



Aula 01

Análise de Sistemas Orientado a Objetos
Prof. Me. Joseffe Barroso de Oliveira

UNIP - Universidade Paulista
Análise e Desenvolvimento de Sistemas

Introdução

- * Os computadores se tornaram elementos chaves em nossas vidas. Aos poucos, assumiram muitas das funções que nos afetam de maneira decisiva. Hoje eles controlam a maioria das transações monetárias, linhas de produção, transporte, comunicação, sistemas de defesa, sistemas de controle de processo etc.
- * Os computadores já são encontrados em quase todas as casas e, em algumas delas, já controlam a utilização de todos os aparelhos eletrônicos, coisa vista apenas em filmes, até há um tempo atrás.

Introdução

- * No entanto, computadores sozinhos são máquinas (*hardware*) inofensivas. Com o tipo certo de programa (*software*), eles podem levá-lo à Lua, literal e figuradamente.
- * É o *software* que dá vida a eles. Quando precisam desempenhar um papel tão importante, uma pequena falha tanto no *hardware* quanto no *software* pode levar a consequências desastrosas.

Introdução

- * Infelizmente, embora haja processos bem definidos, baseados em fundamentos teóricos para garantir a confiabilidade do *hardware*, não se pode dizer a mesma coisa sobre *software*.
- * Embora ainda não exista uma teoria para desenvolvimento de *software*, é necessário que o ele se comporte de maneira previsível, mesmo em situações imprevistas. Por essa razão, existe uma necessidade de gerenciar seu desenvolvimento por meio de um processo bem definido e sistemático.

Introdução

- * A antiga abordagem "codificar e testar" (*code & test*) não será suficiente. Ela pode ser boa para problemas mais simples. No entanto, o *software* precisa ser projetado de forma a lidar com situações extremamente complexas.

Crise do Software

A Crise do Software surgiu devido ao amadorismo existente na condução do processo de desenvolver software.

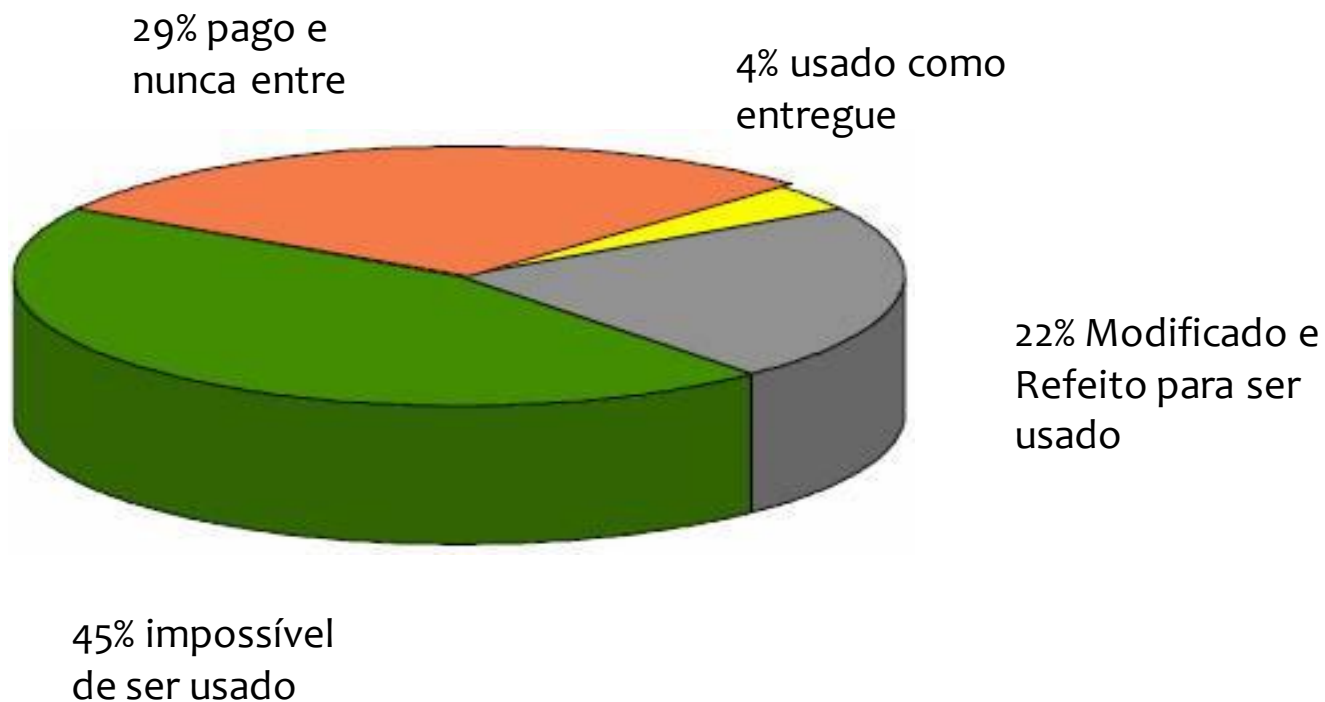
Aconteceu m meados de 1968. O termo expressava as dificuldades do desenvolvimento de software frente ao rápido crescimento da demanda existente, da complexidade dos problemas a serem resolvidos e da inexistência de técnicas estabelecidas para o desenvolvimento de sistemas.

