

Curso de Especialização em Engenharia de Software

Disciplina:

Engenharia de Software

Prof. Dr. Rodolfo Miranda de Barros



APRESENTAÇÃO DA DISCIPLINA

Ementa:

Conceitos e Fundamentos da Engenharia de Software; Processo (paradigmas) de desenvolvimento de software. Apresentar e discutir assuntos emergentes relacionados à área de engenharia de software.

Introdução

Importância do Software!

- A evolução tecnológica e a consequente diminuição do custo dos equipamentos possibilitaram a expansão dos sistemas computacionais;
- Atualmente, o uso dos sistemas computacionais se dá nos mais variados setores da atividade humana, o que os torna um agente ativo na sociedade em que vivemos;
- Neste sentido, cresce a preocupação com o desenvolvimento destes sistemas computacionais, no que diz respeito às técnicas empregadas, gerenciamento do seu desenvolvimento, projeto de banco de dados, softwares utilizados para o desenvolvimento, qualidade do processo de desenvolvimento e do produto final, segurança, entre outros fatores;



Para Refletir ...

Segundo Roger Pressman,
se você não analisa, é
altamente provável que
construa uma solução de
software muito elegante que
resolve o problema errado!



Como o cliente
explicou...



Como o líder de projeto
entendeu...



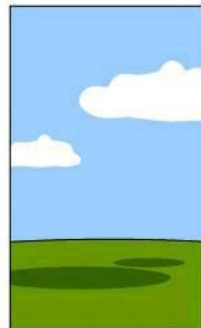
Como o analista
projetou...



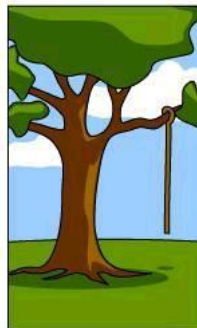
Como o programador
construiu...



Como o Consultor de
Negócios descreveu...



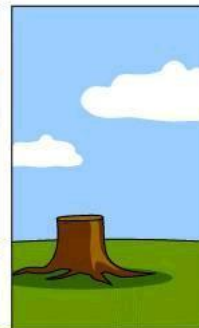
Como o projeto foi
documentado...



Que funcionalidades
foram instaladas...



Como o cliente foi
cobrado...



Como foi mantido...



O que o cliente
realmente queria...

Falhas Famosas de Software - Ariane 5

- Ariane 5, 1996:
 - O foguete explodiu 40 segundos após a sua primeira decolagem;
 - Prejuízo de US\$ 500 milhões;
 - Os cientistas que desenvolveram o foguete Ariane 5, voo 501, reutilizaram parte do código de seu predecessor, o Ariane 4, mas os motores do novo foguete incorporavam também, sem que ninguém desse conta, um bug numa rotina aritmética no computador de voo que falhou segundos após a decolagem do foguete; em decorrência, meio segundo depois o computador principal da missão também apresentou problemas;



[Vídeo do lançamento](#)

Falhas Famosas de Software - London Ambulance System

- London Ambulance System (LAS) - despacho de ambulâncias em Londres, 1992:
 - Morte de pessoas que não foram socorridas em tempo;
 - Problema de Gerência de Software:
 - Responsáveis contrataram uma empresa desconhecida cujo
 - valor cobrado era menor que os cobrados pelas empresas de Renome;
 - Colocaram o sistema no ar sem os devidos testes.
 - Não foi feita uma migração correta do sistema antigo para o novo.



Falhas Famosas de Software - Acelerador médico Therac-25

- **Acelerador médico Therac-25, 1985-1987:**
 - O Therac-25 era um acelerador linear empregado nos hospitais na década de 80 para tratar tumores;
 - A máquina emitia radiação de alta energia sobre células cancerosas sem causar dano ao tecido circundante;
 - Mas devido a uma falha de programação, durante um processo onde efetuavam estas correções, a máquina emitia 100 vezes mais energia do que a requerida;
 - Em consequência deste bug morreram ao menos cinco pacientes e várias dezenas sofreram os efeitos de ficarem expostos a uma elevada radiação, inclusive os próprios funcionários.



