempo médio entre os reparos

# Como medir a qualidade?

## Você precisa estabelecer métricas!



As equipes de projeto precisam desenvolver um conjunto de perguntas específicas para avaliar o grau em que cada fator de qualidade do aplicativo foi satisfeito.

Medidas subjetivas de qualidade de software podem ser vistas como pouco mais que uma opinião pessoal.

Métricas de software representam medidas indiretas de alguma manifestação de qualidade e tentam quantificar a avaliação da qualidade do software.

**Exemplo de Métrica: Qualidade da aula baseado na quantidade de bocejos durante uma hora de aula.**



# Atividade

**Você acabou de assumir a gestão de uma área de desenvolvimento de software. O fluxo de trabalho atual é:**

1. Especificação Funcional
2. Especificação Técnica
3. Desenvolvimento com Teste Unitário
4. Teste Integrado
5. Implantação (Produção)

**Você precisa definir pelo menos 1 métrica para aferição da qualidade. Defina uma métrica para esse fluxo.**



# Atividade - Correção

## 1. Especificação Funcional

1. Quantidade de Requisitos – Avaliar a completude do documento.
2. Taxa de Correção – Porcentagem de erros e inconsistências encontradas.
3. Tempo médio de resposta a dúvidas – Quanto tempo o cliente demora para responder.

## 2. Especificação Técnica

1. Quantidade de Requisitos – Avaliar a completude do documento.
2. Nível de Satisfação dos Stakeholders Técnicos – Satisfação da equipe técnica do cliente que valida e aprova a Especificação Técnica.
3. Número de Mudanças – Quantidade de mudanças realizadas no documento durante o projeto.

## 3. Desenvolvimento com Teste Unitário

1. Cobertura de código com teste unitário - % de código que está coberto com os testes unitários.
2. Tempo médio de execução dos testes – Tempo necessário para executar os testes unitários.
3. Frequência de erros – Quantos erros são encontrados durante a realização dos testes.

# Atividade - Correção

## 4. Teste Integrado

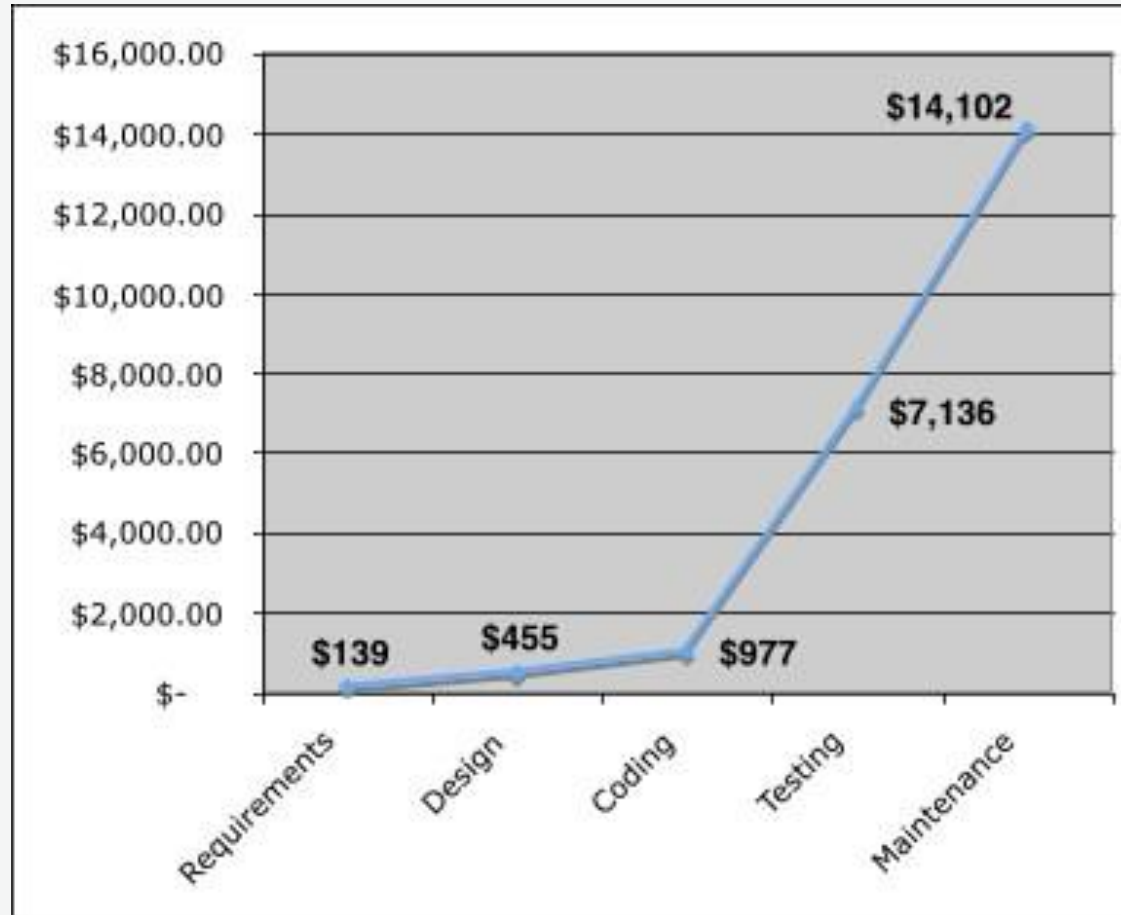
1. Número de erros encontrados
2. Tempo de correção
3. Cobertura do caminho crítico – Quantidade dos itens críticos para a aplicação que foram testados.

