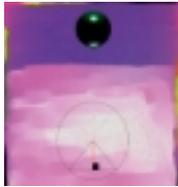




Engenharia de Software



Conceitos básicos sobre E.S:



- **Ambiência**
- **Caracterização do software**
- **Fases de desenvolvimento**



Introdução

Aspectos Introdutórios

Crise do Software

Definição de Engenharia do Software





Crise do Software

Nos primórdios da era computacional, os programas eram pequenos e normalmente escritos por uma pessoa e utilizados por especialistas na área. Os problemas a resolver eram de natureza técnica, lidando com dados numéricos, sendo a ênfase colocada na codificação eficiente de algoritmos bem conhecidos.

Na actualidade, as aplicações são muito diferentes, em vários aspectos:

- São grandes;
- desenvolvidas por equipas muitas vezes pluridisciplinares e durante períodos longos;
- os programadores não são os futuros utilizadores e não são peritos na área em que se insere a aplicação;
- estamos em presença de aplicações OLTP, em ambientes cliente-servidor ou centrados na rede.

Estas alterações ocorridas no ambiente em que se insere o software, levou a que as técnicas de programação ficassem claramente para trás, dado que agora, o software é muito mais extenso e complexo.

Em resultado disso, foram inevitáveis:

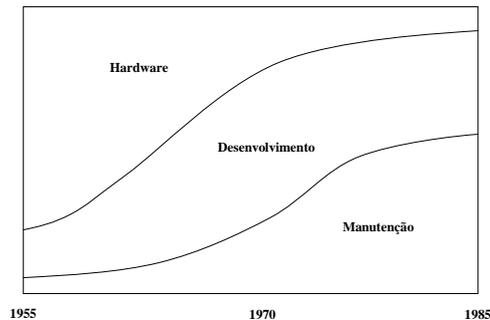
- Imprecisões nas estimativas de custos e duração;
- deficiências na identificação dos requisitos, levando a que as aplicações não se comportassem como esperado;
- falta de produtividade das equipas;
- falta de qualidade e fiabilidade do software, sendo muitos erros detectados após entrega;
- grande dificuldade de manutenção, dado que a docum. é deficiente e difícil fazer alterações;
- carência de profissionais competentes no mercado mundial (e excesso de prof. de comp. insatisfatória).



Crise do Software

Dos pontos anteriores, ressaltam-se a questão da pontualidade, qualidade e custos, dado que é destas que dependem os sistemas que desenvolvemos e que suportam m grande medida a qualidade da vida actual.

- **Pontualidade:** sem ela quaisquer tentativas de temporização de tarefas torna-se impossível, tendo como consequência lesar as duas questões seguintes;
- **Qualidade:** tem a ver com o comportamento esperado, ausência de erros, facilidade de utilização e desempenho, entre outros factores;
- **Custos:** são enormes e tiveram uma evolução como se mostra na figura abaixo.





Definição de Engenharia de Software

Esta crise motivou conferências nos finais dos anos 60 e início dos anos 70, onde surgiu o termo “Engenharia de Software”, criado com conotações um tanto provocatórias, sugerindo a necessidade de desenvolver software de forma análoga à forma como a engenharia civil constrói casas ou pontes:

- Iniciar com uma base teórica;
- Utilizar técnicas de desenho sólidas e comprovadas.

Definição de Engenharia de Software:

- “É o estabelecimento e utilização de princípios sólidos de engenharia por forma a ser criado software económico, fiável e que trabalhe em máquinas reais” (Segundo conferência da NATO)
- “É a aproximação sistemática a desenvolvimento, operação, manutenção e substituição do software” (Definição dada no glossário do IEEE)
- Conceber, construir e manter grandes sistemas de software



Definição de Engenharia e Software

Aspectos fundamentais a reter:

- A E.S. lida com a criação de grandes programas (várias pessoas por períodos superiores a 1/2 ano - a dita “programação em grande”);
- O tema central é o domínio da complexidade, por divisão do problema, como comunicação simples entre as partes;
- A cooperação entre as pessoas é parte integrante da programação em grande;
- A inevitabilidade da evolução do software - como o software modela em parte a realidade, mudando esta, terá de mudar aquele;
- A eficiência com que software é desenvolvido é de importância crucial, dada a necessidade de minimizar os custos envolvidos;
- O software tem de suportar efectivamente os utilizadores, devendo ser desenvolvido não só de forma correcta, mas ser criado o sistema certo, tendo em mente a utilidade, facilidade de utilização, fiabilidade, tempo de resposta, manuais de utilização e treino dos utilizadores.