Software Architecture:
Therac-25 the killer radiation
machine

Falhas Famosas de Software - Sistema de triagem/control de bagagem do aeroporto internacional de Denver (EUA)

- Sistema de triagem/control de bagagem do aeroporto internacional de Denver (EUA):
 - Objetivo: Sistema para controlar 4000 "telecars" ao longo de 21 milhas de trilhos, transportando a bagagem entre os portões e as áreas de retirada de bagagem de 20 companhias aéreas;
- Características:
 - Uma rede de cerca de 100 computadores, 5000 "olhos eletrônicos", 400 receptores de rádio e 56 leitores de códigos de barra, organizando o transporte e entrega segura de cada mala ou valise desde a chegada ao aeroporto até a entrega ao passageiro e respectiva conferência;
- Problemas:
 - Atrasou a inauguração do aeroporto. Custo do sistema: US\$ 193 milhões;
 - Inauguração estava prevista para Out/1993. Em Junho/1994 o sistema ainda não estava funcionando e causava prejuízos de US\$ 1,1 milhão/dia;
 - No começo de 1995 um controle MANUAL de bagagem foi instalado para que o aeroporto pudesse ser inaugurado (com atraso de mais de um ano);
 - Problema de planejamento e gerenciamento.



Engenharia de Software: Definições

- Estudo ou aplicação de abordagens sistemáticas, econômicas e quantificáveis para o desenvolvimento, operação e manutenção de software de qualidade;
- Engenheiros de software devem adotar uma abordagem sistemática e organizada para seu trabalho e usar ferramentas e técnicas/métodos apropriados dependendo do problema a ser solucionado, das restrições de desenvolvimento e dos recursos disponíveis;



Engenharia de Software: Definições

- Segundo o IEEE, Engenharia de Software é a aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção do software. O estudo de abordagens e princípios a fim de obter economicamente softwares confiáveis e que executam de forma eficiente nas máquinas reais;



Engenharia de Software: Definições

- A Engenharia de software abrange um conjunto de três elementos fundamentais: Métodos, Ferramentas e Procedimentos;
- Principais objetivos:
 - Controle sobre o desenvolvimento de software dentro de **custos**, **prazos** e níveis de **qualidade** desejados;
 - Produtividade no desenvolvimento, operação e manutenção de software;
 - Qualidade versus Produtividade;



Engenharia de Software: Definições

- 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.

Engenharia de Software: Definições

- Ferramentas: dão suporte automatizado aos métodos;
- Procedimentos: constituem o elo de ligação entre os métodos e ferramentas;
 - Sequência 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.

Engenharia de Software: Crise do Software

- A crise do software foi um termo utilizado nos anos 70, quando a engenharia de software era praticamente inexistente;
- O termo expressava as dificuldades do desenvolvimento de software frente ao rápido crescimento da demanda por software, da complexidade dos problemas a serem resolvidos e da inexistência de técnicas estabelecidas para o desenvolvimento de sistemas que funcionassem adequadamente ou pudessem ser validados;
- Uma das primeiras e mais conhecidas referências ao termo foi feita por *Edsger Dijkstra*, na apresentação feita em 1972 na *Association for Computing Machinery Turing Award*, intitulada "The Humble Programmer" (EWD340), publicada no periódico *Communications of the ACM*.



Engenharia de Software: Crise do Software - Problemas

- Os problemas mais comuns no desenvolvimento de software relatados pela Crise do Software:
 - Estimativas de prazo e de custo imprecisas;
 - Produtividade das pessoas da área de software não acompanha a demanda;
 - Prazos ultrapassados;
 - Custos acima do previsto;
 - A facilidade de manutenção não era enfatizada como um critério importante, gerando assim custos de manutenção elevados;
 - Não atendimento dos requisitos do usuário;
 - 1/3 dos projetos eram cancelados;
 - 2/3 dos projetos extrapolavam o orçamento.