Causas...

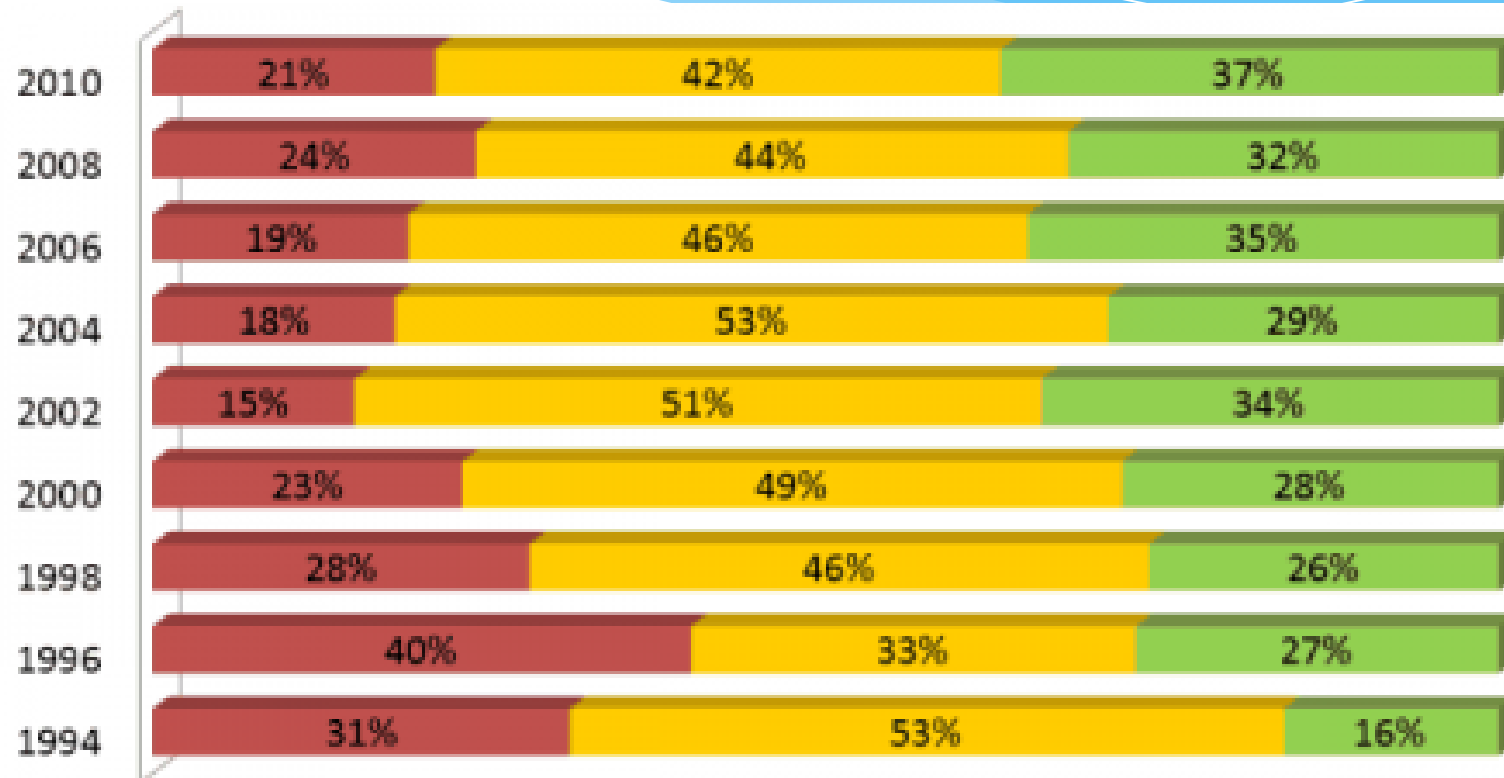
A imaturidade apresentada no desenvolvimento de sistemas foi a causa da crise do software:

