

ENGENHARIA DE SOFTWARE 2 – AULA 3

PROF^a M^a DENILCE VELOSO
DENILCE.VELOSO@FATEC.SP.GOV.BR
DENILCE@GMAIL.COM

RETOMANDO ...

- Aula 1:
- Conceitos de Engenharia de Software
- Atividade 1 – Mapa Mental Processo Software
- Formação Grupos
- Aula 2:
- Conceitos Abordagens Ágeis
- Atividade 2 – Mapa Mental Scrum e RUP
- Discussão sobre Proposta Projeto

ENGENHARIA DE SOFTWARE – DIVERSIDADE NO PROCESSO DE SOFTWARE

- **Métodos, ferramentas e técnicas** específicas dependem da organização que está desenvolvendo o software, do tipo de software e das pessoas envolvidas no desenvolvimento. Não há métodos universais de engenharia de software que sejam adequados para todos os sistemas e empresa. (Sommerville, 2019 p. 10)



ENGENHARIA DE SOFTWARE - MÉTODOS

- Fornecem as informações técnicas para desenvolver softwares (instruções de trabalho).
- Conjuntos de tarefas com técnicas particulares para cada fase do desenvolvimento de software.
- Os métodos envolvem uma ampla variedade de tarefas, que incluem: comunicação, análise de requisitos, modelagem de projeto, construção de programa, testes e suporte.



ENGENHARIA DE SOFTWARE - FERRAMENTAS

- As ferramentas da engenharia de software **fornece suporte automatizado ou semiautomatizado para o processo e para os métodos.** Quando as ferramentas são integradas, de modo que as informações criadas por uma ferramenta possam ser utilizadas por outra, é estabelecido um sistema para o suporte ao desenvolvimento de software, denominado engenharia de software com o auxílio do computador (Ferramentas CASE – *Computer-Aided Software Engineering*).



ENGENHARIA DE SOFTWARE - TÉCNICAS

- As técnicas/procedimentos da engenharia de software organizam as maneiras que o desenvolvimento de software é realizado, pois eles definem a sequência em que os métodos devem ser aplicados, os produtos que devem ser desenvolvidos (relatórios, formulários, documentos, etc) os controles que devem ser aplicados para que a qualidade do software seja garantida, as formas de como as mudanças devem ser feitas e os marcos de referência para que o progresso de desenvolvimento do software possa ser avaliado.

*** Joint Application Development (JAD), prototipação, entrevista, questionário, observação, Implantação da Função de Qualidade (IFQ), casos de uso e pontos de vista*



ENGENHARIA DE SOFTWARE

Apenas um
Exemplo

- O software será desenvolvido utilizando a abordagem ágil SCRUM (**método**), para fazer o gerenciamento do projeto será empregado o JIRA (**ferramenta**) e será aplicada a técnica de Testes de Unidade para garantir a qualidade do código (**técnica**).

***Uma unidade é conhecida como o menor componente possível de software que pode ser testado.*



Slide 7

DV0

O teste de unidade envolve o teste de componentes individuais do programa de software ou aplicativo. O principal objetivo por trás disso é verificar se todas as peças individuais estão funcionando conforme o esperado. Uma unidade é conhecida como o menor componente possível de software que pode ser testado.

DENILCE DE ALMEIDA OLIVEIRA VEL; 2023-03-09T21:01:07.681

ENGENHARIA DE SOFTWARE – ATIVIDADES DO PROCESSO DE SOFTWARE

Apesar da diversidade, todo Processo de Software devem incluir de alguma forma as quatro atividades fundamentais:

1. Especificação - Definição do que o sistema deve fazer e suas restrições.
2. Desenvolvimento - Construção do sistema, incluindo design e implementação.
3. Validação – Verificação e testes para garantir que o software atende aos requisitos.
4. Evolução - Manutenção e atualização do software para atender novas demandas.

** incluem subatividades - Fonte: (Sommerville, 2019 p.30)

ENGENHARIA DE SOFTWARE – ATIVIDADES DO PROCESSO DE SOFTWARE

Pressman (2023) detalha o processo de software em **cinco fases principais**:

- 1. Comunicação** – Envolve levantamento de requisitos e interação com os stakeholders.
- 2. Planejamento** – Definição de cronogramas, recursos e estimativas de custos.
- 3. Modelagem** – Abrange análise de requisitos e design do sistema.
- 4. Construção** – Implementação e testes do software.
- 5. Implantação** – Entrega ao cliente, manutenção e suporte.