Definição de Engenharia e Software

Aspectos fundamentais a reter (continuação):

- A E.S. permite modelar parte do mundo real em software. Esses modelos são grandes, abstractos e complexos. Dessa forma devem tornar-se visíveis em documentos, tais como concepção de sistema, manuais do utilizador, etc.
- A produção de documentos é tão relevante do ponto de vista da E.S. como a própria programação.
- A E.S. é diferente de outras disciplinas de engenharia. Não está constringida por materiais governados por leis físicas ou por processos de manufactura.



Evolução na Produção de Software

Agora, passados 30 anos, fizeram-se progressos enormes em E.S.:

- A capacidade de produzir software eficiente e de confiança melhorou marcadamente
- Temos uma muito melhor compreensão das actividades envolvidas na produção de software
- Desenvolveram-se métodos de especificação, concepção e implementação de software
- Novas notações e ferramentas reduzem o esforço requerido à produção de sistemas grandes e complexos.

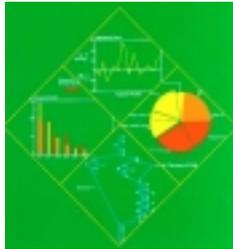
Apesar disso:

- Muitos projectos continuam a atrasar-se ou a ultrapassar o orçamento
- É distribuído e instalado software que não satisfaz as necessidades reais dos clientes
- Novas tecnologias que resultam da convergência dos computadores e comunicações colocam novos desafios aos engenheiros de software





Generalidades sobre o Software



Caracterizar o software

Produto
Processo

Propriedades
Princípios



Tipos de Produtos de Software

- **Produtos Genéricos (Standard)**
 - Sistemas do tipo Stand-Alone, que são produzidos por uma organização que desenvolve software e vendidos no mercado a qualquer cliente. As especificações são produzidas internamente, sendo de âmbito alargado, reflectindo o que se pensa irá vender.
- **Produtos Específicos**
 - Sistemas que são contratados por um cliente específico e desenvolvidos especialmente pelo contratante. As características são definidas detalhadamente e as alterações são negociadas.
- **A maioria dos gastos de software cai na categoria dos produtos genéricos, mas dado o cada vez maior acoplamento de dispositivos hardware aos computadores, há uma cada vez maior procura de controladores de software com a consequente necessidade crescente de software específico.**





Atributos de Produtos de Software

- São características mostradas pelo produto depois de instalado e em uso; não são serviços proporcionados pelo produto.
- São elas:
 - Facilidade de Manutenção - deve ser possível que o software evolua por forma a responder à alteração dos requisitos;
 - Confiança - O software não deve causar estragos físicos ou económicos no caso de ocorrência de uma falha;
 - Eficiência - O software não deve desperdiçar a utilização dos recursos;
 - Facilidade de Utilização - O software deve ter uma interface apropriada com o utilizador e respectiva documentação.

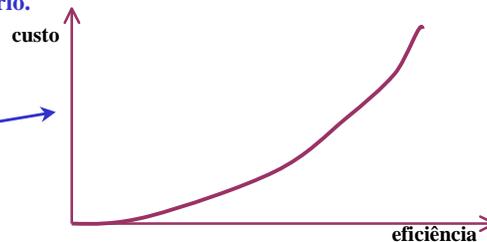


Importância das Características do Produto

- A importância relativa destas características depende do produto e ambiente em que é utilizado.
- A otimização de todos os atributos é difícil dados que alguns são mutuamente exclusivos: ex. uma melhor interface reduz a eficiência do sistema.
- Em alguns casos, um ou outro atributo pode dominar;
 - exemplo: Nos sistemas para aviação, os atributos eficiência e confiança são primordiais.
- Os custos tendem crescer exponencialmente se um nível muito alto de um qualquer atributo for necessário.