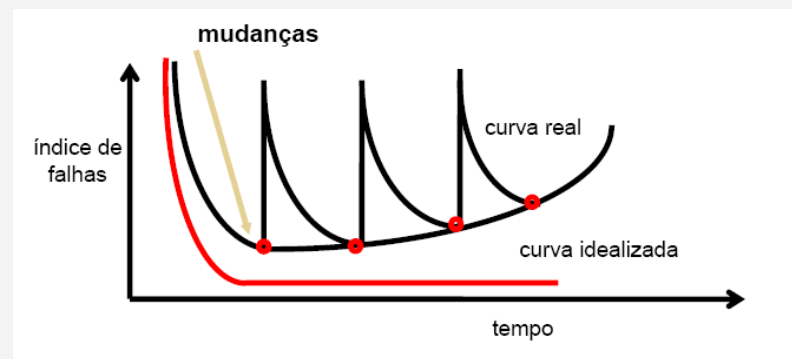
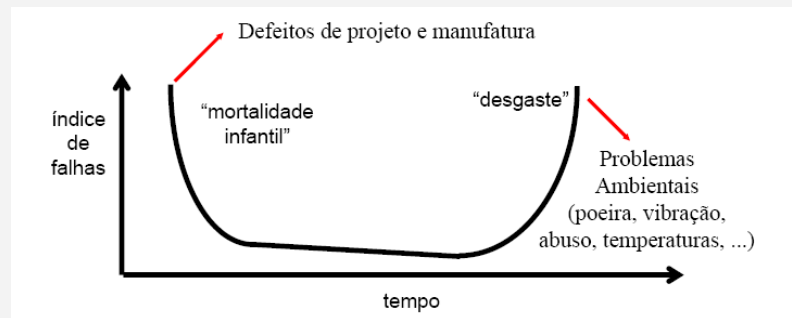
Engenharia de Software: Crise do Software - Problemas

- As estimativas de prazo e de custo frequentemente são imprecisas:
 - “Não dedicamos tempo para coletar dados sobre o processo de desenvolvimento de software”;
 - “Sem nenhuma indicação sólida de produtividade, não podemos avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões”;
- Insatisfação do cliente com o sistema concluído:
 - “Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente”;
- A qualidade de software às vezes é menos que adequada:
 - Só recentemente começam a surgir conceitos quantitativos sólidos de garantia de qualidade de software;
- O software existente é muito difícil de manter:
 - A tarefa de manutenção devora o orçamento destinado ao software;
 - A facilidade de manutenção não foi enfatizada como um critério importante.

Engenharia de Software: Causas dos problemas associados à crise de software

1- PRÓPRIO CARÁTER DO SOFTWARE:

- O software é um elemento de sistema lógico e não físico;
- Conseqüentemente o sucesso é medido pela qualidade de uma única entidade e não pela qualidade de muitas entidades manufaturadas;
- O software não se desgasta, mas se deteriora!



Engenharia de Software: Causas dos problemas associados à crise de software

2- FALHAS DAS PESSOAS RESPONSÁVEIS PELO DESENVOLVIMENTO DE SOFTWARE:

- Gerentes sem nenhuma experiência em software;
- Profissionais da área de software têm pouco treinamento formal em novas técnicas para o desenvolvimento de software;
- Resistência a mudanças.



Engenharia de Software: Causas dos problemas associados à crise de software

2- FALHAS DAS PESSOAS RESPONSÁVEIS PELO DESENVOLVIMENTO DE SOFTWARE:

- Gerentes sem nenhuma experiência em software;
- Profissionais da área de software têm pouco treinamento formal em novas técnicas para o desenvolvimento de software;
- Resistência a mudanças.



3- MITOS DO SOFTWARE: Propagaram desinformação e confusão

- Administrativos;
- Cliente;
- Profissional.



Engenharia de Software: Causas dos problemas associados à crise de software

Mito 1 - Administrativo:

- 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:

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

Engenharia de Software: Causas dos problemas associados à crise de software

Mito 2 - Administrativo:

- Meu pessoal tem ferramentas de desenvolvimento de software de última geração, afinal lhes compramos os mais novos computadores.

Realidade:

- É preciso muito mais do que os mais recentes computadores para se fazer um desenvolvimento de software de alta qualidade.

Engenharia de Software: Causas dos problemas associados à crise de software

Mito 3 - Administrativo:

- Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e tirar 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.

Engenharia de Software: Causas dos problemas associados à crise de software

Mito 1 - Cliente:

- Uma declaração geral dos objetivos é suficiente para se 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.