## 5. Implantação (Produção)

1. Tempo de indisponibilidade – Tempo que a aplicação ficou indisponível.
2. Tempo médio para recuperação de uma falha – Tempo para reestabelecer a aplicação.
3. Quantidade de Releases – Quantidade de implantações realizadas no período.

# Quanto custa a qualidade?

**É melhor prevenir do que remediar!**



Veja no exemplo que o custo de reparação fica extremamente alto logo depois que sai da mão do desenvolvedor.

Conseguem pensar o por quê?

O PO está na empresa para garantir que isso não aconteça!

# Quanto custa a Qualidade?

## Custo de Prevenção

- Planejamento de qualidade (QA, Teste Unitário, PO, Scrum Master, ...)
- Equipamentos/Ambientes de teste
- Treinamento

## Custos de Avaliação (Internos)

- Testes e Depuração
- Coleta de dados e métricas

## Custo de Falha (Internos)

- Retrabalho e Correção
- Efeitos colaterais (altera uma coisa, estraga outra)
- Coleta de Dados e Métricas
- Desgaste da Equipe

## Custos Externos

- Resolução de reclamações
- Retorno e substituição do produto
- Suporte (SAC)
- Reputação
- Satisfação do Cliente
- Responsabilidade Civil

[https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwi6gODV6fXhAhXsGLkGHabKD8oQjRx6BAgBEAU&url=http%3A%2F%2Fwww.folhavitoria.com.br%2Feconomia%2Fblogs%2Fgestaoresultados%2F2012%2F02%2F15%2Fcustos-da-qualidade-1a-etapa%2F&psig=A0vVaw2Ha92fZgovRT\\_oLGGyRDPC&ust=1556645269822925](https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwi6gODV6fXhAhXsGLkGHabKD8oQjRx6BAgBEAU&url=http%3A%2F%2Fwww.folhavitoria.com.br%2Feconomia%2Fblogs%2Fgestaoresultados%2F2012%2F02%2F15%2Fcustos-da-qualidade-1a-etapa%2F&psig=A0vVaw2Ha92fZgovRT_oLGGyRDPC&ust=1556645269822925)



# Quanto custa a Qualidade?

**Better  
SAFE**



**Than  
Sorry!**



## Algumas decisões que impactam na qualidade:

**Decisões de estimativa** – estimativas irracionais de data de entrega levam as equipes a usar atalhos que podem levar à redução da qualidade do produto

