

Engenharia de Software

Seção 1 – Introdução e Conceituação

Objetivos

O aluno deverá reconhecer conceitos básicos, motivação e abrangência da Engenharia de Software.

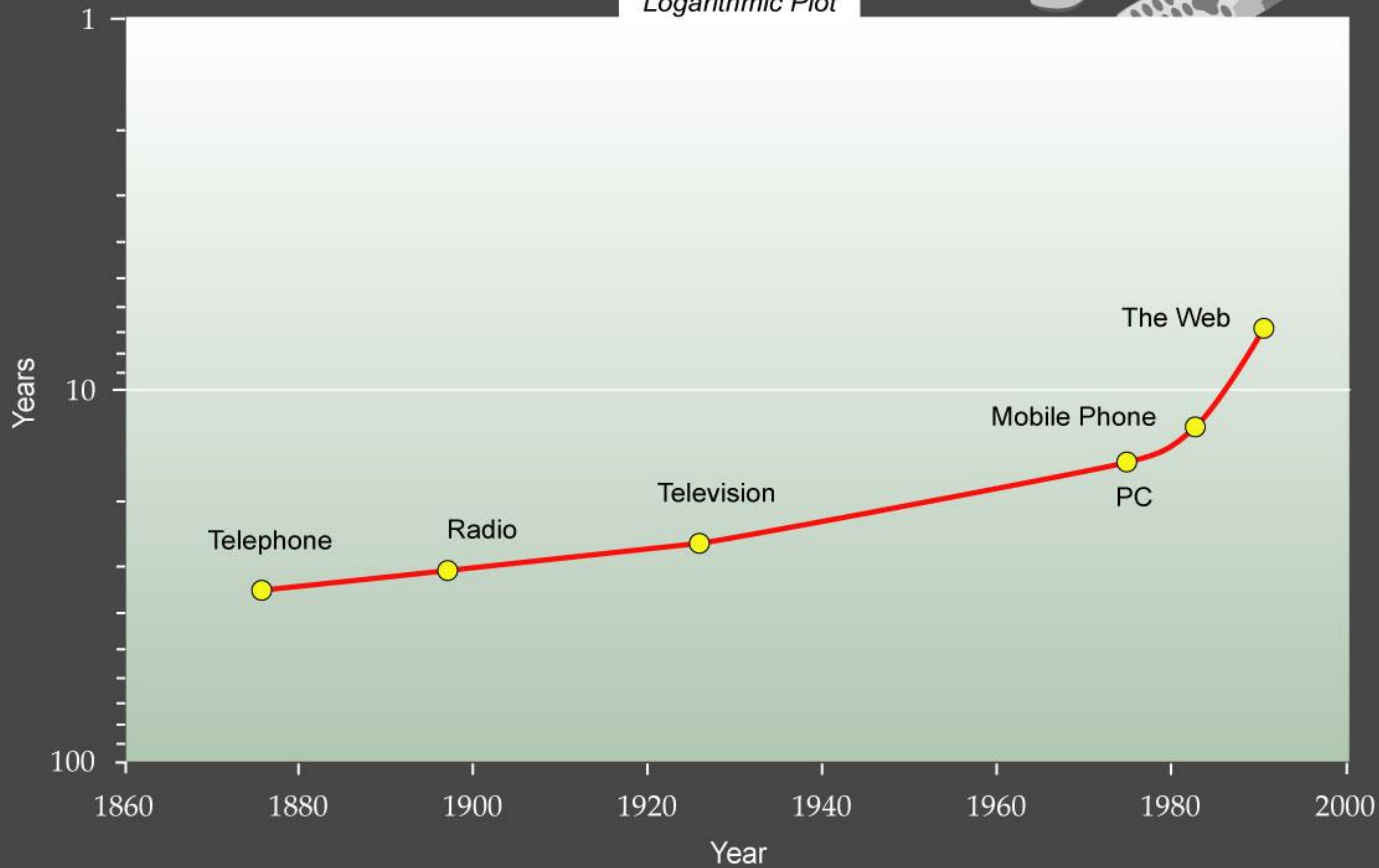


Mass Use of Inventions

Years Until Use by $\frac{1}{4}$ U.S. Population



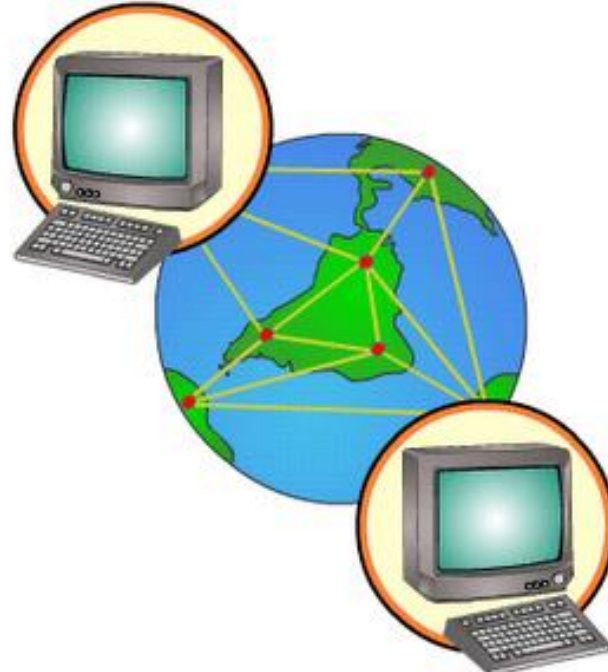
Logarithmic Plot



Software

- ✓ Grande parte das mudanças tecnológicas que vivemos são devidas a software.
- ✓ Software cria negócios.

Fonte: By Marus (Marus) [Public domain], via Wikimedia Commons



Fonte: By Jcarranza (Own work) [Public domain], via Wikimedia Commons

Software (continuação)

- ✓ Software destrói tecnologias.



Fonte: By Golf Bravo (Own work), via Wikimedia Commons



Fonte: By Olaf (Own work) [Public domain], via Wikimedia Commons

Software (continuação)

- ✓ Ninguém, há 50 anos, imaginaria que o software estaria tão presente em nossas vidas;
- ✓ Ninguém imaginaria que software pudesse ser comprado em supermercados;
- ✓ Ninguém imaginaria que a informação estaria tão democratizada por uma rede mundial de computadores;
- ✓ E ninguém imaginaria que se gastaria mais consertando do que desenvolvendo software...

Software (continuação)

O que é Software?

- ✓ Instruções (programas de computadores) que, quando executadas, fornecem as características, função e desempenho desejados;
- ✓ Estruturas de dados que permitem aos programas manipular adequadamente a informação;
