



ENGENHARIA DE REQUISITOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Prof. Renato Matroniani



EDUCAÇÃO
METODISTA

EMENTA

- Introdução à engenharia de requisitos, processos de software, modelos e atividades de processos, RUP, métodos ágeis, XP, SCRUM, introdução à modelagem de sistemas.
- Práticas de Engenharia de Requisitos: Requisitos funcionais e não funcionais, documento de requisitos, especificação de requisitos, processos de engenharia de requisitos.



Referências

- SOMMERVILLE, Ian. Engenharia de Software. 10ª ed. São Paulo: Pearson Prentice Hall, 2011. (Biblioteca Virtual) <https://plataforma.bvirtual.com.br/Leitor/Publicacao/168127/pdf/0>
- PRESSMAN, R. S. Engenharia de Software. Rio de Janeiro: McGraw Hill, 2002, 704p.
- SBROCCO, J. H.; MACEDO, P. C., Metodologias Ágeis: Engenharia de Software sob Medida., São Paulo: Érica, 2012, 256p.
- CHIKOFFSKY, Elliot. Computer-Aided Software Engineering (CASE). COMPUTER IEEE Computer Society, 1993.

Requisitos não-funcionais (cont.)

- Requisitos não-funcionais e metas gerais:

“O sistema deve ser de fácil uso pelo pessoal médico e deve ser organizado de tal maneira que os erros dos usuários sejam minimizados”

X

“A equipe médica deve ser capaz de usar todas as funções do sistema após quatro horas de treinamento. Após esse treinamento, o número médio de erros cometidos por usuários experientes não deve exceder dois por hora de uso do sistema”

Requisitos não-funcionais

- Qual a principal característica que diferencia as duas “metas”?
- Por que a primeira citação dá margem para erros de projeto de software?

Requisitos não-funcionais

- Sempre que possível, os requisitos não funcionais devem ser escritos quantitativamente, para que possam ser objetivamente testados.
- Qual a dificuldade de expressar metas em requisitos mensuráveis?
- Os custos de verificar requisitos objetivamente não funcionais mensuráveis podem ser justificados?

Requisitos não-funcionais

- Aqui estamos falando de percepção e entendimento divergente entre as pessoas que participam do projeto de software!
- Como evitar requisitos genéricos ou não mensuráveis?

Requisitos não-funcionais

Tabela 4.1 Métricas para especificar requisitos não funcionais.

Propriedade	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização de tela
Tamanho	Megabytes Número de chips de memória ROM
Facilidade de uso	Tempo de treinamento Número de <i>frames</i> de ajuda
Confiabilidade	Tempo médio para falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Percentual de eventos que causam falhas Probabilidade de corrupção de dados em caso de falha
Portabilidade	Percentual de declarações dependentes do sistema-alvo Número de sistemas-alvo

Ainda sobre diferenciar RFs e RNFs:

- Com frequência requisitos não funcionais conflitam e interagem com outros requisitos funcionais ou não funcionais.
- Na prática, o documento de requisitos apresenta dificuldade de separá-los.

Documento de requisitos de SW

- também chamado de Especificação de Requisitos de Software.
- *SRS – Software Requirements Specification.*
- declaração oficial do que os desenvolvedores do sistema devem implementar.
- deve incluir requisitos de usuário e requisitos do sistema.
- formato variável (depende do caso)

Documento de requisitos e métodos ágeis

- métodos ágeis de desenvolvimento mudam requisitos de forma rápida que o documento pode se tornar obsoleto.
- Nesses casos, como em eXtreme Programming, substitui-se documentos formais por “cartões” de usuário.
- Nesses cartões, os requisitos são coletados de forma incremental.