**Decisões de Cronograma** – não prestando atenção às dependências de tarefas ao criar o cronograma do projeto

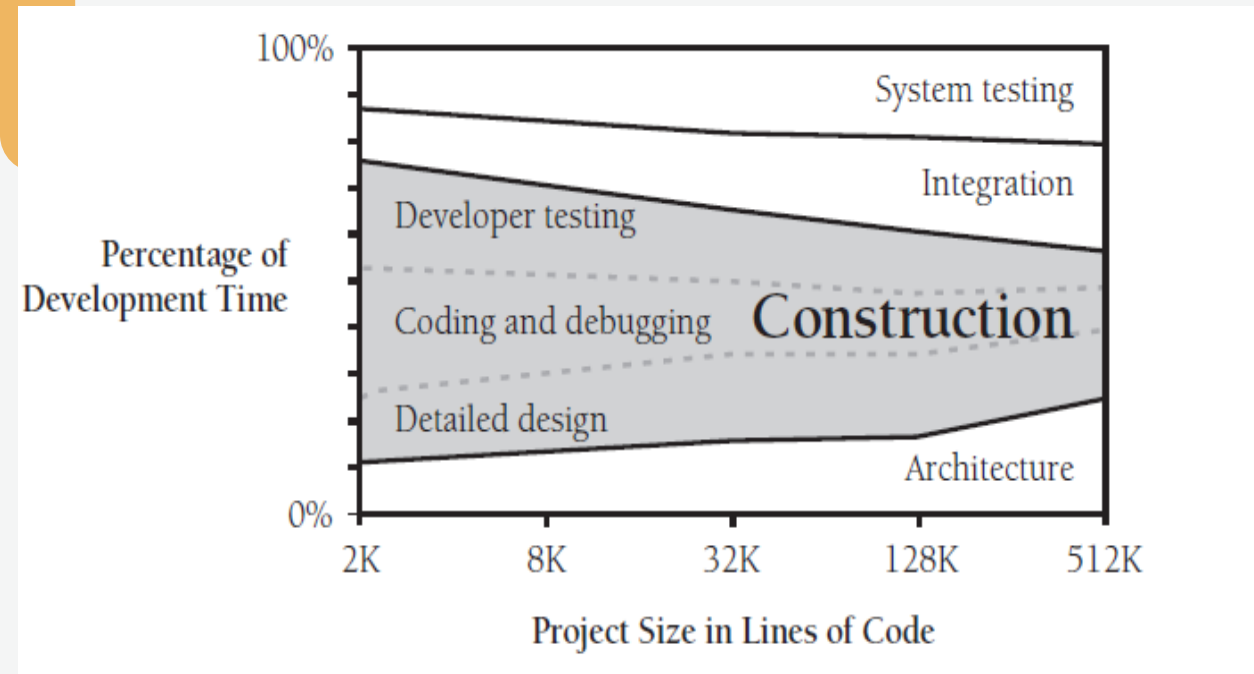
**Decisões orientadas ao risco** – Reagir ao invés de planejar – reagir a cada crise à medida que ela surge, em vez de criar mecanismos para monitorar os riscos, pode resultar em produtos com qualidade reduzida.

# Gerenciamento da Qualidade (PMBOK)

**O gerenciamento da qualidade do projeto trabalha para garantir que os requisitos do projeto, incluindo os requisitos do produto, sejam cumpridos e validados.**

- **Planejar o gerenciamento da qualidade** – O **processo de identificação dos requisitos** e/ou padrões da qualidade do projeto e suas entregas, além da documentação de como o projeto demonstrará a conformidade com os requisitos e/ou padrões de qualidade.
- **Realizar a garantia da qualidade** – O **processo de auditoria dos requisitos de qualidade** e dos resultados das medições do controle de qualidade para garantir o uso dos padrões de qualidade e das definições operacionais apropriadas.
- **Realizar o controle da qualidade** – O **processo de monitoramento e registro dos resultados da execução das atividades de qualidade** para avaliar o desempenho e recomendar as mudanças necessárias.

# Por que é importante a Eng.de Software



A medida que o tamanho do sistema aumenta, o tempo em nas atividades de Arquitetura, Integração e Teste também aumentam.

! O tempo da construção do software reduz proporcionalmente

Livro *Code Complete* pag 654

## Atividades que aumentam:

- Comunicação
- Planejamento
- Gerenciamento
- Lev. Requisitos
- Projeto Funcional
- Arquitetura
- Integração
- Remoção de Defeitos
- Testes
- Documentação

# Abordagens para Gestão da Qualidade

## Satisfação do cliente

Entender, avaliar, definir e gerenciar as expectativas para que os requisitos do cliente sejam atendidos.



**Escopo, contrato, Especificações detalhadas, Protótipos, etc.**

## Prevenção ao invés de inspeção

A qualidade deve ser planejada, projetada e criada, e não inspecionada no decorrer do projeto ou nas entregas do projeto



**Custo da prevenção é menor que o custo do reparo durante o uso.**

## Melhoria contínua

Com iniciativas de programas de qualidade



**CMMI , ISO , MPS.BR**

## Custo da Qualidade

Onde e como é mais efetivo empregar o controle de qualidade para se obter o melhor custo-benefício



**Custo de conformidade – fazer corretamente, Custo da falta de conformidade – não fazer corretamente.**



# Kahoot!





**Vamos a um exemplo**

**UX + Análise**

**Melhora a Qualidade?**

# O que era a demanda e o que foi entregue...

**Criar uma nova tela de cadastro de produtos para o estoque, para que os responsáveis pelo estoque consigam realizar o cadastro de novas unidades sem precisar entrar na tela de cadastro de produto. O cliente precisa conseguir exportar os registros.**