- ✓ Documentos que descrevem a operação e uso dos programas.

Software (continuação)

- ✓ Software é **desenvolvido**, não produzido no sentido clássico como acontece com hardware:
 - ✓ Alta qualidade é obtida com bom projeto;
 - ✓ Diferentes abordagens de construção do “produto”;
 - ✓ Relacionamento dos processos de projeto com as pessoas é diferente nos dois casos;
 - ✓ Custos de software são concentrados na engenharia;

Software (continuação)

- ✓ Software **não se desgasta**:
- ✓ Em hardware, a taxa de falhas se comporta com o tempo como na figura abaixo:

Software (continuação)

- ✓ Inicialmente, existe um número relativamente alto de falhas, usualmente por problemas de projeto ou de fabricação (“mortalidade infantil”).
- ✓ O número de falhas mantém-se baixo por um tempo, a vida útil do hardware.
- ✓ Ao final da vida do hardware, o número de falhas volta a aumentar devido ao **desgaste** das peças.

Software (continuação)

- ✓ Em software, a taxa de falhas se comporta com o tempo como na figura abaixo:

Software (continuação)

- ✓ Defeitos não detectados causam altas taxas de falhas no início da vida do software.
- ✓ Modificações introduzidas no software causam um aumento do número de falhas, devido a novos defeitos.
- ✓ Nem todos os defeitos introduzidos são corrigidos.
- ✓ Logo, software não se desgasta, mas se deteriora...

