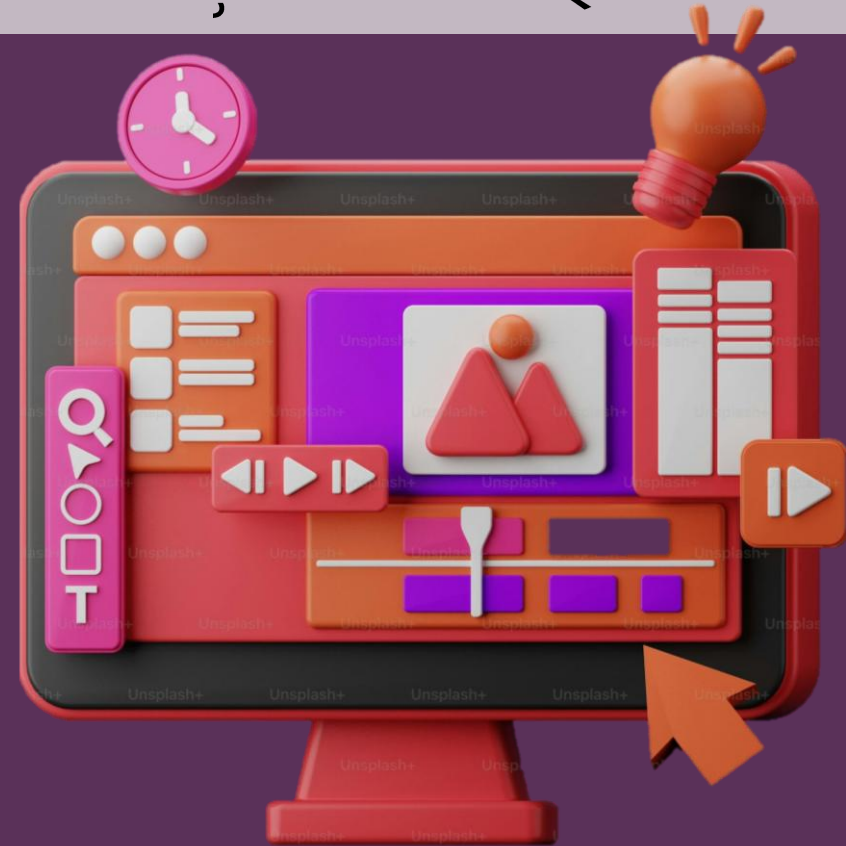


ENGENHARIA DE SOFTWARE

AULA 5 – DEFINIÇÃO DE ESCOPO E ESPECIFICAÇÃO DE REQUISITOS



WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR

Aula de Hoje

- Discutir os principais pontos do TAP (Termo de Abertura do Projeto) e o escopo do software.
- Compreender os elementos que compõe a especificação de requisitos.
- Verificar em qual situação deve ser utilizado um Estudo de Viabilidade.

**Por onde começamos a
documentação de
projetos de software?**

A resposta mais comum: TAP

- O Termo de Abertura do Projeto (TAP) ou Project Charter é o documento que **autoriza** formalmente o projeto.
- Ele designa o **gerente do projeto** e concede a ele a autoridade para utilizar os recursos da organização na execução das atividades do projeto.

O ChatGPT também sugere começar por ele

1. Visão Geral do Projeto

- **Project Charter:** Comece pelo Project Charter, que autoriza e define os aspectos básicos do projeto, como objetivo, escopo, e stakeholders.
- **Descrição do Projeto:** Forneça um resumo do projeto, explicando o que será desenvolvido, qual problema será resolvido e os benefícios esperados.

2. Definição de Requisitos, 3. Especificação de Escopo, 4. Estrutura Analítica do Projeto (EAP), 5. Arquitetura de Software, 6. Plano de Gerenciamento do Projeto, 7. Plano de Testes, 8. Documentação Técnica, 9. Documentação do Usuário, 10. Revisão e Aprovação

Quando usar um TAP?

- O TAP permite definir claramente os objetivos do projeto e quais as suas **fronteiras**.
- Define também o âmbito do projeto bem como o **produto final**.
- Nele são documentados **premissas e riscos** de alto nível.
- O TAP é emitido preferencialmente pelo gerente de projeto junto com sua equipe e com os stakeholders.





**O TAP é um documento para quem?
Cliente ou Desenvolvedor?**

O TAP permite responder questões, como:

1. O que deve ser feito para atingir o objetivo do projeto?
2. Como deve ser feito?
3. Quem vai fazer?
4. Quando deve ser feito?

TAP deve abordar as seguintes questões:

- Requisitos (do usuário) que satisfazem as necessidades do cliente.
- **Objetivos do projeto (que devem ser SMART).**
- Propósito ou justificação do projeto.
- Stakeholders, seus papéis e responsabilidades.
- Expectativas dos stakeholders.
- Identificação do gestor do projeto e nível de autoridade do gerente.
- Identificação e cronograma dos marcos do projeto.
- Premissas ou pressupostos organizacionais (fatores considerados verdadeiros, reais ou certos).
- Restrições organizacionais (fatores que limitam as opções da equipe).
- Investimento (orçamento preliminar – caso tenha).
- Constrangimentos e riscos (em alto nível).
- Descrição do(s) subproduto(s) identificado(s).