ENGENHARIA DE SOFTWARE - ESPECIFICAÇÃO



- Nessa etapa o sistema é **especificado** em termos técnicos, ou seja, desenvolve-se os **requisitos** de sistema que devem atender os **requisitos** de usuário. Na **especificação** passa-se da perspectiva do problema (**requisitos** de usuário) para a perspectiva da solução (**requisitos** de sistema).

ENGENHARIA DE REQUISITOS – O QUE É UM REQUISITO?

Os **requisitos** são a descrição do sistema, **funções e restrições** que são gerados durante o processo de engenharia de requisitos. (Sommerville, 2019)

Tanto Machado (2013) como Bezerra (2007) apontam que **os requisitos** são o pontapé inicial para a definição do escopo de um sistema.



ENGENHARIA DE REQUISITOS – O QUE É UM REQUISITO?

- Um requisito pode ser visto como **uma declaração abstrata de alto-nível**, uma função que o sistema deve fornecer ou uma restrição do sistema.
- **No outro extremo ele pode ser visto como uma especificação detalhada**, matematicamente formal de uma função do sistema.



PARA DESCONTRAIR ... O CLÁSSICO.



Como o cliente explicou...



Como o líder de projeto entendeu...



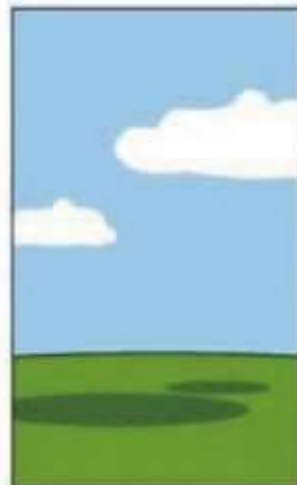
Como o analista projetou...



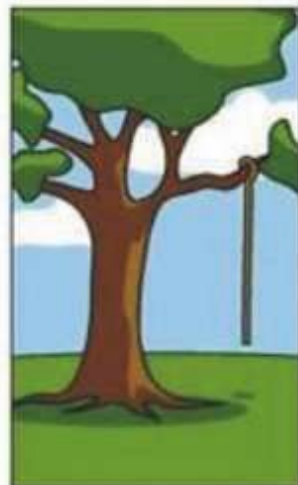
Como o programador construiu...



Como o consultor de negócios descreveu...



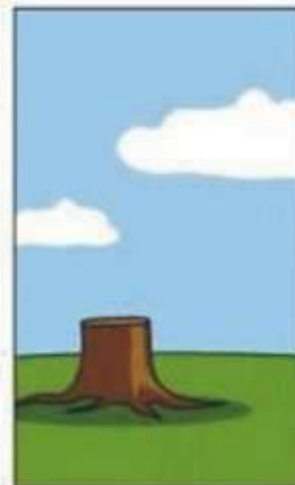
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