- *Projetos estourando o orçamento.
- *Projetos estourando o prazo.
- *Software de baixa qualidade.
- *Software muitas vezes não atendendo os requisitos.
- *Projetos não gerenciáveis e código difícil de manter.

Quantificando o problema



Entregas em números



A crise acabou?



Como o cliente explicou



Como o líder de projeto entendeu



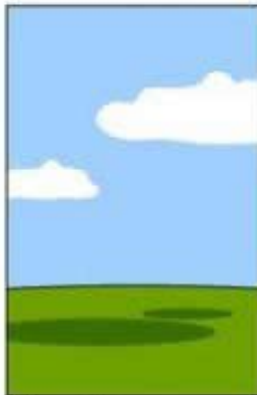
Como o analista projetou



Como o programador implementou



Como o Analista de Negócios descreveu



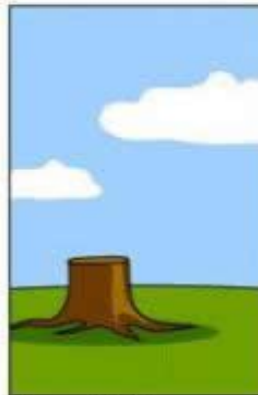
Como foi documentado



Como foi configurado



Como o cliente foi cobrado



Como foi feito o suporte



O que o cliente realmente precisava

Definição

Engenharia de Software poderia ser resumida à utilização de princípios de engenharia para o desenvolvimento de software, ou seja, levantar os requisitos associados, construir modelos para representar a solução a ser desenvolvida, implementar as diversas unidades que irão compor o produto

Sucesso em projeto de Software

“Esforço de Equipe”



- * Qualquer esforço para o desenvolvimento de *softwares* exige uma equipe de especialistas. Por exemplo, a equipe pode ser composta por especialistas em domínios, projetos, codificação, testes e *hardware*.
- * Cada grupo de especialistas deve se concentrar em um aspecto específico do problema e apresentar uma solução adequada. No entanto, nenhum grupo pode trabalhar isoladamente, pois a interação entre os membros das equipes é importantíssima.

Sucesso em projeto de Software

“Metodologia”



- * Existem dois tipos de metodologias de desenvolvimento:
 - * Voltada ao procedimento
 - * Voltada ao objeto
- * Embora, teoricamente, ambas possam ser usadas em qualquer situação de problema, uma delas deve ser selecionada antecipadamente.

Sucesso em projeto de Software

“Documentação”



- * Uma documentação clara, objetiva e precisa dos componentes do processo de desenvolvimento, é crucial para o sucesso de qualquer projeto de software.
- * Comunicação verbal e desenhos do pacote não são suficientes para se entender tal processo. Por exemplo, a documentação é essencial para a aprovação do cliente em várias etapas do processo.
- * Uma vez desenvolvido, o software cria vida. Entretanto, durante sua vida, ele passará por várias mudanças e uma documentação bem feita é essencial para a realização dessas alterações de forma mais eficaz.