Objetivo **SMART**

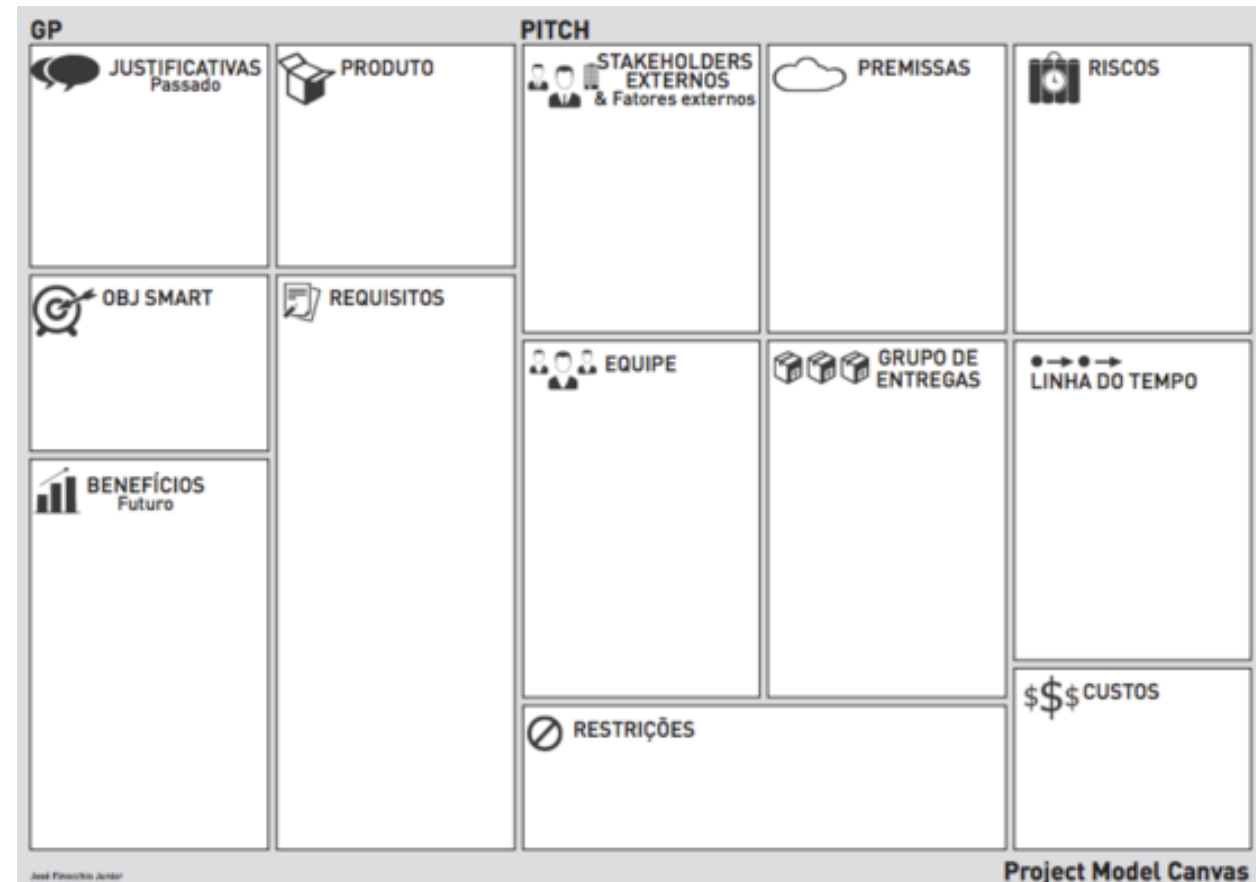
- **S** - Específicos (*Specific*): os objetivos devem ser formulados de forma específica e precisa.
- **M** - Mensuráveis (*Measurable*): os objetivos devem ser definidos de forma a poderem ser medidos e analisados em termos de valores ou volumes.
- **A** - Atingíveis (*Attainable*): a possibilidade de concretização dos objetivos deve estar presente, estes devem ser alcançáveis.
- **R** - Realistas (*Realistic*): os objetivos não pretendem alcançar fins superiores aos que os meios permitem.
- **T** - Temporizáveis (*Time-bound*): os objetivos devem ser definidos em termos de duração.

Como fazer o TAP?

Algumas ferramentas podem auxiliar o processo de desenvolvimento do TAP:

PMCanvas

<http://www.projectmodelcanvas.com/>





Precisamos
conversar
sobre
Escopo e
não Escopo

Escopo do Software

Engenharia de Software, Roger Pressman. Cap. 3 – Seção 3.3

- Escopo de Software é o termo que define a **abrangência das funcionalidades** de um projeto (aplicativo), delineando o que ele deverá ou não atender (e de que forma deverá atender) no que tange às **necessidades** do usuário final do software.
- A gestão do escopo, durante um projeto de software, deve se fixar sobre a questão das **funcionalidades** desenvolvidas para o aplicativo, ou seja, o **escopo funcional** a ser desenvolvido.

Escopo do Software

APRESENTAR FATORES MITIGANTES NÃO SIGNIFICA QUE VOCÊ DEVE DESCREVER O DOCUMENTO DE RISCOS NO TAP.

- Normalmente, a definição do escopo é uma das primeiras atividades da gestão de um projeto de software (o TAP contém uma seção para introduzir o Escopo).
- Escopo deve ser **claro** tanto em nível técnico, quanto administrativo.
- Deve ser **delimitado** – números devem ser declarados, limitações devem ser anotadas e **fatores mitigantes*** devem ser descritos.

*mitigante

Tornar menos penoso, reduzir as consequências. A mitigação do risco é a redução (ou adequação) do risco a valores aceitáveis, sabendo-se que no que se refere à mitigação, o que se deseja evitar não é a ocorrência do fator gerador de risco, mas sua consequência.

O escopo a seguir está correto? Da margem para criar alguma expectativa fora do escopo?