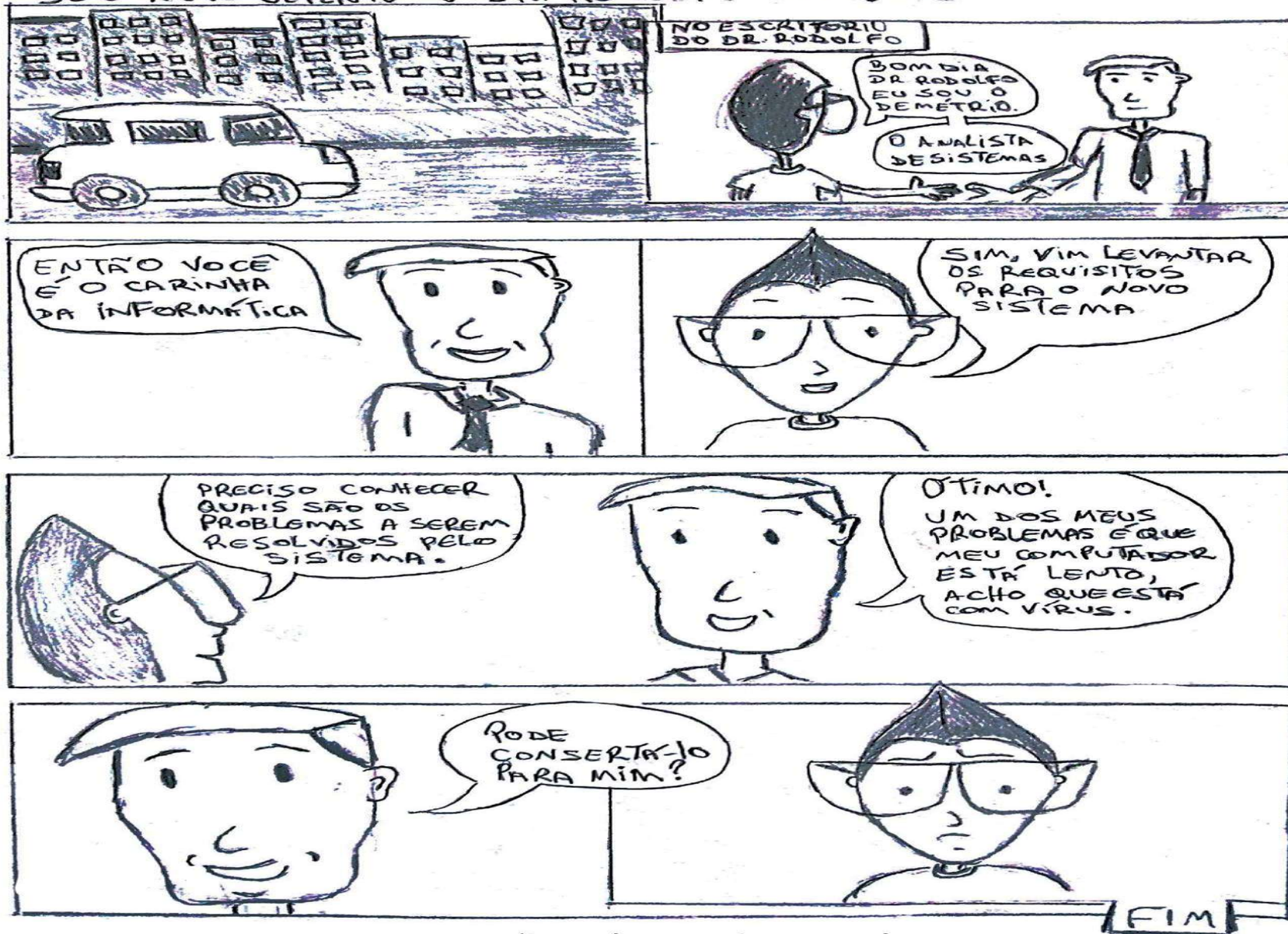
PARA DESCONTRAIR ...



Fonte: <https://www.leanti.com.br/artigos/16/gerenciamento-de-requisitos.aspx>

LEVANTAMENTO DE REQUISITOS

DEMÉTRIO IRÁ VISITAR PELA PRIMEIRA VEZ
SEU NOVO CLIENTE - O DR. RODOLFO - ADVOGADO



ABSTRAÇÃO DE REQUISITOS (DAVIS, 1993)

"Se uma empresa deseja estabelecer um contrato para desenvolvimento de um grande projeto de software, ela tem de definir suas necessidades de maneira suficientemente abstrata para não haver uma solução predefinida. Os requisitos devem ser escritos de modo que os vários concorrentes possam disputar o contrato, oferecendo, talvez, maneiras diferentes de satisfazer às necessidades da empresa.

Depois de assinado o contrato, o contratado deve uma definição mais detalhada do sistema para o cliente, de modo que ele entenda e valide o que o software fará. Esses dois documentos podem ser reunidos em documento de requisitos do sistema."

DAVIS, A. M. *Software requirements: objects, functions and states*. 2nd ed., Upper Saddle River, NJ: Prentice Hall, 1993. 521p.

TIPOS DE REQUISITOS

- Requisitos do usuário

- São declarações em **linguagem natural e também em diagramas**, sobre as **funções** que o sistema deve fornecer e as **restrições** sobre as quais deve operar. Ex.: o sistema deve permitir consultar o estoque de um item, sistema deve emitir nota fiscal, sistema deve permitir listar funcionários.