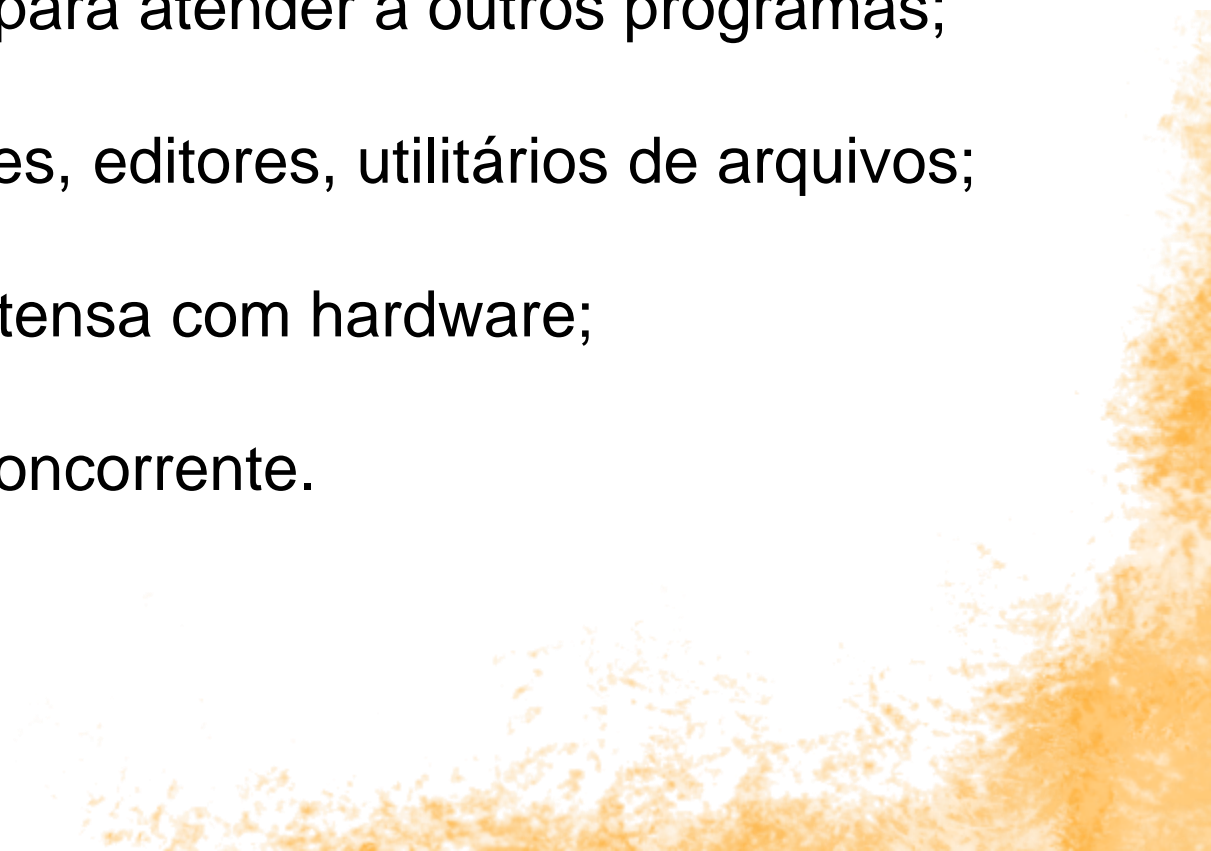


Software (continuação)

- ✓ A componentização ainda é restrita
 - ✓ Hardware é construído baseado em componentes de “prateleira”, selecionados pelo projetista;
 - ✓ Cada componente tem função e interfaces bem definidas;
 - ✓ Boa parte do software ainda é encomendado, não construído com o uso de componentes.

Categorias de Software

- ✓ Software de Sistemas

- ✓ Programas para atender a outros programas;
 - ✓ Compiladores, editores, utilitários de arquivos;
 - ✓ Interação intensa com hardware;
 - ✓ Operação concorrente.
- 
- A decorative orange splash or cloud-like graphic is located in the bottom right corner of the slide, extending from the right edge and slightly upwards.

Categorias de Software

(continuação)

✓ Software Aplicativo

- ✓ Programas dedicados a resolver uma necessidade de negócio específica;
- ✓ Processamento de dados de forma a facilitar decisões de negócio;
- ✓ Processamento de ponto-de-venda, controle de produção;

Categorias de Software

(continuação)

- ✓ Software Científico e de Engenharia
 - ✓ Programas dedicados a resolver problemas numéricos complexos, com uma quantidade de dados grande;
 - ✓ Software para astronomia, meteorologia, etc;
 - ✓ Atualmente a solução de muitos problemas destas áreas está saindo do software puramente matemático e se direcionando para a simulação e modelagem;

Categorias de Software

(continuação)

- ✓ **Software Embarcado**
 - ✓ Programas que residem em uma unidade microprocessada, com funções muito específicas;
 - ✓ Controle de teclado de forno de microondas, computadores de bordo automotivos;



Categorias de Software

(continuação)

✓ Software de Linha

- ✓ Programas de propósito relativamente geral, projetados para prover uma capacidade específica a muitos diferentes usuários;
- ✓ Pode endereçar um nicho de mercado específico (produtos de controle de estoque) ou mercados de consumo de massa (processamento de texto, planilhas, etc.).