Sistema Web para Venda de Pacotes de Acesso a Internet

Definido como uma plataforma WEB para cadastro de clientes e venda de pacotes de conexão com a Internet. Oferecem ao consumidor, fornecedor e público em geral um ambiente agradável e dinâmico, interativo e intuitivo, seguro e disponível a qualquer tempo e lugar, aproximando-os e facilitando bons negócios na área de abrangência de seus serviços.

Considerações sobre a Especificação do Escopo

- Antes de iniciar o desenvolvimento, o documento de escopo já deve estar definido.
- Deve conter todas as tarefas (em linhas gerais) que serão executadas para alcançar o objetivo: a criação do software.



Alice: Você pode me ajudar?

Gato: Sim, pois não.

Alice: Para onde vai esta estrada?

Gato: Para onde você quer ir?

Alice: Eu não sei, estou perdida.

Gato: Para quem não sabe onde vai,
qualquer caminho serve.

Alice no País das Maravilhas, Lewis Carroll

APÓS CONCLUIR O DOCUMENTO DE ESCOPO...

A still from the TV show 'The Big Bang Theory' featuring Sheldon Cooper. He is standing in a hallway, looking slightly to his left with a serious expression. He is holding a very thick stack of papers in front of him with both hands. He is wearing a dark blue button-down shirt over a white t-shirt. The background shows a hallway with a light fixture on the wall and a doorway.

É HORA DE
ESPECIFICAR OS
REQUISITOS?!

Atividade Prática 1

INDIVIDUAL

Considere que você está desenvolvendo um **módulo de login** como um **produto reutilizável** para uma família de software.

Elenque:

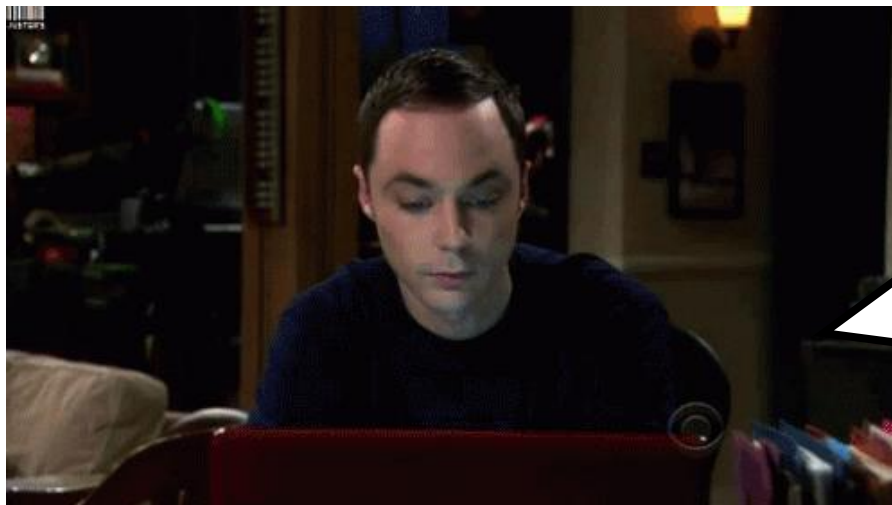
- 3 requisitos funcionais.
- 2 requisitos não-funcionais.

Alguns Exemplos de Respostas que podem ser melhoradas

- NF: bom visual
- RF: senha com um numero mínimo de caracteres
- RF: botao de login
- RF: realizar login do usuário
- NF: formulário de login
- RF: usuário fazer login com email e senha
- NF: selecionar a tag verificação de login
- RF: Lembrar senha
- rf001. A tela de login deve exibir os seguintes campos: CPF; E-mail.
- RF002 se o usuário não tem conta haverá um botão cadastrar-se.
- RNF001. O Sistema deverá funcionar na plataforma WEB.
- RNF002. O Sistema deverá estar disponível depois de validação por um captcha.

Os seguintes Requisitos estão corretos?

- RF 01 – Manipulação de dados e de processamento.
- RF 02 – Cálculos.
- RF 03 – Requisitos de entrega. Ex.: um relatório de acompanhamento deverá ser fornecido toda segunda-feira.
- NF 01 – Requisitos de eficiência. Ex.: o sistema deverá processar n requisições por um determinado tempo.



OS REQUISITOS
ESTÃO INCOMPLETOS
OU NÃO SÃO
MENSURÁVEIS!

Especificação dos Requisitos

Engenharia de Software. Ian Sommerville. Cap. 6

- Requisitos de um sistema são **descrições** dos serviços fornecidos pelo sistema e as suas **restrições** operacionais.
- Refletem as **necessidades dos clientes**.
- O processo de: descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de **Engenharia de Requisitos**.

Requisitos

- O termo não é utilizado de forma consistente.
- Em alguns casos é simplesmente uma **descrição em alto nível** de um serviço que o sistema deve fornecer ou uma restrição.
- Em outro extremo é uma **definição formal e detalhada** de uma função do sistema.

Requisitos

Como devemos proceder na hora de documentar?

Com muitos detalhes? (compreensível apenas para a equipe de desenvolvimento)

Ou em alto nível? (compreensível por todos os stakeholders)

Davis (1993) sugere que...

- As necessidades devem ser definidas inicialmente de **forma suficientemente abstrata** para ser compreendido por todos os stakeholders. Dessa forma, é possível existir uma contribuição maior por parte deles no processo descrito.
- Após aprovação do contrato, o **fornecedor deve redigir uma definição mais detalhada** do sistema para o cliente.
- Os dois documentos (o mais abstrato e mais detalhado) irão gerar o **documentos de requisitos do sistema**.

Requisitos do Usuário e do Sistema

O nível de abstração dos requisitos pode acarretar problemas no processo de implementação, por isso, Sommerville (2011) faz distinção entre os Requisitos do Usuário e Requisitos do Sistema.