Sucesso em projeto de Software “Planejamento”



- * Assim que o desenvolvimento do *software* ocorrer de acordo com os requisitos especificados pelo cliente, é necessário que todo o esforço seja adequadamente estimado para atender as restrições de prazo e custo.

Sucesso em projeto de Software

“Garantia de Qualidade”



- * Os clientes esperam retorno financeiro com a implantação do *software*.
- * Além de atender às necessidades do cliente, o *software* deve, necessariamente, atender aos padrões de qualidade que podem ser em relação a desempenho, segurança etc.

Sucesso em projeto de Software

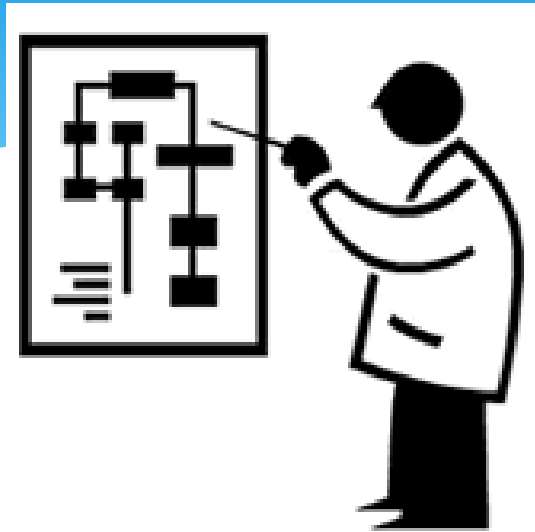
“Usuário Leigo”



- * Usuários leigos são aqueles que não dominam computação, portanto, o *software* deve ter uma interface gráfica que favoreça seu manuseio.

Sucesso em projeto de Software

“Ferramentas”



- * A documentação é importante para um projeto de desenvolvimento de *software*, mas é uma tarefa onerosa e muitos desenvolvedores desistem de fazê-la.
- * Existem ferramentas que são conhecidas como ferramentas de Engenharia de Software Assistida por Computador (*Computer Aided Software Engineering – CASE*) que simplificam o processo da documentação.

Sucesso em projeto de Software

“Conformidade com Padrões”



- * Os padrões são necessários para garantir uma documentação clara, objetiva e sem ambiguidades.
- * Por exemplo, padrões IEEE para especificações de requisitos, projetos etc.
- * Também é possível que um cliente especifique os padrões a serem usados.

Sucesso em projeto de Software

“Reutilização”



- * O esforço de desenvolvimento pode ser otimizado, reutilizando-se componentes já testados, como por exemplo, bibliotecas matemáticas, kits de ferramentas de interface gráficas etc.

Sucesso em projeto de Software

“Manutenção de Software”



- * Todo *software* precisa ser modificado periodicamente, de acordo com as solicitações dos clientes e do uso de novas tecnologias.
- * A equipe de desenvolvimento pode não estar disponível para realizar a manutenção do pacote. Portanto, uma equipe de suporte deverá ser reunida, sempre que necessário, para garantir que o *software* continue a fornecer os serviços necessários.

Sucesso em projeto de Software

“Gerenciamento de Alteração”



- * Sempre que uma alteração deva ser realizada, é necessário estudar seu impacto sobre os vários componentes do *software*.
- * Por exemplo, ao modificar o tipo da variável denominada Global, toda função que utilizá-la será impactada e, a menos que se tome cuidado para minimizar esse efeito, o *software* poderá ter seu desempenho comprometido.

Sucesso em projeto de Software

“Controle de Versão”



- * O *software* está propenso a frequentes alterações durante seu desenvolvimento. Portanto, é importante que o usuário obtenha a versão mais recente.
- * No caso de falhas, será possível recorrer às versões anteriores.

Sucesso em projeto de Software

“Gerenciamento de Risco”



- * Todo esforço de desenvolvimento de software está sujeito a riscos. Por exemplo, indisponibilidade de especialistas, tecnologia, recursos etc.
- * É necessário, portanto, avaliar constantemente os riscos e criar medidas para reduzi-los.