Categorias de Software

(continuação)

- ✓ Aplicações Web

- ✓ Programas que permitem acesso e utilização através da internet ou apenas dentro de uma intranet, baseando-se em páginas dinâmicas;
- ✓ Aplicações de e-commerce;



Categorias de Software

(continuação)


- ✓ Software de Inteligência Artificial
 - ✓ Programas que fazem uso de algoritmos não-numéricos para resolver problemas complexos que não são passíveis de computação ou análise direta;
 - ✓ Exemplos: robótica, sistemas especialistas, reconhecimento de padrões de imagem e de voz, redes neurais, jogos.

Categorias de Software

(continuação)

- ✓ Uma categoria de software não necessariamente inviabiliza ou exclui outra na mesma aplicação;
- ✓ Usualmente existem mesclagens:
 - ✓ Várias aplicações de negócio utilizam técnicas e algoritmos de Inteligência Artificial;
 - ✓ Google Docs é um exemplo de software de linha com técnica de Web Application;
 - ✓ Jogos são software de linha com uso de técnicas de Inteligência Artificial;
 - ✓ Software científico pode ser embarcado, dependendo da plataforma em que estará.

Engenharia de Software

- ✓ “Criação e utilização de sólidos princípios de engenharia a fim de obter softwares econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais. “ (Fritz Bauer, 1969)
 - ✓ Aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, manutenção e operação do software, isto é, a aplicação da engenharia ao software. (IEEE, 1993)
- 

SWEBOK

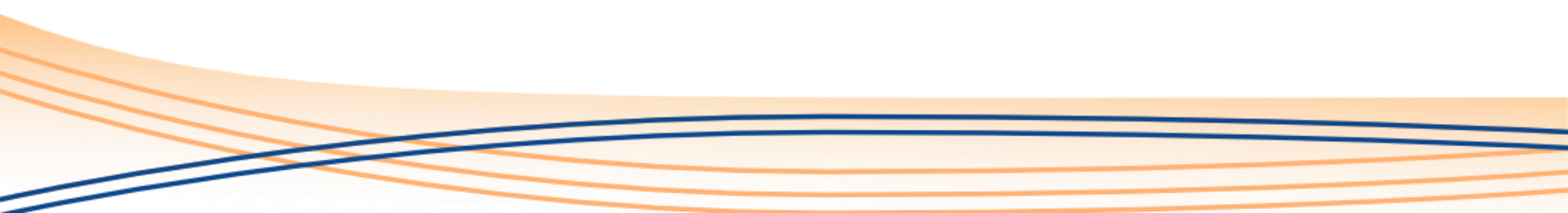
- ✓ Visando organizar o corpo de conhecimento (Body of Knowledge) da Engenharia de Software, o IEEE compilou o SWEBOK: Guia para o Corpo de Conhecimento da Engenharia de Software.
- ✓ O guia procura contribuir para a criação de uma profissão “Engenheiro de Software”, pelo fato de o SWEBOK representar um amplo consenso sobre o que este profissional deveria saber.
- ✓ O Guia **não é** o corpo de conhecimento. Este está disponível pela bibliografia existente.

Objetivos do SWEBOK

- ✓ Promover uma visão consistente da Engenharia de Software;
- ✓ Clarear o lugar e as fronteiras da Engenharia de Software com relação a outras disciplinas;
- ✓ Caracterizar o conteúdo da disciplina de Engenharia de Software;

Objetivos do SWEBOK

(continuação)

- ✓ Prover um acesso de tópicos ao Corpo de Conhecimento da Engenharia de Software;
 - ✓ Prover uma fundação para desenvolvimento de currículo e para material de certificação e licenciamento.
- 

Organização do SWEBOK

- ✓ O SWEBOK foi dividido em 10 Áreas de Conhecimento (Knowledge Areas), mais uma área especial que engloba o conhecimento necessário em outras disciplinas que não a Engenharia de Software;
- ✓ Adicionalmente, o conhecimento mapeado pelo SWEBOK foi dividido em duas categorias:
 - ✓ Conhecimento geralmente aceito
 - ✓ Conhecimento avançado e pesquisa

KA's do SWEBOK

- ✓ Requisitos de Software
- ✓ Projeto de Software
- ✓ Construção de Software
- ✓ Teste de Software
- ✓ Manutenção de Software
- ✓ Gerenciamento da Configuração de Software
- ✓ Gerenciamento da Engenharia de Software
- ✓ Processo de Engenharia de Software
- ✓ Ferramentas e Métodos de Engenharia de Software
- ✓ Qualidade de Software
- ✓ Conhecimentos de disciplinas relacionadas