

Engenharia de Software

Prof.^a Dr.^a Elisa Moriya Huzita

Ementa: Introdução à Engenharia de Software. Processo de Software. Modelos de Processo de Software. Requisitos de Software. Modelagem de Software. Linguagem de Modelagem. Validação e Manutenção de Software. Ferramentas de Apoio ao Desenvolvimento de Software. Qualidade de Software.

1. Introdução à Engenharia de Software

1.1 Software: Conceitos, importância e características

1.2 Tipos de Aplicações de Software

1.3 Crise de software

1.4 A Engenharia de software

- Software

“criação intelectual compreendendo os programas, procedimentos, regras e qualquer documentação correlata à operação de um sistema de processamento de dados.” ISO9000-3

Produto de Software

“Conjunto completo de programas de computador, procedimentos e documentação correlata, assim como dados prejetados para entrega a um usuário.” ISO9000-3

Produto:

mercado - editor ou
usuário específico - negócio

Outros termos: sistemas de software, aplicações, aplicativos

LAS (*London Ambulance System*) - chamadas de emergência atribuição de ambulâncias a acidentes.

Abrange uma população de ~6.8 milhões, transporta mais de 5.000 pacientes/ dia, recebe entre 2.000 e 2.500 chamadas/dia.

Falhas no sistema (2 dias) causaram problemas tais como:

- Alocação ineficiente de ambulâncias (duplicada e atrasada - múltiplas ambulâncias enviadas para o mesmo acidente ou o veículo mais próximo não foi enviado para o acidente mais próximo);
- Crescente lista de espera;
- Aumento do tempo de resposta proporcional ao aumento de mensagens e da lista de espera;
- Crescente número de telefonemas repetidos (“call backs”).

- Software é desenvolvido e não manufaturado no sentido clássico.
- Software não se deteriora ... não existem componentes de reposição.
- Software é feito sob encomenda, ao invés de ser construído a partir de componentes (está mudando).
- A evolução tecnológica afeta diretamente as facilidades que podem ser incorporadas ao software e as técnicas de construção destes.
- Tempo para transferência de tecnologia.

Como evoluiu ao longo do tempo?

O que se pode dizer da construção?

- Sistemas batch
 - sistemas personalizados
- Sistemas multi-usuários
 - BD, produto (sw)
- Sistemas distribuídos
 - Incorporou inteligencia, redes neurais, redução custo hw
- Computação paralela
 - OO, comunicação intergalática

- Computação baseada em componentes, baseada em agentes, computação móvel, computação orientada a serviços
- Computação em nuvem
-

- Software básico
- Software comercial
- Software para engenharia e aplicações científicas
- Software embarcado
- Software para computadores pessoais
- Software baseados em inteligência artificial
- Software de entretenimento
- Software legado



Problema → → solução
(mundo real) (computacional)
("cliente") (desenvolvedor)

Alguns problemas.....

- A sofisticação do hardware ultrapassa nossa habilidade de construir software que utilize todo o potencial do hardware
- Nossa habilidade para construir novos produtos de software não acompanha a demanda
- Nossa habilidade de manter os produtos de software existentes estão ameaçadas por projetos ineficientes e recursos inadequados (pessoal mal treinado ou mal alocado)
- Software de baixa qualidade (desempenho, interoperabilidade, reuso, entendimento equivocado dos requisitos)

De um modo geral decorrentes de:

- De complexidade e tamanho
- De comunicação
- De projeto (arquitetura)
- Problemas de gestão

Fatores que estavam dificultando

- ▶ estimativas de prazo e de custo ↑
- ▶ produtividade das pessoas ↓
- ▶ qualidade de software ↓
- ▶ software difícil de manter ↑

CRISE DE SOFTWARE

Causas dos problemas:

1. **Próprio Caráter do Software**

- O software é um elemento de sistema lógico e não físico.
- Consequentemente, o sucesso é medido pela qualidade de uma única entidade

O software não se desgasta, mas se deteriora!!!

2. Falhas das Pessoas Responsáveis pelo Desenvolvimento de Software

- Gerentes sem background em software
- Os profissionais da área de software nem sempre recebem treinamento formal em novas técnicas para o desenvolvimento de software
- Resistência a mudanças

3. Mitos do software

Os principais:

Administrativos

Cliente

Profissional

ADMINISTRATIVO – MITO 1

Já temos um manual repleto de padrões e procedimentos para a construção de software.

Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?