Cadastro do Estoque

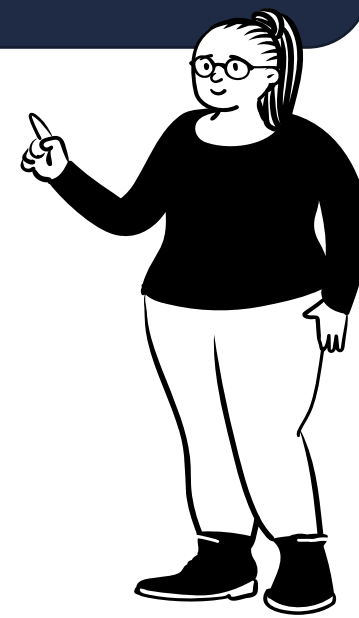
Digite o código ou nome do produto

Consulta

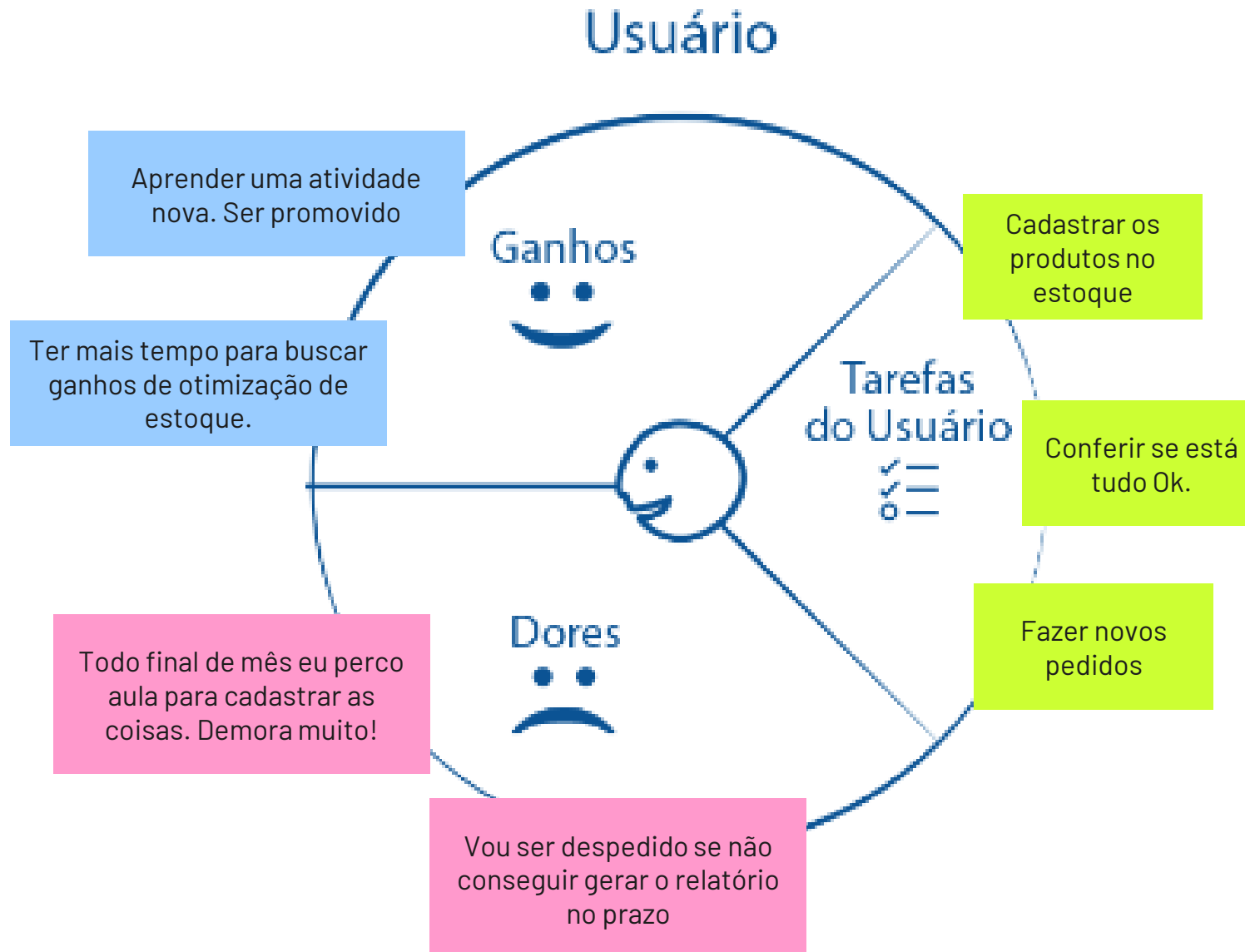
Produto	Qtde Antiga	Adicionar Unidades	Ação
Massa corrida	10	5	Salvar

