

Unicesumar
Londrina

Análise e Desenvolvimento de Sistemas

Análise de Requisitos

Juliana Costa Silva
`juliana.costa@unicesumar.edu.br`

5 de maio de 2020

Aula 1

Introdução

Engenharia de Requisitos

Prática

Aula 2

Leitura recomendada

Referências

Definição:



O que são REQUISITOS?

O que são REQUISITOS?

Requisitos:

Requisitos são objetivos ou restrições estabelecidas por clientes e usuários do sistema que definem as diversas propriedades do sistema.

Projetar e construir software é desafiador, criativo e divertido.

Projetar e construir software é desafiador, criativo e divertido.

- ▶ Esta atividade é tão envolvente, que muitos desenvolvedores querem começar imediatamente, antes de terem um entendimento claro daquilo que é necessário.

Projetar e construir software é desafiador, criativo e divertido.

- ▶ Esta atividade é tão envolvente, que muitos desenvolvedores querem começar imediatamente, antes de terem um entendimento claro daquilo que é necessário.
- ▶ Eles argumentam que as coisas ficarão mais claras à medida que forem construindo o software;

Projetar e construir software é desafiador, criativo e divertido.

- ▶ Esta atividade é tão envolvente, que muitos desenvolvedores querem começar imediatamente, antes de terem um entendimento claro daquilo que é necessário.
- ▶ Eles argumentam que as coisas ficarão mais claras à medida que forem construindo o software;
- ▶ Que os envolvidos no projeto serão capazes de entender a necessidade apenas depois de examinar as primeiras iterações do software;

Projetar e construir software é desafiador, criativo e divertido.

- ▶ Esta atividade é tão envolvente, que muitos desenvolvedores querem começar imediatamente, antes de terem um entendimento claro daquilo que é necessário.
- ▶ Eles argumentam que as coisas ficarão mais claras à medida que forem construindo o software;
- ▶ Que os envolvidos no projeto serão capazes de entender a necessidade apenas depois de examinar as primeiras iterações do software;
- ▶ Que as coisas mudam tão rápido que qualquer tentativa de entender os requisitos de forma detalhada será perda de tempo;

Projetar e construir software é desafiador, criativo e divertido.

- ▶ Esta atividade é tão envolvente, que muitos desenvolvedores querem começar imediatamente, antes de terem um entendimento claro daquilo que é necessário.
- ▶ Eles argumentam que as coisas ficarão mais claras à medida que forem construindo o software;
- ▶ Que os envolvidos no projeto serão capazes de entender a necessidade apenas depois de examinar as primeiras iterações do software;
- ▶ Que as coisas mudam tão rápido que qualquer tentativa de entender os requisitos de forma detalhada será perda de tempo;
- ▶ Que o primordial é produzir um programa que funcione e que todo o resto é secundário.

[Pressman, 2011].

As afirmações são verdadeiras?



- ▶ Sim! Se o projeto for pequeno;

As afirmações são verdadeiras?



- ▶ Sim! Se o projeto for pequeno;
- ▶ Com entregas em menos de um mês;

As afirmações são verdadeiras?



- ▶ Sim! Se o projeto for pequeno;
- ▶ Com entregas em menos de um mês;
- ▶ Com esforços simples.

As afirmações são verdadeiras?



- ▶ Sim! Se o projeto for pequeno;
- ▶ Com entregas em menos de um mês;
- ▶ Com esforços simples.
- ▶ A medida que o software cresce as argumentações são falhas.

O amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos é chamado de **Engenharia de Requisitos**.

O amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos é chamado de **Engenharia de Requisitos**.

O amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos é chamado de **Engenharia de Requisitos**.



Dilbert by Scott Adams, 2006

Fonte: [esderpa, 2012]

A engenharia de requisitos abrange sete tarefas distintas:

1. Concepção;
2. Levantamento;
3. Elaboração;
4. Negociação;
5. Especificação;
6. Validação;
7. Gestão.

1. Em equipe (máximo 4 integrantes);
2. Trabalhar com o projeto da AEP;
3. Realizem então as tarefas de Engenharia de Requisitos conforme apresentadas em sala, aplicando ao seu projeto da AEP.
4. A aplicação será apresentada na próxima aula ao vivo;

Na concepção do projeto, estabelecemos um entendimento básico do problema, as pessoas que querem uma solução, a natureza da solução desejada e a eficácia da comunicação e colaboração preliminares entre os demais envolvidos e a equipe de software.

Certamente parece bastante simples: perguntar ao cliente, aos usuários e aos demais envolvidos quais são os objetivos para o sistema ou produto, o que deve ser obtido, como o sistema ou produto atende às necessidades da empresa e, por fim, como o sistema ou produto deve ser utilizado no dia a dia.

Mas isso nao é simples... Na verdade, é muito difícil.

As informações obtidas do cliente durante a concepção e o levantamento são expandidas e refinadas durante a elaboração. Essa tarefa concentra-se no desenvolvimento de um modelo de requisitos refinado, que identifique os diversos aspectos da função, do comportamento e das informações do software.

Não é raro clientes e usuários pedirem mais do que é possível, dados os recursos limitados do negócio. Também é relativamente comum diferentes clientes ou usuários proporem necessidades conflitantes, argumentando que sua versão é “essencial para nossas necessidades especiais”. É preciso conciliar esses conflitos por meio de um processo de negociação. Devemos solicitar a clientes, usuários e outros envolvidos para que ordenem seus requisitos e discutam sua prioridade.

No contexto de sistemas baseados em computador (e software), o termo especificação assume diferentes significados para diferentes pessoas. Especificação pode ser um documento por escrito, um conjunto de modelos gráficos, um modelo matemático formal, um conjunto de cenários de uso, um protótipo ou qualquer combinação dos fatores citados.

INFORMAÇÕES



Modelo de especificação de requisitos de software

Uma especificação de requisitos de software (SRS, *software requirements specification*) é um artefato criado quando uma descrição detalhada de todos os aspectos do software a ser construído deve ser especificada antes de o projeto começar. É importante notar que uma SRS formal nem sempre é por escrito. Na realidade, há várias ocasiões em que o esforço gasto em uma SRS talvez fosse mais bem aproveitado em outras atividades de engenharia de software. Entretanto, quando um software for desenvolvido por terceiros, quando a falta de uma especificação criar graves problemas de negócio ou quando um sistema for extremamente complexo ou crítico para o negócio, será justificável uma SRS.

Karl Wieggers [Wie03], da Process Impact Inc., desenvolveu uma planilha bastante útil (disponível em www.processimpact.com/process_assets/srs_template.doc) que pode servir como diretriz para aqueles que precisam criar uma SRS completa. Uma descrição geral por tópicos é apresentada a seguir:

Sumário

Histórico de Revisão

1. Introdução
 - 1.1 Finalidade
 - 1.2 Convenções do documento
 - 1.3 Público-alvo e sugestões de leitura
 - 1.4 Escopo do projeto
 - 1.5 Referências

2. Descrição geral

- 2.1 Perspectiva do produto
- 2.2 Características do produto
- 2.3 Classes e características do usuário
- 2.4 Ambiente operacional
- 2.5 Restrições de projeto e implementação
- 2.6 Documentação para usuários
- 2.7 Hipóteses e dependências

3. Características do sistema

- 3.1 Características do sistema 1
- 3.2 Características do sistema 2 (e assim por diante)

4. Requisitos de interfaces externas

- 4.1 Interfaces do usuário
- 4.2 Interfaces de hardware
- 4.3 Interfaces de software
- 4.4 Interfaces de comunicação

5. Outros requisitos não funcionais

- 5.1 Requisitos de desempenho
- 5.2 Requisitos de segurança – privacidade
- 5.3 Requisitos de segurança – integridade
- 5.4 Atributos de qualidade de software

6. Outros requisitos

Apêndice A: Glossário

Apêndice B: Modelos de análise

Apêndice C: Lista de problemas

Uma descrição detalhada de cada tópico de SRS pode ser obtida fazendo-se o download da planilha SRS na URL citada anteriormente neste quadro.

Para mais informações sobre Engenharia de Requisitos, leia:

[Pfleeger, 2004]
Capítulo 4.



[esdarpa, 2012] esdarpa (2012).

A difícil arte de analisar requisitos.

[Pfleeger, 2004] Pfleeger, S. (2004).

Engenharia de software: teoria e prática.

Prentice Hall.

[Pressman, 2011] Pressman, R. S. (2011).

Engenharia de software: uma abordagem profissional. 7^a edição.

Ed: McGraw Hill.

Vamos em frente!
Até a próxima aula.