- Requisitos de Sistema

- Estabelecem detalhadamente as funções e as restrições de sistema. O documento de requisitos de sistema, algumas vezes chamado de **especificação funcional**, deve ser preciso. Ele pode servir como um contrato entre o comprador do sistema e o desenvolvedor do software. Ex.: o usuário deve escolher a opção consultar estoque, o usuário deve escolher o item a ser consultado.

- Especificação de Projeto de Software

- É uma **descrição abstrata do projeto de software**, que é uma base para o projeto e a implementação mais detalhados. **Acrescenta mais detalhes à especificação de requisitos do sistema.** Ex. : o usuário deve escolher a opção consultar estoque, o usuário deve escolher o item a ser consultado. Caso o item a ser consultado não exista, deverá emitir uma mensagem ao usuário.



TIPOS DE REQUISITOS - EXEMPLOS

Definição de
Requisito de
usuário

O Sistema deve emitir
solicitação de compra de
material

Especificação de
Requisito de
sistema

1. O usuário deve solicitar a opção de solicitação de material na tela.
2. O usuário deve escolher o material através do código ou da descrição.
3. O usuário deve preencher a quantidade necessária.
4. O usuário deve escolher a data de necessidade do material.
5. Caso o usuário não preencha alguns dos campos, informar erro ao usuário.
6. Caso a data seja menor que a data atual, informar erro ao usuário.
7. Após a gravação da solicitação, a mesma deverá ficar disponível ao gerente de compras para aprovação.

TIPOS DE REQUISITOS - EXEMPLOS

Definição de Requisito de usuário

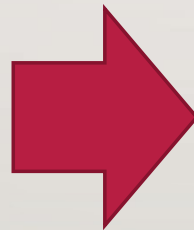
O Sistema Mentcare deve gerar relatórios de gestão mensais, mostrando o custo dos medicamentos prescritos por cada clínica naquele mês.

Especificação de Requisito de sistema

1. No último dia útil de cada mês, deve ser gerado um resumo dos medicamentos prescritos, seu custo e a clínica que os prescreve,
2. O sistema deve gerar o relatório para impressão após as 17h30 do último dia útil do mês.
3. Deve ser criado um relatório para cada clínica, listando o nome de cada medicamento, a quantidade total de prescrições, a quantidade de doses prescritas e o custo total dos medicamentos prescritos.
4. Se os medicamentos estiverem disponíveis em dosagens diferentes, devem ser criados relatórios diferentes para cada dosagem.
5. O acesso aos relatórios de medicamentos deve ser restrito aos usuários autorizados, conforme uma lista de controle de acesso produzida pela gestão.

TIPOS DE REQUISITOS - LEITORES

Requisitos de usuário



Gerentes do cliente
Usuários Finais do sistema
Engenheiros do cliente
Gerentes do contratado
Arquitetos de sistema

Requisitos de sistema



Usuários finais do sistema
Engenheiros do cliente
Arquitetos de sistema
Desenvolvedores de software

TIPOS DE REQUISITOS - STAKEHOLDERS

Os stakeholders do sistema (pessoas interessadas) incluem qualquer um que de alguma maneira seja afetado pelo sistema, ou tenha algum interesse nele. Podem variar de usuários finais de uma sistema, a gerentes e *stakeholders* externos, como autoridades reguladoras, que certificam a aceitabilidade do sistema.

Exemplo:

Em um sistema de nota fiscal, podem ser: os usuários que emitem a nota, os clientes que recebem a nota, os funcionários do departamento fiscal, contabilidade e a receita federal.

Em um sistema na área de saúde, pode ser: funcionários, médicos, enfermeiros, COREN, CRM, ANS.



ENGENHARIA DE REQUISITOS

É o processo de descobrir, analisar, documentar e verificar as funções e restrições que o usuário requer para o sistema. (Sommerville, 2019)

- Requisitos funcionais e não funcionais
- Processo engenharia de requisitos
- Elicitação de Requisitos
- Especificação de Requisitos
- Validação de Requisitos
- Mudança de Requisitos



REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS

- Os requisitos de sistema de software podem ser vistos como:
 - **Requisitos Funcionais**
 - São declarações de **funções que o sistema deve fornecer**, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer.
 - **Requisitos não-Funcionais**
 - São restrições sobre **os serviços ou as funções oferecidas pelo sistema**. Entre eles destacam-se restrições de tempo, restrições sobre o processo de desenvolvimento, padrões, entre outros.
 - **Requisitos de domínio**
 - São requisitos que se originam do domínio de aplicação do sistema e que refletem características desse domínio. **Podem ser requisitos funcionais e não funcionais**. (específicos da área saúde, educação, meio ambiente etc)

DV0

Se é área da saúde, telecomunicações, educação etc --> DOMÍNIO (necessidades, problemas, processos específicos)

DENILCE DE ALMEIDA OLIVEIRA VEL; 2023-03-16T16:02:41.709

REQUISITOS FUNCIONAIS

- Descrevem a funcionalidade ou os serviços que se **espera que o sistema forneça**.
- Dependem do tipo de software que está sendo desenvolvido, dos usuários de software e do tipo de sistema.
- Podem ser declarações de alto nível do que o sistema deve fazer.
- Devem descrever os serviços do sistema em detalhes.



REQUISITOS NÃO-FUNCIONAIS

- São aqueles que não dizem respeito, diretamente às funções específicas fornecidas pelo sistema. Eles estão relacionados a propriedades como confiabilidade, tempo de resposta, memória, legislação etc.
- Requisitos não funcionais de processo podem por exemplo solicitar o uso de uma determinada ferramenta CASE, linguagem de programação ou método de desenvolvimento.
- Requisitos não funcionais podem ser mais importantes que requisitos funcionais individuais pois a falha em não cumprir um requisito não funcional pode tornar o sistema inútil. Ex. *sistema de controle de tráfego aéreo falhe em atender ao requisito não funcional de alta disponibilidade.*

REQUISITOS NÃO-FUNCIONAIS - PROPRIEDADES EMERGENTES

As **propriedades emergentes** são uma consequência da relação entre os componentes do **sistema**. Elas não estão associadas a nenhuma parte do sistema mas estão lá. Só podem ser acessadas e medidas uma vez que os componentes estejam integrados no **sistema**. – Aparecem quando todas as partes de um **sistema** trabalham juntas para atingir algum objetivo. Exemplo: confiabilidade, disponibilidade, segurança, escalabilidade.

DV0



Slide 26

DV0

A escalabilidade do sistema é uma propriedade emergente. Ela não é inerente a nenhum componente individual, mas surge da forma como todos os componentes interagem e se coordenam.

DENILCE DE ALMEIDA OLIVEIRA VEL; 2024-08-29T15:55:49.773

DV0 0

Um exemplo genérico seria uma orquestra, as propriedade emergentes surgem quando todos tocam juntos.

DENILCE DE ALMEIDA OLIVEIRA VEL; 2024-08-29T15:57:06.041

CLASSIFICAÇÃO DOS REQUISITOS NÃO-FUNCIONAIS

- Requisitos de Produto

- São os requisitos que especificam o comportamento do produto, entre os exemplos estão os requisitos de desempenho sobre com que rapidez o sistema deve operar e quanta memória ele requer, os requisitos de confiabilidade, que estabelecem a taxa aceitável de falhas, os requisitos de portabilidade e os requisitos de facilidade de uso.

Exemplos: Toda entrada de dados deve ser padrão TXT, o sistema Mentcare deve estar disponível de 2ª a 6ª feira das 08h30 às 17h30. O tempo de inatividade dentro do horário normal de trabalho não deve exceder cinco segundos em qualquer dia.



CLASSIFICAÇÃO DOS REQUISITOS NÃO-FUNCIONAIS

- Requisitos Organizacionais
 - São procedentes de políticas e procedimentos nas organizações do cliente e do desenvolvedor. Entre os exemplos estão os padrões de processo que devem ser utilizados, os requisitos de implementação, como a linguagem de programação ou o método de projeto utilizado, e os requisitos de fornecimento (quando o produto e seus documentos devem ser entregues).

Exemplos: O processo de desenvolvimento e a documentação de todo o sistema devem ser entregues conforme norma padrão xxx.yyyy. Os usuários devem se autenticar usando sua carteira de identidade de autoridade de saúde (CRM, COREN).