ALTO NÍVEL DE
GENERALIZAÇÃO

**REGRA DE
NEGÓCIO**

- ESCOPO DO PROJETO
- OBJETIVOS DE NEGÓCIO
- SOLICITAÇÕES DO PATROCINADOR

**REQUISITOS DO
USUÁRIO**

- AMBIENTE DE USO
- NECESSIDADES DO USUÁRIO
- PROCESSOS DO USUÁRIO

**REQUISITOS DE
SISTEMA**

- REQUISITO FUNCIONAL
- REQUISITO NÃO FUNCIONAL

DETALHAMENTO
ESPECÍFICO



Requisitos do Usuário e do Sistema

- **Requisitos do Usuário:** declarações em linguagem natural (com diagramas) de quais serviços são esperados do sistema e as restrições que ele deve operar.
- **Requisitos do Sistema:** definem (detalhadamente) as funções, os serviços e as restrições operacionais do sistema. As vezes este documento é chamado de **Especificação Funcional**.

Um requisito do Usuário pode ser tornar **vários** requisitos do Sistema

Definição de requisitos do usuário

1. LIBSYS deve manter o acompanhamento de todos os dados exigidos pelas agências de licenciamento de direitos autorais no Reino unido e em outros lugares.

Especificação dos requisitos do sistema

1.1. Ao solicitar um documento ao LIBSYS, deve ser apresentado ao solicitante um formulário que registra os detalhes do usuário e da solicitação feita.

Especificação dos requisitos do sistema

1.2. Os formulários de solicitação do LIBSYS devem ser armazenados o sistema durante cinco anos, a partir da data da solicitação;

Especificação dos requisitos do sistema

1.3 Todos os formulários do LIBSYS devem ser indexados por usuário, nome do material solicitado e fornecedor da solicitação.

Especificação dos requisitos do sistema

1.4 O LIBSYS deve manter um registro de todas as solicitações feitas ao sistema;

Requisitos do Sistema

Os requisitos do sistema podem ser classificados em:

- **Funcionais:** declarações de serviços que o sistema deve oferecer. Como o sistema deve se comportar ante entradas específicas e determinadas situações.
- **Não-Funcionais:** possíveis restrições sobre os serviços que o sistema oferece. Inclui restrições de timing, sobre processo de desenvolvimento e uso de padrões. Se aplica ao sistema como um todo.
- **Domínio:** requisitos provenientes do domínio da aplicação do sistema e que refletem as características e restrições desse domínio.

Normas

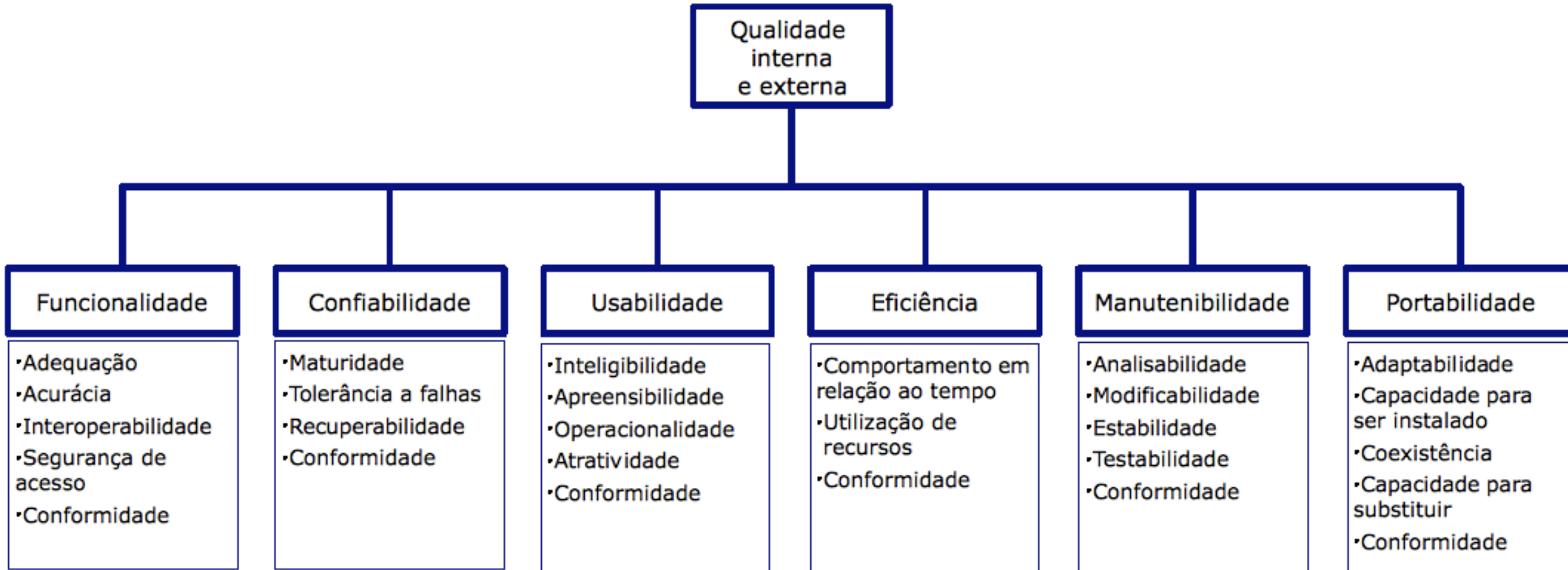
<https://www.abntcatalogo.com.br/>

- ISO/EIC 9126-1: cancelada
- ISO/EIC 25030: em vigor
- <https://www.sis.se/api/document/preview/908994/>

ISO/IEC 25030 is part of the ISO/IEC 25000 SQuaRE series of standards. The series consists of the following divisions under the general title *Software product Quality Requirements and Evaluation (SQuaRE)*:

- ISO/IEC 2500n, *Quality Management Division*,
- ISO/IEC 2501n, *Quality Model Division*,
- ISO/IEC 2502n, *Quality Measurement Division*,
- ISO/IEC 2503n, *Quality Requirements Division*, and
- ISO/IEC 2504n, *Quality Evaluation Division*.