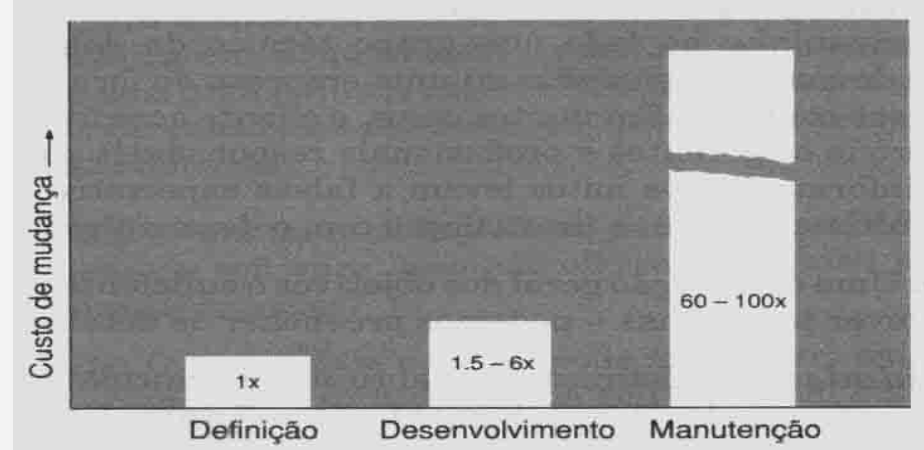
Engenharia de Software: Causas dos problemas associados à crise de software

Mito 2 - Cliente:

- 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 das Mudanças

Engenharia de Software: Causas dos problemas associados à crise de software

Mito 1 - Profissional:

- 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.

Engenharia de Software: Causas dos problemas associados à crise de software

Mito 2 - Profissional:

- 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.

Solução para a Crise do Software

Engenharia de Software

- A Engenharia de software abrange um conjunto de três elementos fundamentais: Métodos, Ferramentas e Procedimentos;
- Principais objetivos:
 - Controle sobre o desenvolvimento de software dentro de custos, prazos e níveis de qualidade desejados;
 - Produtividade no desenvolvimento, operação e manutenção de software;
 - Qualidade versus Produtividade;

O Mundo do desenvolvimento de Software - Resumindo

Crise do Software (~1970):

- Desenvolvimento de Software como “arte”;
- Problemas de execução – erros;
- Prazos extrapolados;
- Custos inesperados – correção de erros e adaptação do código às reais necessidades do usuário;
- Empresas dependentes de computadores com sistemas legados que necessitam modificações, mas com código/documentação ilegível ou inexistentes;
- Insatisfação de usuários.

O Mundo do desenvolvimento de Software - Resumindo

O que acontecia? (ou ainda acontece?)

- Falta de alinhamento de expectativas;
- Falta de preparo da equipe do projeto;
- Planejamento inexistente ou insuficiente;
- Incapacidade de prever riscos;
- Problemas de comunicação;
- Requisitos mal definidos;
- Limitação de recursos.



O Mundo do desenvolvimento de Software - Resumindo

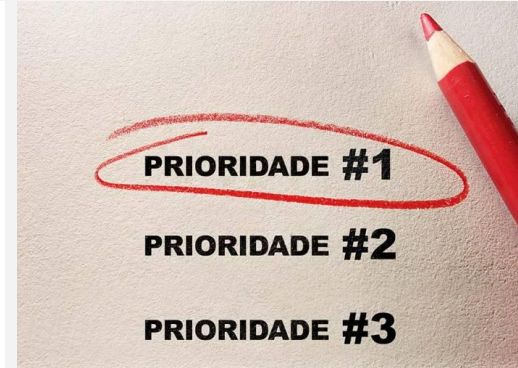
O que precisamos fazer? Onde está a bala de prata?

- Na linguagem?
- No ambiente?
- Nas notações?
- Nas ferramentas CASE?
- Nas metodologias?
- Na falta de planejamento e gerenciamento?

O Mundo do desenvolvimento de Software - Resumindo

Para que planejar?

- Para garantir que estamos sempre fazendo a coisa mais importante que se tem a fazer;
- Para coordenar a interação das pessoas;



O Mundo do desenvolvimento de Software - Resumindo

Paradoxo de Cobb - Martin Cobb - Treasury Board of Canada Secretariat

"We know why projects fail, we know how to prevent their failure - so why do they still fail?"



REFERÊNCIAS

PRESSMAN, Roger S. Engenharia de Software. Mc Graw Hill, 6 ed, Porto Alegre, 2010.

SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.