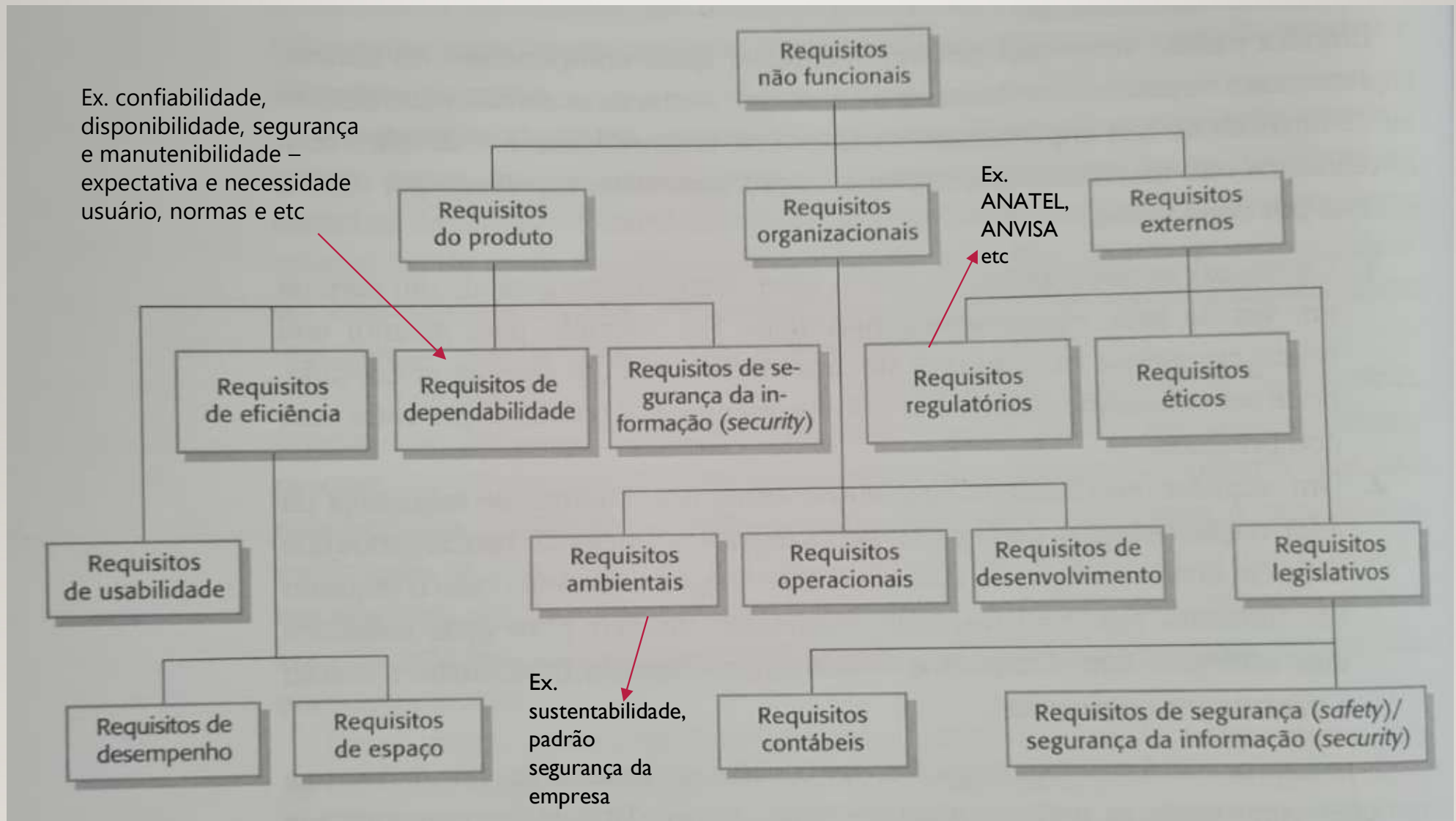
CLASSIFICAÇÃO DOS REQUISITOS NÃO-FUNCIONAIS

- Requisitos Externos
 - São requisitos procedentes de fatores externos tais como interoperabilidade, requisitos legais e éticos.

Exemplos: O sistema não deve revelar aos operadores nenhuma informação pessoal sobre os clientes além do nome e número de referência.



TIPOS DE REQUISITOS NÃO-FUNCIONAIS



DV0

Requisitos de dependabilidade são requisitos que especificam as propriedades de confiabilidade, disponibilidade, segurança e manutenibilidade de um sistema. Esses requisitos são essenciais para garantir que um sistema atenda às necessidades e expectativas dos usuários, bem como às normas e regulamentos aplicáveis.