ISO 9126-1

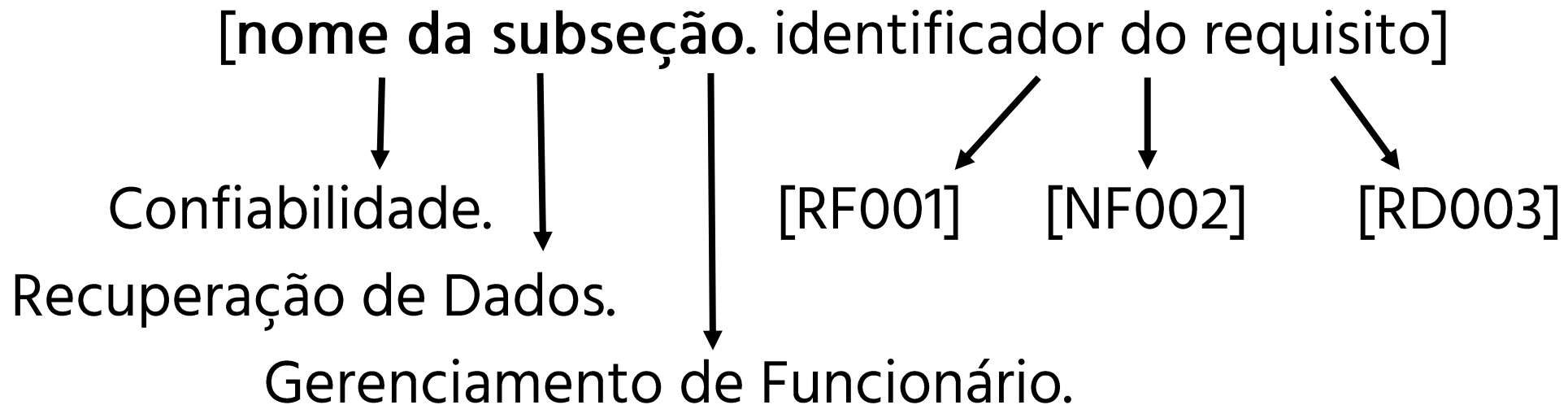




**Todos os requisitos
precisam ser verificáveis!**

Documentando Requisitos

Por convenção, a referência a requisitos é feita através do **nome da subseção** onde eles estão descritos (Requisito Funcional, por exemplo), seguidos do identificador do requisito (um número), de acordo com a especificação a seguir:



Por exemplo...

- O requisito funcional [Recuperação de dados.RF016] deve estar descrito em uma **subseção** chamada “Recuperação de dados”, em um bloco identificado pelo número [RF016].
- Já o requisito não-funcional [Confiabilidade.NF008] deve estar descrito na **subseção de requisitos não-funcionais de Confiabilidade**, em um bloco identificado por [NF008].
- Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] ou [NF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

Guia para Elaboração de Requisitos

- Definir um formato padrão e usar para descrever todos os Requisitos.
- Utilizar o idioma de forma consistente. Usar “**deve**” para requisitos **obrigatórios**, “**deveria**” para requisitos **desejáveis**.
- Usar texto **destacado** para identificar as partes principais do requisito.
- Evitar o uso do jargão de computação: Manter, upar, instanciar, persistir, dropar, buildar.

Requisitos Funcionais devem possuir:

1. Descrição do requisito – Em linguagem natural.
2. Prioridade – Essencial (ou obrigatória, alta, máxima), importante (média) ou Desejável (baixa, mínima).
3. Entradas e pré-condições – necessárias ao requisito.
4. Saídas e pós-condições – após a execução do requisito.

Exemplo de Requisito Funcional

3.4 Importação/Exportação

[RF001] Anexar documentos

Descrição do caso de uso: Este caso de uso permite que anexar documentos gerais a componentes. Por exemplo, anexar o *template* do *Documento de Requisitos* ao fluxo de requisitos.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

Entradas e pré-condições: deve receber como entrada o caminho absoluto para um arquivo no sistema de arquivos.

Saídas e pós-condição: O documento é anexado ao componente.

Requisitos Não Funcionais devem possuir:

1. A descrição do requisito logo abaixo do título do requisito.
2. A prioridade deve ser marcada.

[NF002] Desempenho

Embora não seja um requisito essencial ao sistema, deve ser considerada por corresponder a um fator de qualidade de software.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

Prioridade de Requisitos

As prioridades são descritas em três níveis: “essencial”, “importante” e “desejável”, por exemplo.

4. Requisitos não-funcionais

[NF001] Usabilidade

A interface com o usuário é de vital importância para o sucesso do sistema. Principalmente por ser um sistema que não será utilizado diariamente, o usuário não possui tempo disponível para aprender como utilizar o sistema.

O sistema terá uma interface amigável ao usuário primário sem se tornar cansativa aos usuários mais experientes. Em especial, o módulo de publicação HTML possuirá um wizard para ajudar o usuário.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

MAS O QUE
REALMENTE
SIGNIFICA CADA UMA
DAS PRIORIDADES?



Prioridade Máxima, Obrigatória, Alta...