Gerar Excel

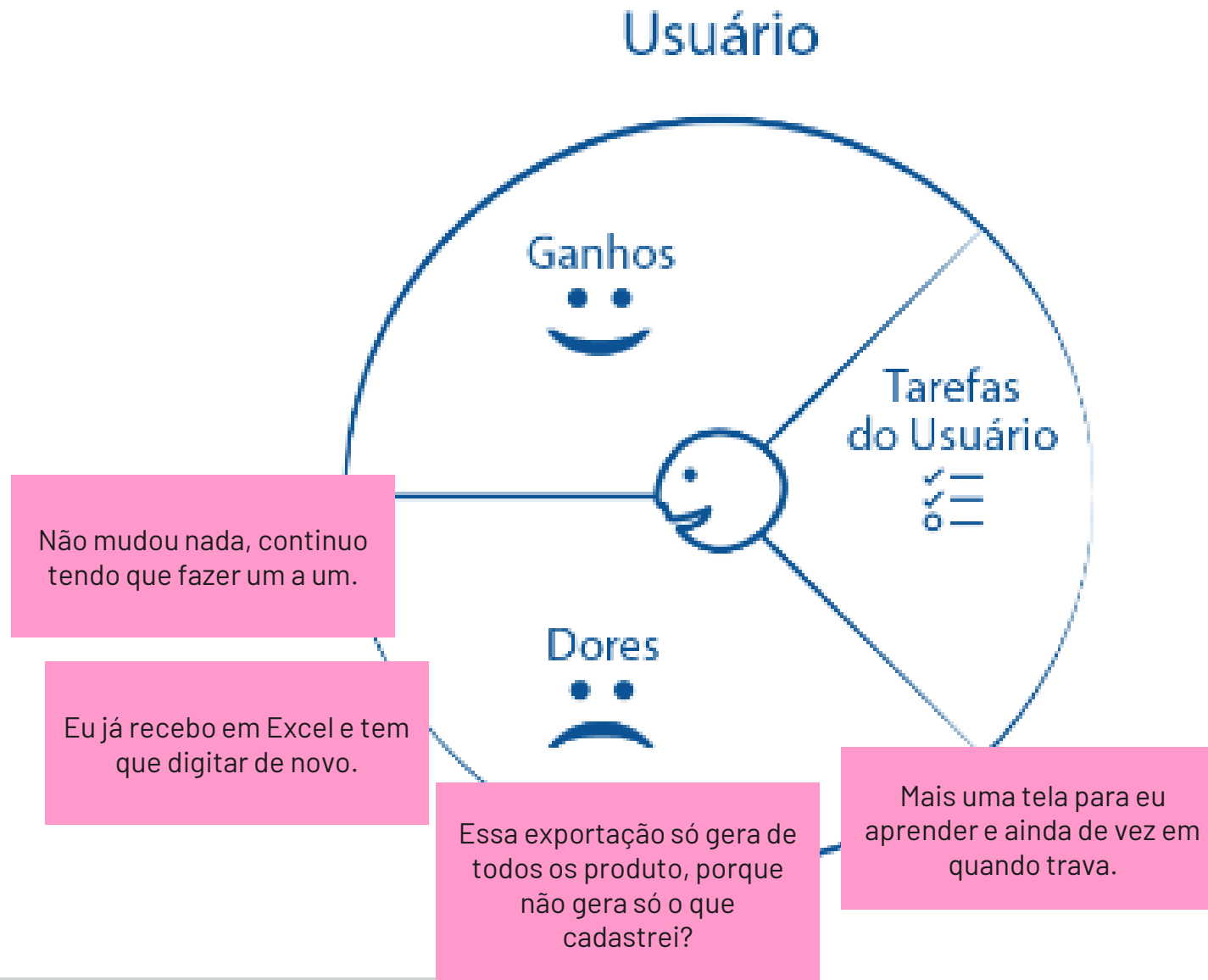
Até parece funcional,  
mas vamos ver qual a  
real necessidade do  
usuário...