Lei da diminuição do retorno

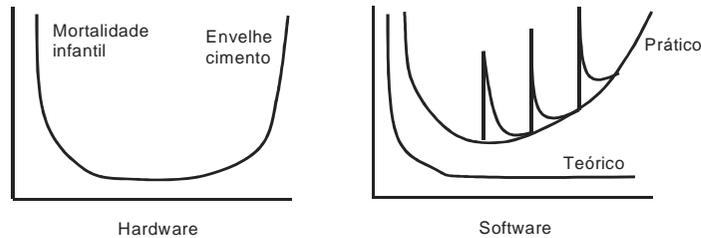
- A relação entre os custos e melhoramentos não é linear



Software

Caracterização do produto

- Propriedades genéricas
 - **Maleabilidade** - O software pode e muitas vezes precisa de ser modificado (ao contrário de uma ponte ou avião)
 - **Produção** - O software é desenhado e implementado por pessoas, ao invés de ser manufacturado como o hardware
 - **Vida** - O software deteriora-se de modo diferente do hardware



Software

- Outras propriedades:
 - **Correcção** - Funciona de acordo com a especificação (requisitos funcionais). Propriedade matemática
 - **Confiabilidade** - Um software é confiável se o utilizador pode *depende* dele. Propriedade estatística
 - **Robustez** - Um software é robusto se desempenha as funções esperadas mesmo em situações não antecipadas. Propriedade mal definida
 - **Desempenho** - Espera-se que o software cumpra um certo nível de desempenho. O mesmo que eficiente: económico na utilização de recursos
 - **Amigável** - Fácil de utilizar. Propriedade subjectiva
 - **Verificável** - Um software é verificável se as suas propriedades podem ser verificadas (por análise ou teste) com facilidade. É uma propriedade importante quando a aplicação é crítica



Software

- Outras propriedades:
 - **Manutenção/Evolução** - Modificável após a produção
 - Reparável - Correção de defeitos num limite de tempo
 - Alterável - Alterações de requisitos funcionais
 - **Compreensibilidade** - Facilidade em analisar o código. Permite atingir outras propriedades, como alterável e verificável
 - **Reutilização** - Software utilizável em outros produtos
 - **Portabilidade** - Software utilizável em ambientes ou plataformas diferentes
 - **Interoperabilidade** - Capacidade para coexistir e cooperar com outros sistemas



Propriedades e Qualidade de Software

- As propriedades do Software, de que derivam directa ou indirectamente a qualidade do software, como já vimos, interactuam. Esta interacção pode ter um impacto positivo ou negativo.

	Correcção	Fiabilidade	Eficiência	Integridade	Utilização	Manutenção	Teste	Flexibilidade	Portabilidade	Reutilização	Interoperacionalidade
Correcção											
Fiabilidade	+										
Eficiência											
Integridade			-								
Utilização	+	+	-	+							
Manutenção	+	+	-		+						
Teste	+	+	-		+	+					
Flexibilidade	+	+	-	-	+	+	+				
Portabilidade			-	-		+	+				
Reutilização		-	-	-		+	+	+	+		
Interoperacionalidade			-	-					+		





Software

Caracterização do processo

- Propriedades:
 - **Produtividade** - Medida da eficiência do processo
 - **Controlo** - Capacidade para entregar o software num determinado prazo
 - **Visibilidade** - Um processo é visível se todos os seus passos forem documentados
- Princípios:
 - Diversidade de métodos e técnicas que devem ser seguidas de modo a atingir as propriedades desejadas quer para o produto quer para o processo



Princípios

- **Rigor e formalidade** - Aproximação destinada a melhorar a confiabilidade do produto, produtividade e controlo do processo
- **Separação de preocupações** - Separar os diferentes aspectos de um problema de modo a tratá-los separadamente
 - Separação no tempo - Controlar interdependências
 - Separação de qualidades - Por exemplo, desenhar software correcto e mais tarde aumentar a sua eficiência
 - Separação de vistas - Por exemplo, analisar separadamente fluxos de dados e de controlo
 - Separação em partes - Modularidade





Princípios

- **Modularidade** - Decompor o software em partes (lat. *divide et impera*). Destinada a aumentar a compreensibilidade, reutilização e portabilidade do software. Deve caracterizar-se por:
 - Elevada **coesão** dos elementos de um módulo
 - Reduzida **interdependência** entre módulos
- **Abstracção** - Processo destinado a identificar os aspectos importantes e ignorar os detalhes
 - Princípio largamente utilizado em linguagens de programação, algoritmos, modelos
- **Antecipação das alterações** - Facilitar a reparação e alteração do software. Relacionado com:
 - Desenho do software
 - Controlo de versões



Princípios

- **Generalidade** - Tentar descobrir o problema mais genérico por detrás do problema a resolver
 - Compromisso entre generalidade, eficiência, custo e controlo
- **Incrementalidade** - Caracteriza um processo que se desenvolve iterativamente
 - Prototipagem