- É o requisito sem o qual o sistema não entra em funcionamento.
- São requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

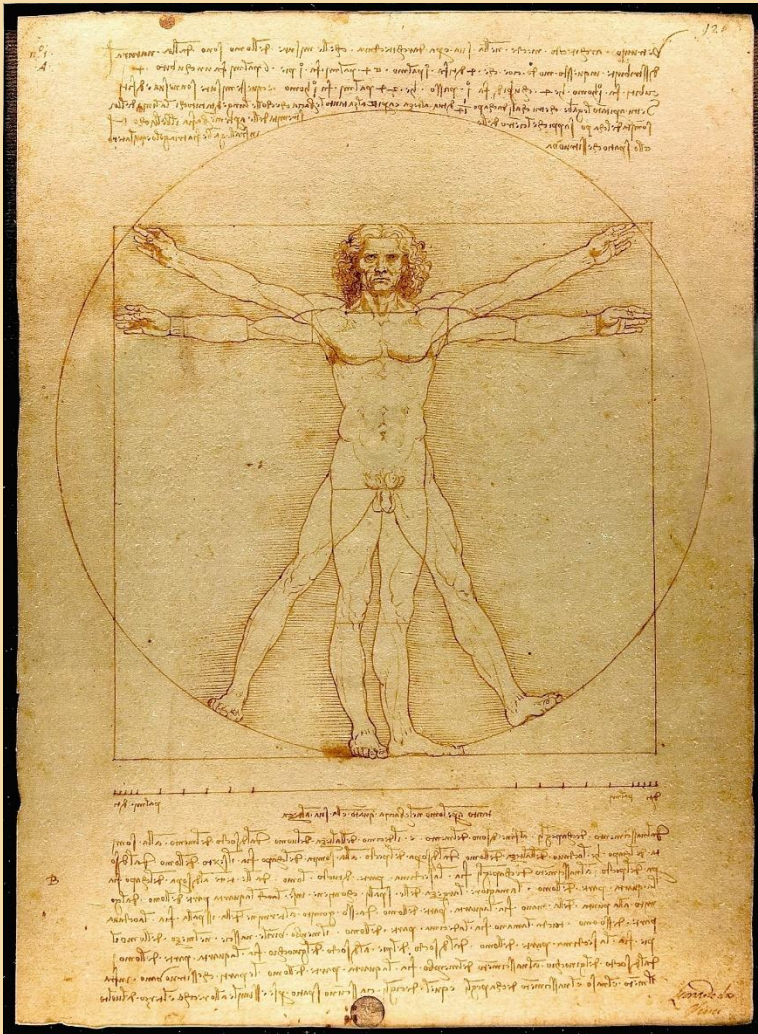
Prioridade Importante, Média..

- É o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória.
- Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado.

Prioridade Desejável, Baixa, Mínima...

- É o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele.
- Estes requisitos podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

Compare os requisitos com partes do corpo humano...



1. O corpo humano funciona sem as unhas? Dentes?
2. O corpo humano funciona sem um braço? Perna?
3. O corpo humano funciona sem um coração? Cérebro?

1. Prioridade: Baixa (unhas, dentes)
2. Prioridade: Média (braço, perna)
3. Prioridade: Alta (coração, cérebro)

MOSCOW

- Must Have (Tenho que fazer)
- Should Have (Devo fazer)
- Could Have (Poderia fazer)
- Won't Have (Não vou fazer)

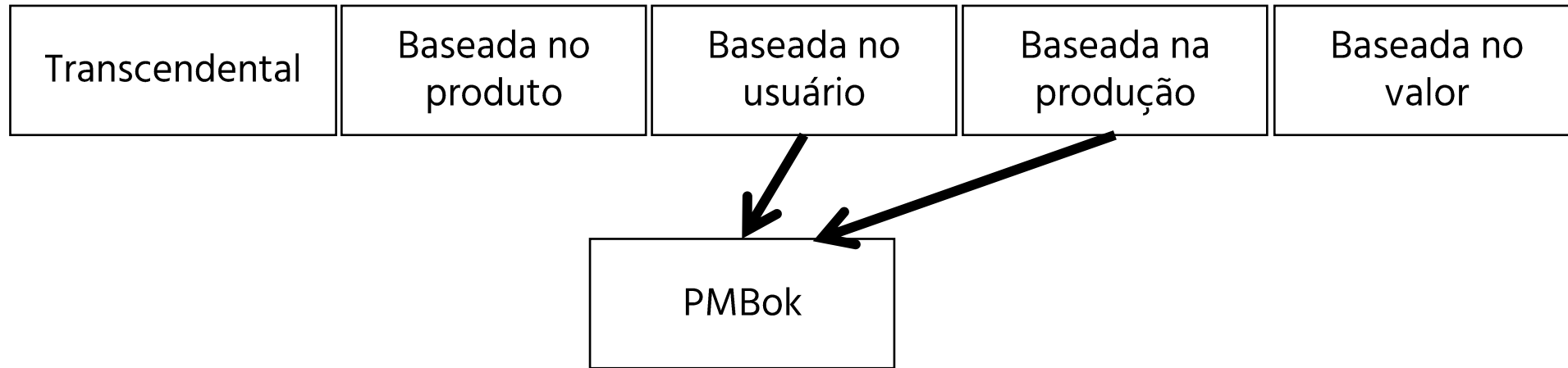
Exemplo da caneta Bic!





A SATISFAÇÃO TOTAL
DO CLIENTE ESTÁ
LIGADA A QUAL DAS
TRÊS PRIORIDADES
DE REQUISITOS?

Definição de Qualidade (Garvin, 1988)



- **Baseado no Usuário:** “O grau até o qual um conjunto de características inerentes **satisfaz** as necessidades do usuário”;
- **Baseado na Produção:** “Um projeto com **qualidade** é aquele concluído em conformidade com os requisitos, especificações e adequação ao uso”.