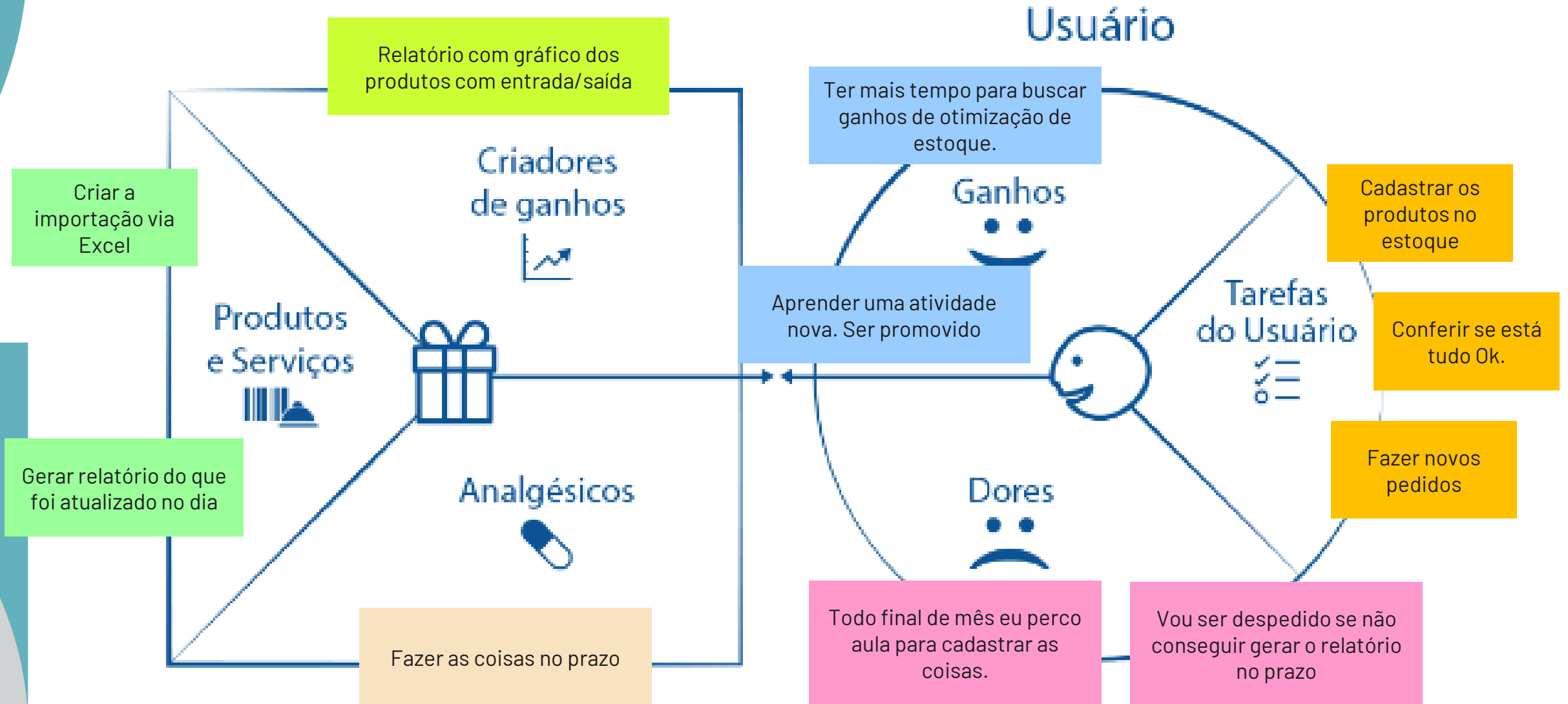
# Vamos exercitar a EMPATIA – **Antes do sistema**



# Vamos exercitar a EMPATIA – **Depois do Sistema**



# Vamos exercitar a EMPATIA (Visão Completa)



# CMMI e MPS.BR



# CMMI e MPS.BR

**CMMI** – Desenvolvido pela SEI – Universidade de Carnegie Mellon

## Modelo de Capacidade e Maturidade Integrado

**Objetivo:** Conjunto de práticas que servem como referência para que empresas possam melhorar os processos e desempenho no desenvolvimento de produtos e serviços.

**Programa MPS.BR** – SOFTEX com apoio do Ministério da Ciência, Tecnologia e Inovação.

## Melhoria de Processo do Software Brasileiro

**Objetivo:** Melhorar a capacidade de desenvolvimento de software nas empresas brasileiras.

Desenvolver e disseminar modelos de melhoria de processos que atendam às necessidades da Indústria Brasileira de Software e Serviços de TI.