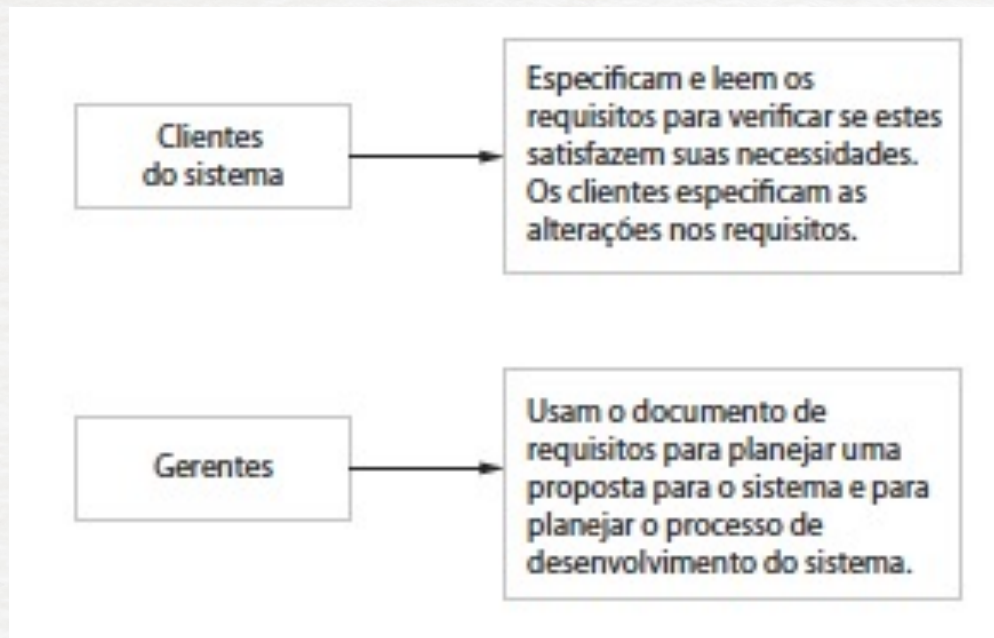
Documento de requisitos e métodos ágeis

- esses cartões incrementais podem ser aplicados a outras metodologias, como *Scrum*?
- Sommerville (2011/2016) menciona que esse tipo de abordagem é boa para sistemas de negócios onde os requisitos são instáveis.
- Mesmo assim o autor menciona a necessidade de um documento de apoio.

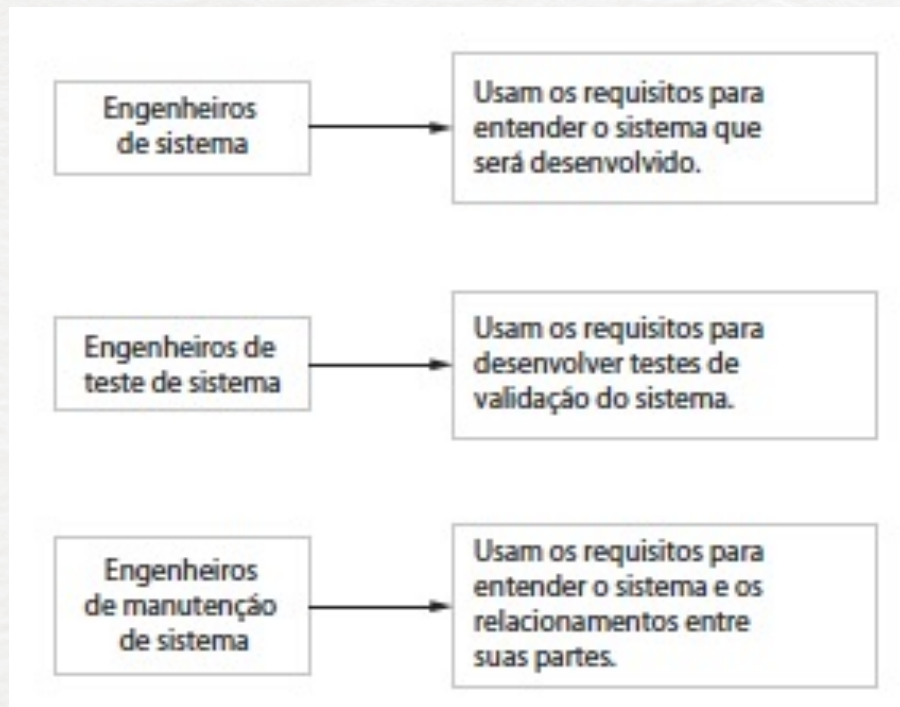
Documento de requisitos e métodos ágeis

- o documento de requisitos tem um conjunto diversificado de usuários: desde a alta administração até Engenheiros de Desenvolvimento de Software.
- O documento deve ter compromisso com a comunicação eficaz e eficiente dos requisitos.
- O compromisso deve se estender às atualizações.

Usuários de um documento de ER



Usuários de um documento de ER



Texto baseado em SOMMERVILLE (2011)



**EDUCAÇÃO
METODISTA**

Estrutura de um documento de ER (baseado na norma corresponde da IEEE, 1998)

- norma genérica que pode ser adaptada para usos específicos.
- forma estendida para incluir evolução prevista do sistema.

Estrutura de um documento de ER (baseado na norma corresponde da IEEE, 1998)

Capítulo	Descrição
Prefácio	Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.
Glossário	Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.
Definição de requisitos de usuário	Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.

Estrutura de um documento de ER (baseado na norma corresponde da IEEE, 1998)

Arquitetura do sistema	Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.
Especificação de requisitos do sistema	Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.
Modelos do sistema	Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.

Estrutura de um documento de ER (baseado na norma corresponde da IEEE, 1998)

Evolução do sistema	Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.
Índice	Vários índices podem ser incluídos no documento. Pode haver, além de um índice alfabético normal, um índice de diagramas, de funções, entre outros pertinentes.