Qualidade Transcendental

- A qualidade é vista como sinônimo de perfeição e de excelência, não necessitando de complemento algum para qualificação de um produto ao chegar nesse estágio.
- Em casos como esse percebe-se que essa visão de qualidade não é passível de crítica e que a experiência é o único requisito capaz de qualificar alguém a fazer tal julgamento.
- É como se conceituar qualidade fosse além das capacidades intelectuais humanas, mas **que quando a vê a reconhece**.

Qualidade Baseada no Produto

- Quando se trata de verificar a qualidade dos produtos, ela é considerada uma variável mensurável e precisa. As diferenças podem ser notadas entre produtos que, obviamente, sejam similares ou que estejam dentro de uma mesma classificação de uso, por exemplo, caso contrário seria injusto.
- Essa classificação confere uma cadeia hierárquica entre produtos sob o critério da qualidade. Alguns critérios podem ser utilizados para auxiliar na classificação dos produtos, como: durabilidade e resistência.
- Porém, quando o critério passa a ser subjetivo, como estética, a classificação pode ser comprometida pela dificuldade de mensuração.

Qualidade Baseada no Usuário

- A premissa básica da abordagem baseada no usuário é que a qualidade está naquele **produto que melhor atende às necessidades e expectativas dos consumidores no momento que eles desejam.**
- Igualmente à abordagem transcendental, esse conceito também reflete uma visão subjetiva da qualidade e enfrenta dois problemas: como agregar ao produto preferências individuais altamente variáveis de forma que o mercado perceba a qualidade nele inserida e como distinguir atributos de um produto que sinalizem a qualidade percebida pelos consumidores.
- A abordagem baseada no usuário iguala qualidade com satisfação máxima, o que não reflete a realidade.

Qualidade Baseada na Produção

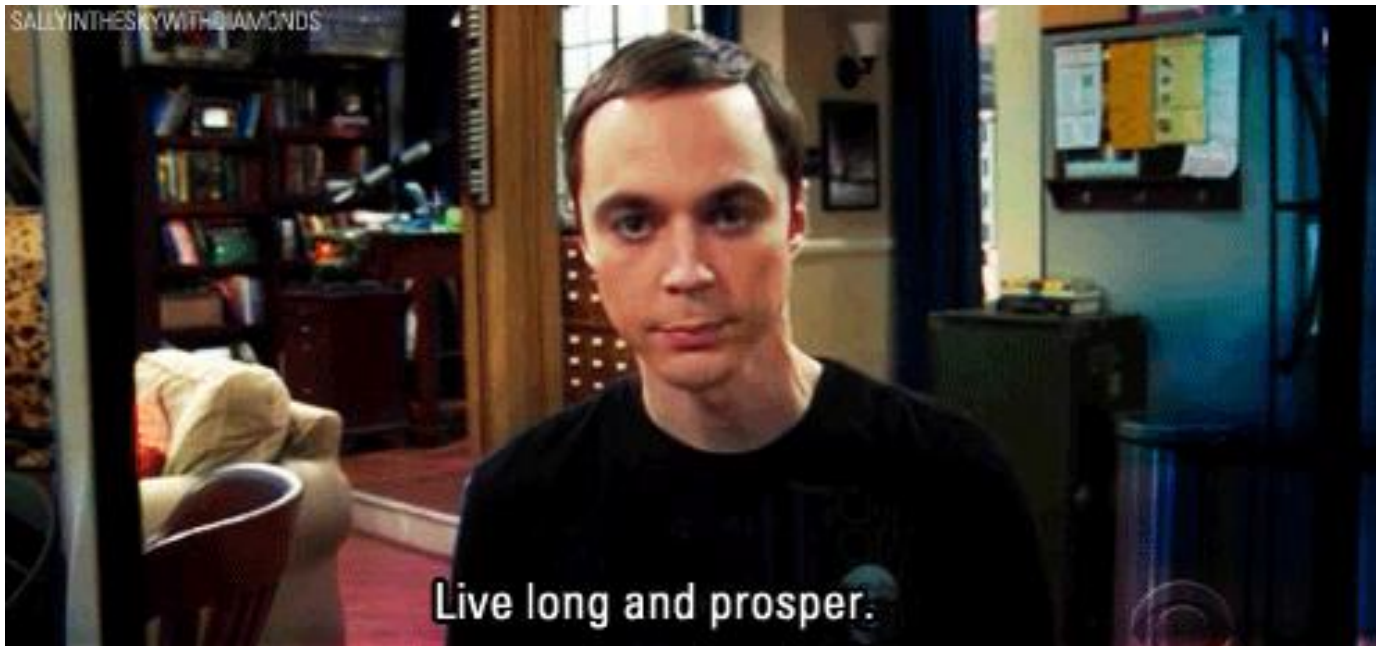
- Essa abordagem relaciona qualidade com engenharia e produção, ou seja, **um produto com qualidade é aquele que não sofreu modificação desde o seu projeto**. A partir dessa premissa surgiu o conceito de “fazer certo da primeira vez”.
- Nesses termos, um Mercedes pode ser um carro de qualidade como um Fiat Palio. O enfoque dessa abordagem é interno, porque supõe que um produto ou serviço que se desvie das especificações provavelmente será mal produzido e não será confiável, proporcionando uma satisfação menor ao consumidor.
- Essa abordagem está vinculada à redução de custos, pois a melhoria da qualidade (diminuição do número de erros) leva a menores custos: impedir defeitos é comprovadamente mais barato que corrigir falhas.