## CMMI e MPS.BR


A certificação **CMMI** é muito cara, pelo valor somente algumas áreas tem.

Dado que era muito caro implementar o **CMMI** o **Brasil criou uma certificação similar baseada no CMMI.**

A certificação foi quebrada em blocos (níveis) para ficar mais acessível para as empresas.

# Modelos - CMMI vs MPS.BR



<b>5</b>	Análise Causal e Resolução - CAR Inovação e Melhoria Organizacional - OID	<b>A</b>	Análise de Causas de Problemas e Resolução 
<b>4</b>	Desempenho do Proc. Organizacional - OPP Gerência Quantitativa de Projeto - QPM	<b>B</b>	Gerência Quantitativa do Projeto
<b>3</b>	Foco no Processo da Organização - OPF Definição do Processo da Organização - OPD Treinamento Organizacional - OT Gerência Integrada de Projeto - IPM Gerência de Risco - RSKM Desenvolvimento de Requisitos - RD Solução Técnica - TS Integração de Produto - PI Verificação - VER Validação - VAL Análise de Decisão e Resolução - DAR	<b>C</b>	Análise de Decisão e Resolução Gerência de Riscos Desenvolvimento de Reutilização
		<b>D</b>	Desenvolvimento de Requisitos Integração do Produto Projeto e Construção do Produto Verificação Validação
		<b>E</b>	Gerência de Recursos Humanos Avaliação e Melhoria do Processo Org. Definição do Processo Organizacional Gerência de Reutilização
<b>2</b>	Gerência de Requisitos - REQM Planejamento de Projeto - PP Acompanhamento e Contr. de Projeto - PMC Gerência de Acordos com Fornecedores - SAM Gar. da Qual. de Processo e Produto - PPQA Gerência de Configuração - CM Medição e análise - MA	<b>F</b>	Medição Gerência de Configuração Aquisição Garantia da Qualidade
		<b>G</b>	Gerência de Requisitos Gerência de Projeto

(ISO 15.504)

## MPS/BR – Mais informações

<https://softex.br/mpsbr/>

<https://www.promovesolucoes.com/quais-sao-os-niveis-de-maturidade-do-mps-br/>



# CRITÉRIOS DE ACEITAÇÃO & UAT

# Não esquecer nunca!

De **NADA** adianta um software sem bug que não atende o que o cliente pediu!

## Atenção para:

- **REQUISITO DETALHADO**
- **CRITÉRIO(S) DE ACEITAÇÃO**  
(Não é Teste de Aceitação, é visão do cliente)

- **Critérios de Aceitação** é o que deve ser testado.
- **Teste de Aceitação** são os testes realizados para garantir que os Critérios de Aceitação foram atendidos.



# TESTE UAT

Teste de Aceitação  
do usuário



# Teste UAT

**Teste de Aceite (ou Aceitação) do Usuário**, é a **última fase do processo de teste de software**, onde é verificado se o produto atende a necessidade para o qual foi construído.

É um teste realizado pela equipe de QA (Testes) e normalmente replicado pelo cliente – o PO pode (ou deve) executar junto com o cliente.

**O Teste UAT traduz as regras de negócios e funcionalidades, em ações que devem ser executadas pelo software.**

**O Teste UAT deve cobrir TODOS os cenários de testes da aplicação.**



# Teste UAT – Caso, Cenário e Script de Teste

**Caso de Teste** – É a funcionalidade que será testada. Exemplo:

- Testar o cadastro de usuários;

**Cenário de Teste** – É a descrição da funcionalidade que será testada . Exemplo:

- Cadastrar um usuário;
- Cadastrar um produto;
- Preencher o formulário de contato;
- Navegar em determinada página;

**Script de Teste** – São as ações necessárias para realizar o teste. Exemplo:

- Entrar na página principal do site/software, clicar no botão cadastro e preencher o CPF.
- Entrar na página principal do site/software e clicar no link Sobre Nós;

# Teste UAT – Caso, Cenário e Script de Teste

**Massa de Teste** – São os dados que devem ser utilizados no teste. Exemplo:

- Digitar no campo CPF o valor: 222.222.222-22;
- Deixar o campo CPF vazio;
- Digitar no campo CPF o valor: 123abc;

**Resultado Esperado** – É a descrição do resultado que deve acontecer quando o usuário utiliza a massa de testes definida. Exemplo:

- Script de Teste: Entrar na página principal do site/software, clicar no botão cadastro e preencher o CPF.
- Massa de Teste: digitar no campo CPF o valor: 222.222.222-22;
- **Resultado Esperado**: Exibir mensagem: O CPF inserido é inválido.

