Perfil:

Emerson Antonio Klisiewicz.

Possui graduação em Ciências da Computação, pela Pontifícia Universidade Católica do Paraná (1994), Especialização em Sistemas de Informações Gerenciais , pela Pontifícia Universidade Católica do Paraná (1995), Especialização em Redes e Sistemas Distribuídos pela Pontifícia Universidade Católica do Paraná (1997), com experiência de mais de 20 anos em áreas de tecnologia de empresas estatais e financeiras. Currículo Lattes:
http://buscatextual.cnpq.br/buscatextual/visualizacv.do?id=K4282839T3

AULA 03 – Engenharia de Software

Introdução:

Na terceira aula, vamos continuar a abordar a evolução da Análise de Sistemas, dando ênfase agora a Engenharia de software que estabelece e usa de princípios de engenharia de forma a obter economicamente software confiável e que funcione eficientemente em máquinas reais.

Contextualizando:

Software

São programas de computadores, em suas diversas formas, e a documentação associada. Um programa é um conjunto de soluções algorítmicas, codificadas numa linguagem de programação, executado numa máquina real. Software é um produto conceitual e lógico. Também temos produtos genéricos, stand-alone que são produzidos e vendidos no mercado e produtos sob encomenda ou seja, sistemas encomendados.

Características do Software

- ✓ Invisibilidade: Software é invisível e invisualizável.
- ✓ Complexidade: Software é mais complexo do que qualquer outro produto construídos por seres humanos.
- ✓ Mutabilidade: Existe sempre uma pressão para se fazer mudanças em um software.
- ✓ Conformidade : O software deve ser desenvolvido conforme o ambiente. Não é o ambiente que deve se adaptar ao software.
- ✓ Se o software esta conforme os requisitos (o ambiente) todo o suporte operacional deve se adaptar ao software.

Engenharia de Software

O estabelecimento de objetivos gerais é suficiente para se começar a escrever programas. Dê a uma pessoa técnica um bom livro de programação e você terá um programador. Mudanças no software podem ser feitas facilmente porque ele é "flexível" e até que o programa esteja "rodando" não é possível verificarmos a sua qualidade. Uma vez que o programa esteja escrito e funcionando, nosso trabalho está feito. Um projeto é bem sucedido se conseguirmos um programa funcionando corretamente. A Engenharia de Software se preocupa em sistematizar o desenvolvimento através de modelos, técnicas e ferramentas para o produto e para o processo.

Objetivos da Engenharia de Software

- ✓ Aplicação de teoria, modelos, formalismos, técnicas e ferramentas da ciência da computação e áreas afins para o **desenvolvimento** sistemático de software.
- ✓ Aplicação de métodos, técnicas e ferramentas para o **gerenciamento** do processo de desenvolvimento.
- ✓ Produção da **documentação** formal destinada a comunicação entre os membros da equipe de desenvolvimento bem como aos usuários.

Divisões:

a-) Métodos: proporcionam os detalhes de como fazer para construir o software.

- ☞ Planejamento e estimativa de projeto
- ☞ Análise de requisitos de software e de sistemas
- ☞ Projeto da estrutura de dados
- ☞ Algoritmo de processamento
- ☞ Codificação
- ☞ Teste
- ☞ Manutenção

b-) Ferramentas: dão suporte automatizado aos métodos. Existem atualmente ferramentas para sustentar cada um dos métodos. Quando as ferramentas são integradas é estabelecido um sistema de suporte ao desenvolvimento de software chamado CASE - Computer Aided Software Engineering.

c-)Procedimentos: constituem o elo de ligação entre os métodos e ferramentas.

- ☞ Seqüência em que os métodos serão aplicados.
- ☞ Produtos que se exige que sejam entregues.

- ☞ Controles que ajudam assegurar a qualidade e coordenar as alterações.
- ☞ Marcos de referência que possibilitam administrar o progresso do software.

Princípios:

E a evolução se baseou nos chamados Ciclos de Vida de Sistemas. Dentro desse contexto temos as seguintes fases:

Fase de definição:

– Análise e Especificação.

– Estudo de Viabilidade.

– Estimativas Planejamento.

Deve-se analisar os requisitos, recursos e restrições para apresentar soluções, estudar a viabilidade, planejar e gerenciar o desenvolvimento e a partir de estimativas e análise de riscos que se utilizam de métricas. Esta fase encerra-se com o contrato de desenvolvimento.

Fase de desenvolvimento:

– Design.

– Implementação e integração.

– Verificação e Validação.

Design conceitual, design da interface de usuário, design da arquitetura de software, design de algoritmos e estruturas de dados. Codificação, compilação, integração e verificação de programas (testes, inspeção, depuração). Testes beta, avaliação de usabilidade, avaliaçãode desempenho, etc...

Fase de operação:

– Distribuição.

– Instalação.

– Configuração.

– Utilização.

– Administração.

– Manutenção.

Instalação e configuração, utilização, correção de erros, implementação de novas versões, novas situações de operação, hardware e sistemas operacionais.

Fase de retirada:

– Migração.

– Reengenharia.

– Rengenharia reversa.

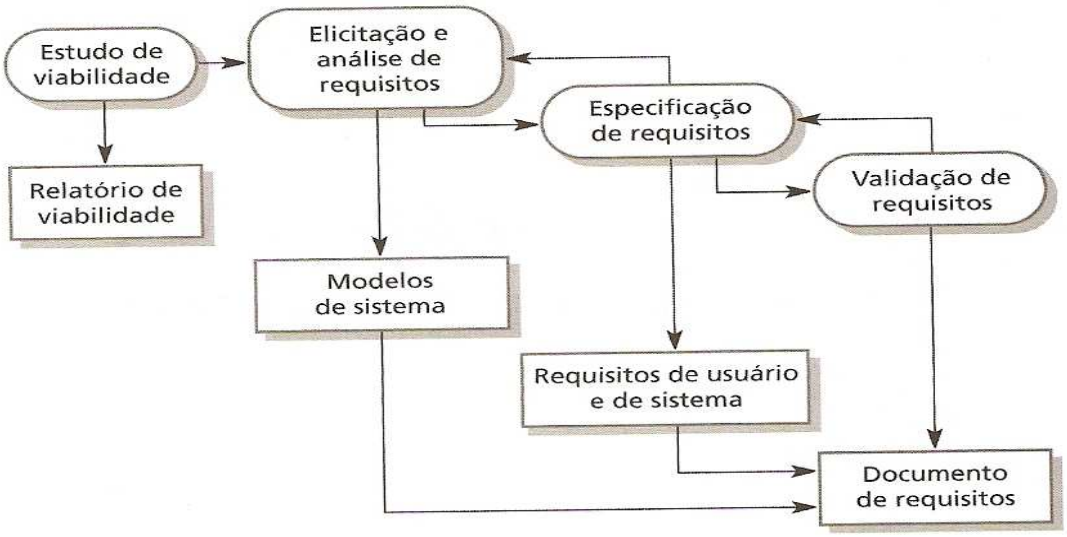
Evolução do software, ajustes em sistemas legado ou reengenharia de Software e sistema.

Engenharia de Requisitos

"Estabelecer quais funções são requeridas pelo sistema e as restrições sobre a operação e o desenvolvimento do sistema. Um processo que envolve todas as atividades exigidas para criar e manter o **documento de requisitos de sistema**". Sommerville p. 46.

Objetivo:

Criar e manter um documento de requisitos. Possui 4 subprocessos: Estudo de viabilidade, elicitação e análise de requisitos, especificação e validação de requisitos.



a-) Estudo de Viabilidade:

Estudo que indica se o esforço em desenvolver a idéia vale a pena. Visa tanto a tomada de decisão e também a sugestão de possíveis alternativas de solução.

Deve oferecer informações para ajudar na decisão e se o projeto pode ou não ser feito. Se o produto final irá ou não beneficiar os usuários interessados e também a possibilidade de escolha das alternativas entre as possíveis soluções. Sistema organizacional apresentado

Devemos começar por estudos de usuários, políticas, funções, objetivos, e problemas com o sistema apresentado. Inconsistências, funcionalidades inadequadas, performance também devem ser o foco como também os objetivos e outros requisitos para o novo sistema.

b-) Requisitos:

Uma sentença identificando uma capacidade, uma característica física ou um fator de qualidade que limita um produto ou um processo. (IEEE 1220-1994).

O requisito é uma condição cuja exigência deve ser satisfeita. Se a condição é produzir algo, diz-se que o requisito é funcional e se a condição é caracterizar algo (propriedade, comportamento, restrição, etc,...), diz-se que o requisito é não-funcional.

b.1.) Requisito Funcional:

Requisitos funcionais correspondem à listagem de todas as coisas que o sistema deve fazer. Depende do tipo do software, usuários esperados e do tipo do sistema onde o software é usado.

b.1.) Requisito Não Funcional:

Requisitos não funcionais são restrições e qualidades que se coloca sobre como o sistema deve realizar seus requisitos funcionais. Usabilidade, confiabilidade, desempenho, configurabilidade, portabilidade e segurança são como esses requisitos podem ser classificados.

c-) Casos de Uso:

São textos que descrevem e justificam as funcionalidades de uma plataforma ou sistema através das interações com seus usuários ou entidades externas ao sistema. Criados a partir de estruturas previamente definidas, são utilizados para descobrir, registrar e avaliar os requisitos do

sistema para os desenvolvedores e desenvolver uma MÉTRICA para estimar a dimensão funcional (functional size) de um sistema ou projeto. Cada caso de uso tem uma descrição o qual descreve a funcionalidade que irá ser construída no sistema proposto.

O "cadastro em um sistema", o "login em uma plataforma", "um pedido de informação", "o fechamento de uma compra", podem ser descritos tecnicamente com casos de uso interligados sobre um sistema de comércio online ou mesmo de um processo batch normal que será disparado por algum gatilho ou ação qualquer.

Síntese

Sistemas de software são reconhecidamente importantes ativos estratégicos para diversas organizações. Os sistemas têm papel vital no apoio aos processos de negócio, então é fundamental que os sistemas funcionem de acordo com os requisitos estabelecidos. Neste contexto, uma importante tarefa no desenvolvimento de software é a identificação e o entendimento dos requisitos dos negócios que os sistemas vão apoiar.

Não importa quão bem projetado ou codificado está um programa, se ele for mal analisado e especificado desapontará o usuário e trará aborrecimentos ao desenvolvedor.