Qualidade Baseada no Valor

- A visão dessa abordagem **define qualidade em termos de custo e preço**. Dessa forma, **um produto de qualidade oferece desempenho equitativo ao seu preço** ou ao seu custo de forma aceitável pelo mercado.
- Pelo fato de trabalhar com dois conceitos relacionados, mas distintos, essa abordagem coloca-se a frente das demais em relação à questão subjetiva da conceituação da qualidade, porém, é de difícil aplicação, pois seus limites não são bem definidos além de depender da variabilidade das necessidades de cada consumidor.

Qualidade x Prioridade de Requisitos

Para o produto ter qualidade ele deve atender todos os requisitos apresentados pelos stakeholders... e ir além!



**Já definimos o escopo, os
requisitos, como fazer o produto...
Mas será que este será um
produto útil para o cliente?**

É neste cenário que entra o estudo de viabilidade!

Estudo de Viabilidade

Engenharia de Software. Ian Sommerville. Cap. 7

- Em todos os sistemas novos, o processo de engenharia de requisitos **deve começar** com um **estudo de viabilidade** (segundo Ian Sommerville).
- O relatório produzido deve responder principalmente, a esta pergunta:
- O sistema é útil para empresa?

O Estudo de Viabilidade deve gerar um relatório

As entradas para o Estudo de Viabilidade são:

- Conjunto **preliminar** de **requisitos** de negócio.
- **Esboço** da descrição do sistema.
- Como o sistema pretende **apoiar** os processos de negócio.

Este estudo responde também estas perguntas:

- O sistema **contribui** para os objetivos gerais da organização?
- O sistema pode ser **implementado com tecnologia atual** e dentro das restrições definidas de custo e prazo?
- O sistema pode ser **integrado a outros sistemas** já implantados?

Se o sistema não apoia os objetivos da empresa, ele tem algum valor real para ela?



Outras questões que o Estudo de Viabilidade deve responder

1. Como a organização se comportaria se o sistema não fosse implantado?
2. Quais os problemas com os processos atuais e como o sistema ajudaria a reduzir esses problemas?
3. Qual será a contribuição direta do sistema para os objetivos e requisitos da empresa?
4. As informações podem ser transferidas e recebidas de outros sistemas da organização?
5. O sistema requer tecnologia que ainda não foi utilizada na organização?
6. O que deve ser apoiado pelo sistema e o que não precisa ser apoiado?

Considerações sobre o Estudo de Viabilidade

- Deve ser feito ao final do estudo de viabilidade uma recomendação para a pergunta:

O desenvolvimento do sistema deve ou não prosseguir?

- Com base neste estudo, é possível:
 - Propor mudanças do escopo, orçamento e prazo.
 - Sugerir requisitos de alto nível adicionais para o sistema.

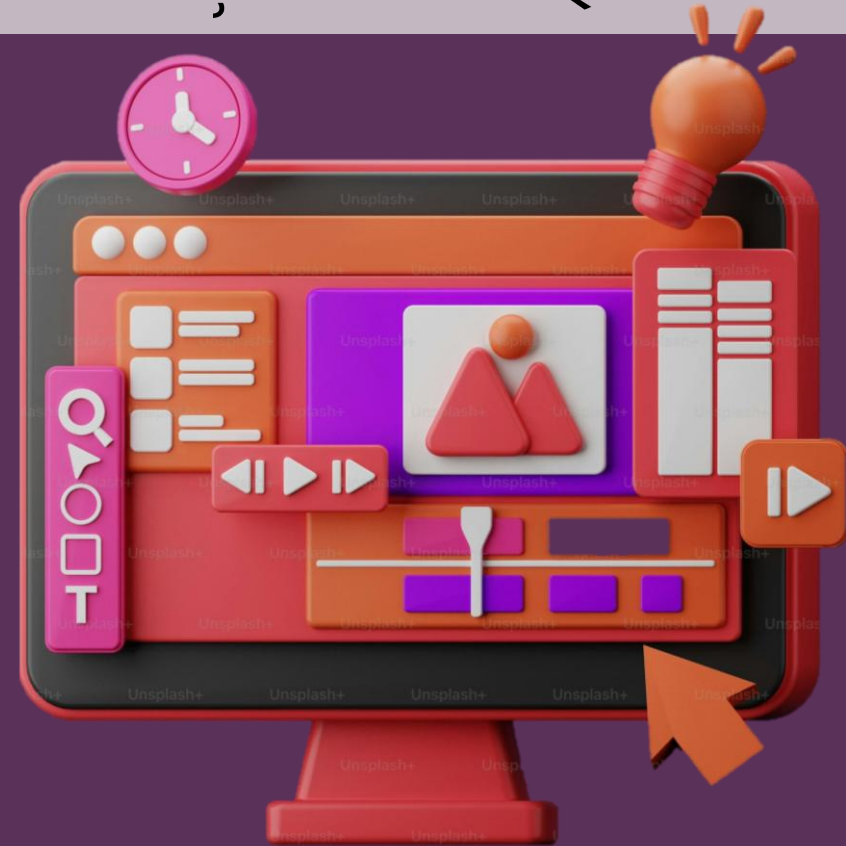
Lista de Exercícios 4

Referências

- PRESSMAN, R. Engenharia de Software. 7. ed. 2011 – Capítulo 5.
- SOMMERVILLE, I. Engenharia de Software. 10. ed. 2018 – Capítulo 4.
- VAZQUEZ, C. E.; SIMÕES, G. S. Engenharia de Requisitos: software orientado a negócios. 2016.

ENGENHARIA DE SOFTWARE

AULA 5 – DEFINIÇÃO DE ESCOPO E ESPECIFICAÇÃO DE REQUISITOS



WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR