

REQUISITOS, MODELAGEM E ANÁLISE DE DADOS

PLANO DE ENSINO E INTRODUÇÃO

Professor Joe Jonas Vogel

Apresentação do professor

- Joe Jonas Vogel
- MBA em Empreendedorismo e Inovação - SENAI
- Bacharel em Sistemas de Informação - Católica
- Anterior:
 - Pesquisador em robótica na Católica;
 - Pesquisador em ML, IA e novas tecnologias na WEG.
- Atual:
 - CEO & Founder da Semente Urbana
 - Professor na Católica
- Conhecimentos: Robótica, IoT, Big Data, Machine Learning, Desenvolvimento e Gestão
- Interesses: IoT, Biotecnologia, Cultivo Indoor / Urban Farm, Dados e Startups

Plano de Ensino

- Introdução
- Requisitos:
 - Conceitos
 - Tipos
 - Técnicas de levantamento
 - Gerenciamento
- Estudo de viabilidade
- Modelagem
 - Conceitos
 - UML (Conceito, Diagramas, etc)
 - Avaliação
- Análise de dados
- Trabalho final

Cronograma

- O conteúdo planejado no PE será distribuído ao longo de 80 aulas (20 noites)
- Cronograma

INTRODUÇÃO

Engenharia de Software

- Empresas buscando apoio nos fundamentos de Engenharia de Software para o desenvolvimento de sistemas
- Engenharia de Software, de acordo com o IEEE, é a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, operação e manutenção de software
 - Sistemática por que parte do princípio de que existe um processo de desenvolvimento definindo as atividades que deverão ser executadas
 - Disciplinada por que parte do princípio de que os processos definidos serão seguidos.
 - Quantificável por que se deve definir um conjunto de medidas a serem extraídas do processo durante o desenvolvimento de forma que as tomadas de decisão relacionadas ao desenvolvimento do software (por exemplo, melhoria de processo) sejam embasadas em dados reais, e não em “achismos”

Alguns de seus principais objetivos são:

- Qualidade de software;
- Produtividade no desenvolvimento, operação e manutenção de software;
- Permitir que profissionais tenham controle sobre o desenvolvimento de software dentro de custos, prazos e níveis de qualidade desejados.

	% do Custo de Desenvolvimento	% dos erros introduzidos	% dos erros encontrados	Custo relativo de correção
Análise de Requisitos	5	55	18	1
Projeto	25	30	10	1 - 1.5
Códificação e teste de unidade	50			
Teste	10	10	50	1 - 5
Validação e Documentação	10			
Manutenção		5	22	10 - 100

Figura 1. Panorama atual em projetos de desenvolvimento

- Menor % do custo
- Maior número de erros introduzidos
- Custo de correção baixo
- Se os defeitos não forem tratados no início, influenciará no encarecimento do projeto e na queda da confiança

Reforçando a importância que as atividades relacionadas a requisitos devem possuir na indústria de software, estudo realizado pelo Standish Group, considerando 350 companhias e 8.000 projetos de software, em 1995 revelou que:

- 16,2% dos projetos são finalizados com sucesso, ou seja, cobre todas as funcionalidades em tempo e dentro do custo previsto;
- 52.7% dos projetos são considerados problemáticos, ou seja, não cobre todas as funcionalidades exigidas, custo aumentado e está atrasado.
- 31,1% dos projetos fracassam, ou seja, o projeto é cancelado durante o desenvolvimento.

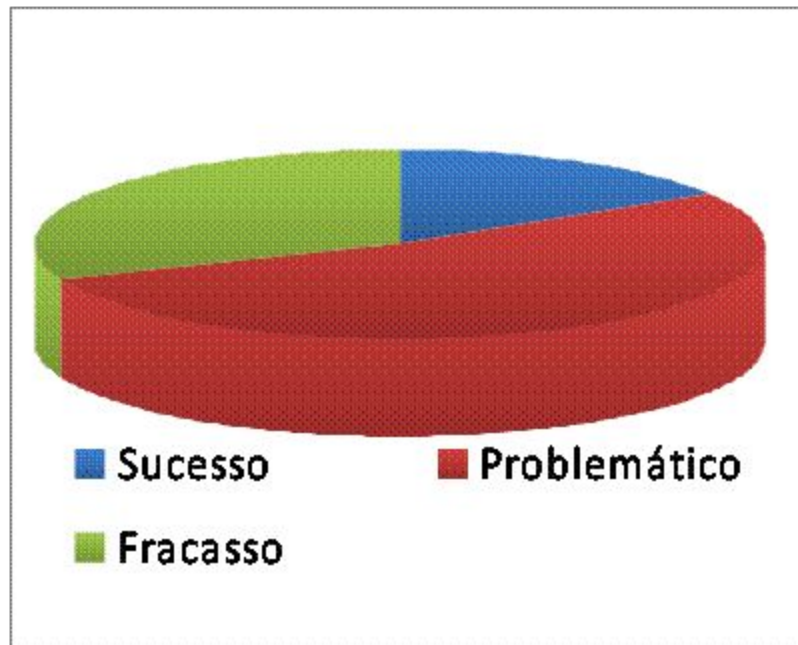


Figura 2. Distribuição da conclusão de projetos de software

Fatores Críticos	%
1. Requisitos Incompletos	13.1%
2. Falta de Envolvimento do Usuário	12.4%
3. Falta de Recursos	10,6%
4. Expectativas Irreais	9,9%
5. Falta de Apoio Executivo	9,3%
6. Mudança de Requisitos e Especificações	8,7%
7. Falta de Planejamento	8,1%
8. Sistema não mais necessário	7,5%

Figura 3. Fatores críticos para o sucesso do projeto

Podemos perceber que três dos principais fatores estão relacionados às atividades de requisitos: (1) Requisitos Incompletos; (2) Falta de Envolvimento do Usuário; (6) Mudança de Requisitos e Especificações.

Neste ponto, sabemos que um trabalho mais criterioso na área de requisitos é fundamental para o sucesso de projetos de software.

O que é um requisito???

Requisitos

Existem diferentes definições encontradas na literatura técnica para requisitos:

- Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos;
- As descrições das funções e restrições são os requisitos do sistema;
- Um requisito é uma propriedade que o software deve exibir para resolver algum problema no mundo real;
- Uma condição ou uma capacidade que deve ser alcançada ou estar presente em um sistema para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto...

Podemos entender requisitos como sendo o conjunto de necessidades explicitadas pelo cliente que deverão ser atendidas para solucionar um determinado problema do negócio no qual o cliente faz parte

Requisitos são importantes para:

- Estabelecer uma base de concordância entre o cliente e o fornecedor sobre o que o software fará;
- Fornecer uma referência para a validação do produto final;
- Reduzir o custo de desenvolvimento (como vimos anteriormente, requisitos mal definidos causam retrabalho).

- Uma vez que temos os requisitos do sistema proposto, é importante avaliarmos se é viável ou não a construção do mesmo.

Estudo de Viabilidade

- O objetivo de um estudo de viabilidade, como o próprio nome já diz, é avaliar sob o ponto de vista operacional, técnico, econômico e organizacional se o projeto é viável.
- O documento destina-se aos stakeholders do sistema (alguém que tenha alguma influência direta ou indireta sobre os requisitos do sistema).
(SOMMERVILLE,2011, p. 70).
- O estudo de viabilidade acontece após a especificação de requisitos de negócio, ou seja, é o segundo passo do processo de engenharia de requisitos.

- Justifica-se a sua realização a fim de analisar e responder algumas questões do ponto de vista da viabilidade operacional, técnica, de cronograma e econômica, como:
 - É mais viável desenvolver um sistema novo ou utilizar algum alternativa já disponível no mercado?
 - Quais são os riscos envolvido no projeto?
 - O sistema proposto contribui para a organização?
 - Quais as vantagens do sistema proposto em relação aos sistemas disponíveis no mercado?
 - O sistema proposto é útil ao cliente e resolverá o problema dele?
 - Analisando as condições econômicas, organizacionais e temporais é viável desenvolver o sistema?
- Uma vez demonstrada a viabilidade do sistema proposto, entramos então no processo de desenvolvimento do sistema

Processos de Software

- Existem vários modelos de processo de software (ou paradigmas de engenharia de software)
- Cada um representa uma tentativa de colocar ordem em uma atividade inerentemente caótica
- Um modelo de ciclo de vida ou modelo de processo pode ser visto como uma representação abstrata de um esqueleto de processo, incluindo tipicamente algumas atividades principais, a ordem de precedência entre elas e, opcionalmente, artefatos requeridos e produzidos.
- De maneira geral, um modelo de processo descreve uma filosofia de organização de atividades, estruturando as atividades do processo em fases e definindo como essas fases estão relacionadas.

- Em geral, o ciclo de vida de um software envolve, pelo menos, as seguintes fases:
 - Planejamento
 - Análise e Especificação de Requisitos
 - Projeto
 - Implementação
 - Testes
 - Entrega e Implantação
 - Operação
 - Manutenção

Modelagem de Sistemas

- Uma das atividades chave em um processo de software é a modelagem do sistema proposto
- A ideia aqui é parecida com a fase de projeto na construção civil, onde através de mapas e diagramas a obra é modelada e estes então servem como guia para toda a equipe envolvida na construção.
- Durante a modelagem de sistemas, são criados diferentes diagramas que demonstram de diferentes maneiras cada parte do sistema a ser desenvolvido
- Com base nestes diagramas, todos os membros da equipe podem fazer a sua parte, análogo ao que acontece na construção civil onde o pedreiro, o encanador e o eletricista através do projeto podem cada um focar em sua parte da obra.

Modelagem de Dados

- Modelagem de dados é o ato de explorar estruturas orientadas a dados
- Como outros artefatos de modelagem, modelos de dados podem ser usados para uma variedade de propósitos, desde modelos conceituais de alto nível até modelos físicos de dados
- Com a modelagem de dados identificamos tipos de entidades da mesma forma que na modelagem de classes identificamos classes
- A ideia aqui é analisar os dados que serão utilizados/gerados na aplicação, isso é de suma importância para que se possa desenvolver o software de modo assertivo.

Trabalho final

- Ao final da disciplina e como nota 3 de vocês, faremos então a análise e modelagem de um sistema real.
- O intuito é praticar e fixar o conteúdo visto na disciplina e assim torná-los aptos a trabalhar em grandes projetos de software.

Fundamentação em Engenharia de Requisitos

- Vocês irão ler o artigo “Especificação de Requisitos: Uma Introdução” disponível na pasta Materiais de Aula na disciplina do Teams
- Este artigo será fundamental para a compreensão dos temas que serão abordados nas próximas aulas