Realidade - Mito 1:

- Será que o manual é usado?
- Os profissionais sabem que ele existe?
- Ele reflete a prática moderna de desenvolvimento de software?
- Ele é completo?
- Ele está atualizado?

Administrativo *Mito 2* :

Meu pessoal tem Ferramentas de desenvolvimento de software de última geração.

Realidade:

É preciso muito mais do que os mais recentes computadores e as melhores ferramentas para se desenvolver software de alta qualidade.

Administrativo Mito 3:

Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e recuperar o atraso

Realidade:

O desenvolvimento de software não é um processo mecânico igual à manufatura. Acrescentar pessoas em um projeto torna-o ainda mais atrasado.

Pessoas podem ser acrescentadas, mas somente de uma forma planejada.

CLIENTE: *Mito 1:*

Uma declaração geral dos objetivos é suficiente para começar a escrever programas - podemos preencher os detalhes mais tarde.

Realidade:

Uma definição inicial ruim é a principal causa de fracassos dos esforços de desenvolvimento de software.

É fundamental uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação.

CLIENTE: *Mito 2:*

Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível.

Realidade:

Uma mudança, quando solicitada tardiamente num projeto, pode ser maior do que a ordem de magnitude mais dispendiosa da mesma mudança solicitada nas fases iniciais.

Magnitude:

inicial: 1

desenvolvimento :1,5 a 6

manutenção: 60 a 100

PROFISSIONAL: *Mito 1:*

Assim que escrevermos o programa e o colocarmos em funcionamento nosso trabalho estará completo.

Realidade:

Os dados da indústria indicam que entre 50 e 70% de todo esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente.

PROFISSIONAL: *Mito 2* :

Enquanto não tiver o programa "funcionando", eu não terei realmente nenhuma maneira de avaliar sua qualidade.

Realidade:

Um programa funcionando é somente uma parte de uma Configuração de Software que inclui todos os itens de informação produzidos durante a construção e manutenção do software.

O que pensam gerentes, técnicos

Eles perguntam:

- Porque é preciso tanto tempo para terminar os programas?
- Porque os custos são tão altos?
- Porque não se consegue encontrar todos os erros antes que o software seja liberado para os clientes?
- Porque existe uma dificuldade em medir o progresso à medida que o software está sendo construído?

A preocupação em resolver essas questões tem levado à adoção das práticas da **Engenharia de Software**

O que é Engenharia de Software?

► Definição (1) - (Nato Conference 1968)

- Estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente software que seja confiável e que funcione eficientemente em máquinas reais.

► Definição (2) - (IEEE Glossary of Software Engineering)

- Aplicação de uma abordagem sistêmica, disciplinada e mensurável para o desenvolvimento, operação e manutenção de software, isto é, a aplicação da engenharia ao software.

Definição (3) – (SEI Report on Undergraduate Software Engineering Education)

- Engenharia é a aplicação sistemática de conhecimento científico na criação e construção de soluções eficientes e de baixo custo para problemas práticos do ser humano
- Engenharia de software é o ramo da engenharia que aplica os princípios da ciência da computação e da matemática para obter tais soluções para os problemas de software.

Definição (4) Sommerville (2011),

- É uma disciplina cujo foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até sua manutenção.

O que se espera?

Rigor

Disciplina

Qualidade

Precisamos de : processos, ferramentas →
qualidade

Engloba:

- Modelos/Processos de produção de software,
- Padrões de projetos,
- Testes de software,
- Medições,
- Manutenções,
- Garantia da qualidade,

Suporte de ferramentas:

- Modelagem
- Teste
- Gestão de configuração

Uma ilustração para fatores de insucesso de projeto



Imagem: Shutterstock

Disponível em: <<https://www.youtube.com/watch?v=QPiR8jTMLdI>>

Considere a empresa ComidaBoa que atua no ramo alimentício. Apesar de ela estar no mercado há 5 anos, todas as atividades na empresa são realizadas manualmente. Dentre as atividades incluem-se: atendimento ao cliente, controle de estoque dos insumos para preparar os pratos diariamente, entrega dos pratos solicitados pelos clientes, pagamento aos fornecedores, compra de fornecedores., controle de recebimentos dos clientes. No início a empresa não tinha muitos clientes, o que permitia a realização destas atividades de forma satisfatória. No entanto, com o decorrer do tempo começaram a perceber que estavam perdendo o controle da situação.

A seu ver como poderia ajudar o proprietário da ComidaBoa?

Engenharia de Software

Prof.^a Dr.^a Elisa Moriya Huzita