DENILCE DE ALMEIDA OLIVEIRA VEL; 2023-03-16T16:04:10.780

MÉTRICAS PARA ESPECIFICAR REQUISITOS NÃO FUNCIONAIS

Propriedade	Métrica
Velocidade	Transações processadas/segundo
	Tempo de resposta do usuário/evento
	Tempo de atualização da tela
Tamanho	Megabytes/número de chips de ROM
Facilidade de uso	Tempo de treinamento
	Número de quadros de ajuda
Confiabilidade	Tempo médio até a falha
	Probabilidade de indisponibilidade
	Taxa de ocorrência de falhas
	Disponibilidade
Robustez	Tempo para reiniciar após a falha
	Porcentagem de eventos causando falhas
	Probabilidade de corromper dados em uma falha
Portabilidade	Porcentagem de declarações dependentes do sistema-alvo
	Número de sistemas-alvo

EXEMPLO PARA TEMPO DE RESPOSTA NOS REQUISITOS NÃO FUNCIONAIS

https://www.hostinger.com.br/tutoriais/core-web-vitals#Largest_Content_Paint_LCP_%E2%80%94_Tempo_de_Carregamento_da_Pagina

REQUISITOS NÃO-FUNCIONAIS - COMPLIANCE

- Palavra comum no dia a dia das organizações. Antigamente ela era relacionada com a área de direito e auditoria. Porém, com o crescimento dos diferentes softwares de apoio ao negócio, passa a ser uma preocupação dos analistas de requisitos que precisam levar em consideração ao longo de todo o ciclo de Gerência de Requisitos. (COMPLIANCE,2021)
- Compliance significa **estar conforme, que implica que as organizações devem satisfazer aos requisitos das regulamentações aplicáveis**. A conformidade legal impõe controles que focam na criação e retenção de informações, assim como proteção, integridade e disponibilidade, influenciando diretamente os processos e sistemas de informação das empresas. A não conformidade legal resulta em clientes ou investidores não satisfeitos, danos à imagem da empresa, penalizações e até em processos criminais (OTTO e ANTON, 2007).



ATIVIDADE 3

- Fazer download da planilha Exercicio_Aula3_Requisitos_NaoFuncionais e realizar o exercício solicitado.
- Subir/Upar o arquivo na sua área do GitHub na pasta Atividade3: SEUUSUARIO/ES2N/Atividade3

PRÓXIMOS PASSOS PARA O PROJETO INTEGRADOR - HOJE



Discussão sobre a Proposta do Projeto.

PRÓXIMOS PASSOS PARA O PROJETO INTEGRADOR – SEMANA QUE VEM



Terceira versão da proposta (formato doc, ES2N-Proposta v3.0, textos são informais)



REFERÊNCIAS

- BEZERRA, Eduardo. **Princípios de análise e projeto de sistema com UML**. Rio de Janeiro: Elsevier, 2007. 6ª reimpressão.
- COMPLIANCE. **Compliance: Lidando com os Requisitos Legais no Software**. Disponível em: <https://www.dheka.com.br/compliance-lidando-com-requisitos-legais-no-software/> Acesso em: 04 de mar. de 2021.
- GC. Disponível em: **Fundamentos de Engenharia de Software**. Disponível em: <https://www.grancursosonline.com.br/download-demonstrativo/download-apostila/codigo/kEYYuxcBxow%3D> Acesso em: 01 de mar. de 2021.
- MACHADO, Felipe Nery Rodrigues. **Análise e gestão de requisitos de software**: onde nascem os sistemas. 1ª. ed. São Paulo: Érica, 2011.
- PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: McGraw-Hill, 2011.
- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2019.
- _____. Materiais Fornecidos pelo autor. Disponível em: <<https://www.slideshare.net/software-engineering-book/ch4-req-eng>> Acesso em: 01 de mar. de 2021.
- OTTO, P.N.; ANTÓN A. I. **Addressing legal requirements in requirements engineering**. IEEE International Requirements Engineering Conference (RE'07). IEEE, Los Alamitos, CA, 5–14. DOI:10.1109/RE.2007.65 (2007).