# Scripts e Massa de Teste - Vamos testar a inserção de dados de um CPF em um Formulário

Para acessar seus documentos,  
por favor informe seu nº de CPF

CPF

---

## Caminho Feliz

- Inserir um CPF válido;

## Caminho Infeliz

- Inserir um CPF inválido;
  - Preencher com somente letras;
  - Preencher com letras e números;
  - Preencher com dados iguais (ex: 222.222.222-22);
  - Preencher com conteúdo e espaço;
  - Preencher com caracteres faltando;
  - Preencher com caracteres em excesso;
  - Preencher com emoji; ...
- Deixar vazio;

# Scripts e Massa de Teste - Vamos testar a inserção de dados de um CPF em um Formulário

## Caminho Feliz

- Inserir um CPF válido;

## Caminho Infeliz

- Inserir um CPF inválido;
- Deixar vazio;
- Preencher com somente letras;
- Preencher com letras e números;
- Preencher com dados iguais (ex: 222.222.222-22);
- Preencher com conteúdo e espaço;
- Preencher com caracteres faltando;
- Preencher com caracteres em excesso;
- Preencher com emoji; ...

**Precisamos testar o comportamento do sistema e o feedback!**

- O que a aplicação precisa retornar a cada tipo de erro?
  - CPF inválido?
  - CPF precisa ser preenchido?
  - CPF suporta somente números?
  - CPF incompleto?
- Todos os erros estão tratados?
- Como as informações estão sendo gravadas no banco de dados?

# DICA: Feedback para o usuário e segurança



The image shows a Facebook login interface. At the top is the Facebook logo. Below it is the text "Entrar no Facebook". A red-bordered box contains the error message "Credenciais inválidas" and "Nome de usuário ou senha inválida". Below this are two input fields: the first contains the email "dasdas@fasfdas.com" and the second is labeled "Senha". A blue "Entrar" button is below the fields, and a link "Esqueceu a senha?" is at the bottom.

- Quando é um formulário de autenticação (**login e senha**), por motivo de segurança **o feedback para o usuário precisa ser genérico.**

**Entregável de PI**

# **Planilha de Testes UAT**

**> Cenários core do projeto**

# Critérios de Aceitação

Os **Critérios de Aceitação** são as **condições** que um **produto de software deve atender para ser aceito por um usuário, cliente ou outro sistema**. Eles são exclusivos para cada história de usuário e **definem o comportamento** do recurso **da perspectiva do usuário final**. **Critérios de aceitação bem escritos ajudam a evitar resultados inesperados.**



# BDD

## Critérios de Aceitação com BDD



# Exemplos de Critérios de Aceitação utilizando BDD

## BDD (Behavior Driven Development) – Desenvolvimento Orientado a Comportamento

- Abordagem de desenvolvimento que se concentra em descrever o comportamento esperado de forma clara, simplificando o entendimento dos envolvidos (clientes, desenvolvedores, analistas, etc...).

**Dado** uma condição, **Quando faço** alguma ação,  
**Então espero** algum resultado.

## Exemplos de Critérios de Aceitação – BDD (Behavior Driven Development)

### > Cenário: Compra com cartão (sem crédito)

**Dado que** eu não tenha crédito suficiente, **quando** eu tentar fazer uma compra, **então espero** que o sistema bloqueie e emita uma mensagem significativa.

### > Cenário: Cadastrar produto (sem acesso)

**Dado que** eu não tenha acesso para cadastrar novo produto, **quando** eu tentar acessar a tela de cadastro, **então espero** que a opção esteja desabilitada.

# Atividade – Critérios de Aceitação – Individual

## 1. Usabilidade

- Fatores Humanos
- Estética
- Documentação

## 2. Confiabilidade

- Frequência e severidade das falhas (MTBF, MTTR)

## 3. Desempenho

- Velocidade de Processamento
- Escalabilidade
- Disponibilidade

## 4. Facilidade de Suporte

- Extensibilidade – Capacidade de receber novas funcionalidades
- Compatibilidade
- Reparabilidade.

Analise os atributos de qualidade ao lado e **pensando como CLIENTE** detalhe 5 critérios de aceitação para o seu projeto de PI.

### Exemplo Usabilidade:

**Dado que** eu sou um usuário idoso  
**quando** eu acessar o site **então espero**  
que a fonte não seja muito pequena.

**Agradeço**  
a sua atenção!

**Fábio Figueredo**

fabio.figueredo@sptech.school

SÃO  
PAULO  
TECH  
SCHOOL