



PROPOSTA DO PROJETO

PROGRAMA DE PÓS-GRADUAÇÃO EM
SISTEMAS DE INFORMAÇÃO - PPGSI

Naiara Alflen

AGENDA

Engenharia de Requisitos

- Etapas
- Requisitos Funcionais e Não funcionais

Proposta do projeto

Agenda proposta

Definição do PO

ENGENHARIA DE REQUISITOS

Fase inicial do projeto

- Conhecer a necessidade do cliente / usuário
- Desenvolvimento de um software completo, consistente e sem ambiguidade

Processo com as atividades

- Elicitação,
- Análise,
- Especificação,
- Validação
- Gerenciamento.

ELICITAÇÃO

O principal objetivo da elicitação é descobrir qual o problema precisa ser resolvido (NUSEIBEH; EASTERBROOK, 2000).

Entender em qual contexto o sistema será usado, para facilitar sua implementação (BATTOOL et al., 2013).

ANÁLISE

Priorizar, detectar e resolver os conflitos entre os requisitos

Requisitos de diversos stakeholders podem entrar em conflito, fazendo necessário a negociação e priorização de requisitos

ESPECIFICAÇÃO DOS REQUISITOS

Descrição completa do que o sistema fará (KOTONYA; SOMMERVILLE, 1996).

- Metodologias tradicionais x metodologias ágeis

VALIDAÇÃO

Busca encontrar problemas nos requisitos (SOMMERVILLE, 2010).

Processo de examinar o documento de requisitos para garantir o desenvolvimento de um software que atenda as expectativas dos (KOTONYA; SOMMERVILLE, 1996).

GERENCIAMENTO

No decorrer de todo o ciclo de vida do desenvolvimento de software, deve-se realizar o gerenciamento dos requisitos (CURCIO et al., 2018)

REQUISITOS FUNCIONAIS

Descrevem o que o sistema deve fazer (SOMMERVILLE, 2010),

Expressam as necessidades dos usuários.

As atividades que o sistema deve executar sem a preocupação das limitações físicas (BUITRON; FLORES-RIOS; PINO, 2018).

REQUISITOS NÃO FUNCIONAIS

Como o sistema executará as atividades do software

É um estágio importante para seleção de tecnologia, alocação de hardware e definição de padrões de desenvolvimento (YOUNAS et al., 2017).

Descrevem a qualidade do software (LIMING et al., 2016)

- Norma ISO 9126

ISO 9126

Crítérios de qualidade:

- Funcionalidade,
- Confiabilidade,
- Usabilidade,
- Eficiência,
- Manutenibilidade
- Portabilidade

FUNCIONALIDADE

Capacidade do software em atender as necessidades que foram propostas pelo usuário.

- Adequação -> funções
- Acurácia -> precisão
- Interoperabilidade -> interação
- Segurança -> proteção

CONFIABILIDADE

Prestar o serviço corretamente, sem perda das informações do usuários (SOMMERVILLE, 2010)

“Software deve estar disponível quando necessário e deve funcionar corretamente e sem efeitos colaterais indesejáveis, como a divulgação de informações não autorizadas” Sommerville (2010) .

- Maturidade -> Evitar falhas
- Tolerância a falhas -> Desempenho
- Recuperabilidade -> Recuperar dados

USABILIDADE

Satisfação do usuário

Software que pode ser facilmente aprendido e utilizável pelo usuário (LIMING et al., 2016).

- Inteligibilidade -> Compreender
- Apreensibilidade -> Aprender
- Operacionalidade -> Operar
- Atratividade -> Visual

EFICIÊNCIA

O software deve ser projetado de forma a não desperdiçar os recursos do sistema (SOMMERVILLE, 2010).

- Comportamento em relação ao tempo -> Processamento
- Utilização dos recursos -> Tipos e Quantidade

MANUTENIBILIDADE

○ sistema precisa ser projetado para facilitar a adição ou edição de funcionalidades e a exclusão de recursos não utilizáveis (LIMING et al., 2016).

- Analisabilidade -> diagnóstico
- Modificabilidade -> modificação
- Estabilidade -> evitar efeitos
- Testabilidade -> validação

PORTABILIDADE

Capacidade do software ser transferido de um ambiente para outro

- Adaptabilidade -> Ambiente
- Capacidade de ser instalado -> Ambiente
- Coexistência -> Compartilhamento de recursos
- Capacidade para substituir -> Mesmo propósito

PROPOSTA DO PROJETO

Sistema para avaliação de funcionários

Protótipo funcional das telas

- WEB
- Editor de texto: Visual Studio Code
 - <https://code.visualstudio.com/>
- Repositório/hospedagem do site: Github Pages
 - <https://pages.github.com/>

GRUPOS

1 - SCRUM	8- SCRUM
2 - SCRUM	9 - DT
3 - DT	10 - SCRUM
4 - DT	11 - DT
5 - DT	12- SCRUM
6 - SCRUM	13- SCRUM
7 - DT	14 - DT

REUNIÕES

15/04 - Skype / Hangout

03/05 – Presencial

XX/XX – Presencial

REFERÊNCIAS

NUSEIBEH, B.; EASTERBROOK, S. Requirements engineering: A roadmap. In: Proceedings of the Conference on The Future of Software Engineering. New York, NY, USA: ACM, 2000. (ICSE '00), p. 35–46. ISBN 1-58113-253-0. Disponível em: <http://doi.acm.org/10.1145/336512.336523i>.

BATTOOL, A.; MOTLA, Y.; HAMID, B.; ASGHAR, S.; RIAZ, M.; MUKHTAR, M.; AHMED, M. Comparative study of traditional requirement engineering and Agile requirement engineering. 2013. 1006-1014 p.

KOTONYA, G.; SOMMERVILLE, I. Requirements engineering with viewpoints. Software Engineering Journal, v. 11, n. 1, p. 5–18, 1996.

SOMMERVILLE, I. Software engineering. [S.l.]: Pearson, 2010. Citado 5 vezes nas páginas 14, 15, 16, 17 e 18. SOUZA, C. L. de C.; SILVA, C. An Experimental Study of the Use of Design Thinking as a Requirements Elicitation Approach for Mobile Learning Environments.

CURCIO, K. D. C.; NAVARRO, T.; MALUCELLI, A.; REINEHR, S. Requirements engineering: a systematic mapping study in agile software development. v. 139, 01 2018. ISSN 0164-1212.

BITRON, S. L.; FLORES-RIOS, B. L.; PINO, F. J. Elicitación de requisitos no funcionales basada en la gestión de conocimiento de los stakeholders. scielocl, v. 26, p. 142 – 156, 03 2018. ISSN 0718-3305.

LIMING, w.; PA, N. C.; ABDULLAH, R.; RAHMAN, W.; TEE, M. Exploring functional and non-functional requirements of social media on knowledge sharing. v. 93, p. 595–605, 11 2016.