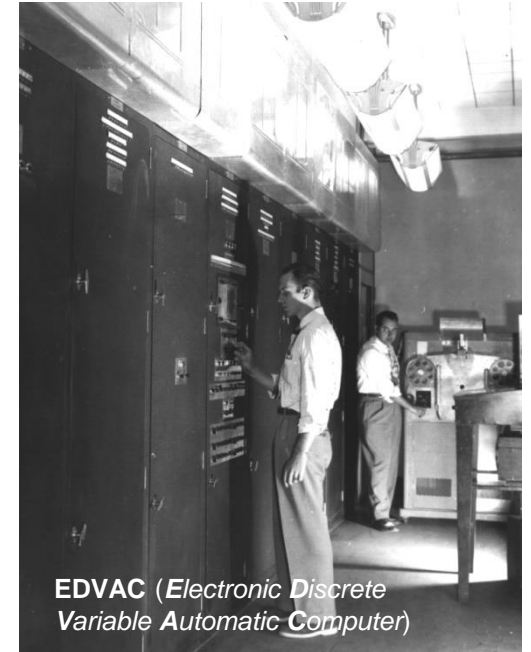
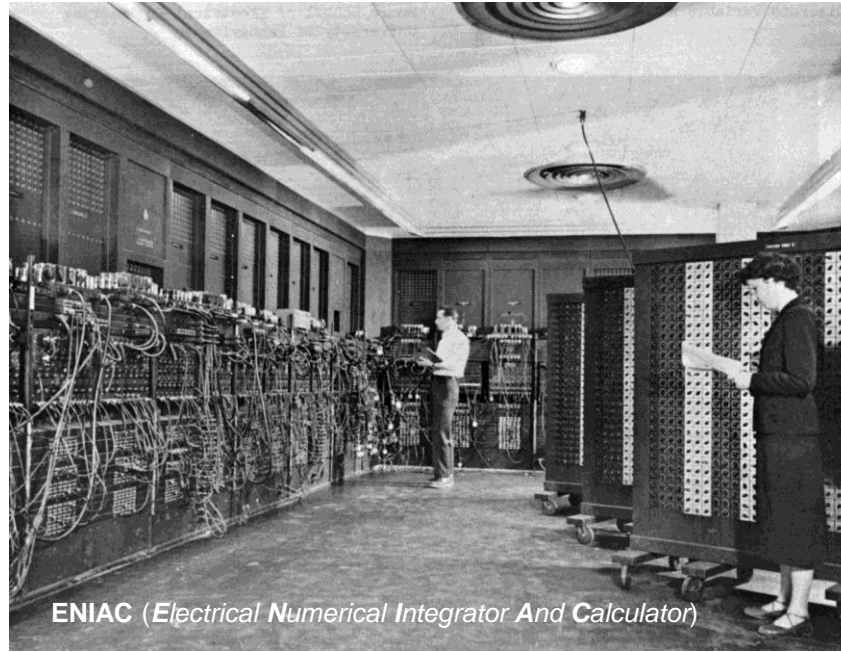
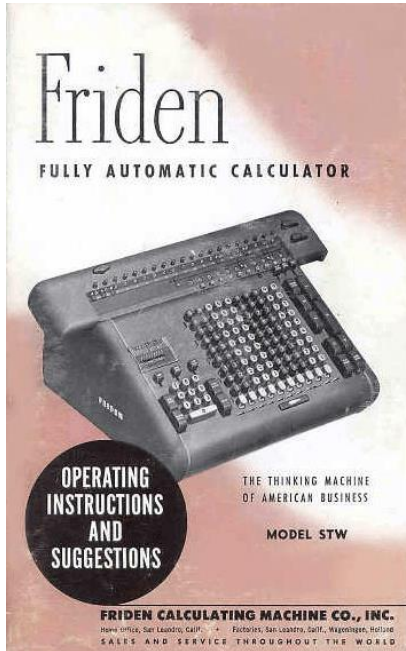

Engenharia de Software

Introdução

Luís Morgado

Instituto Superior de Engenharia de Lisboa
Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Software



Programa

- Conjunto de instruções e de dados que determinam a operação de um computador

Software

- Especificação de configuração de um sistema computacional que determina a sua operação



“Computer software continues to be the single most important technology on the world stage”

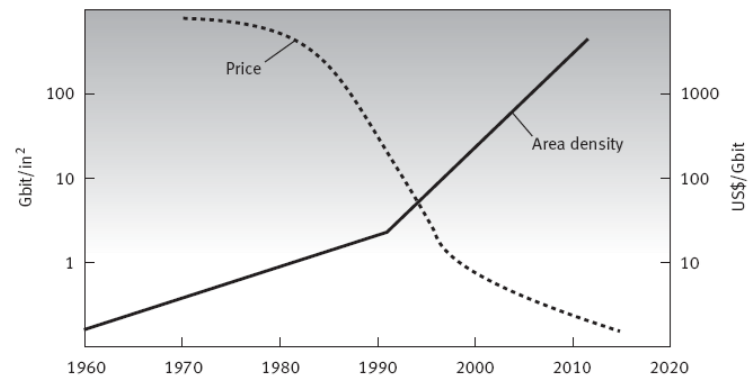
[Pressman & Maxim, 2019]

Evolução tecnológica

- Miniaturização
- Interligação
- Massificação
- Ubiquidade
- Volatilidade
- Complexidade
- Dependência crítica

Exemplo: Memória secundária

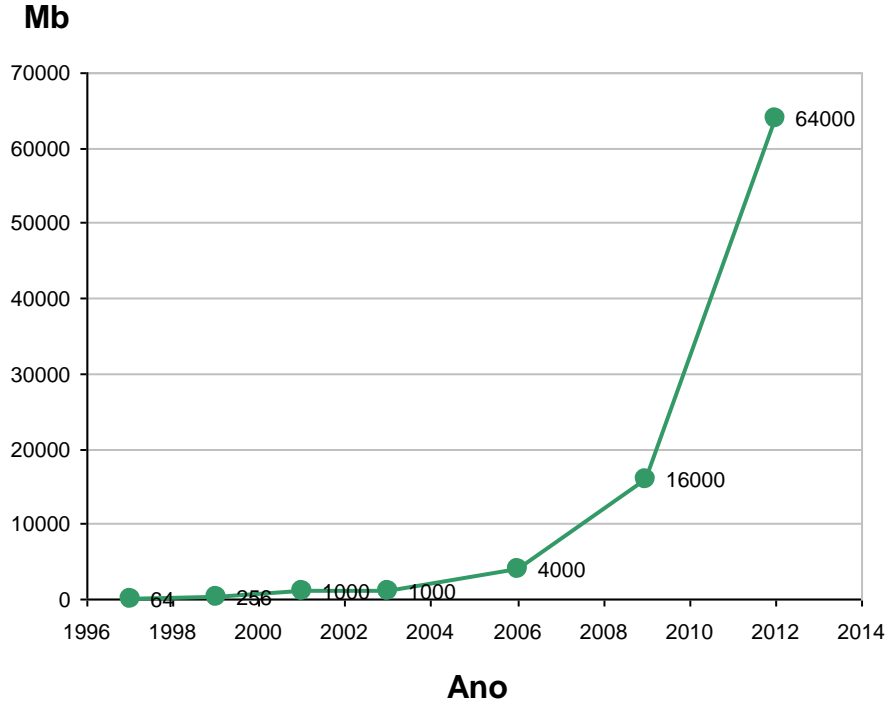
- Redução de custo
- Aumento da capacidade (exponencial)



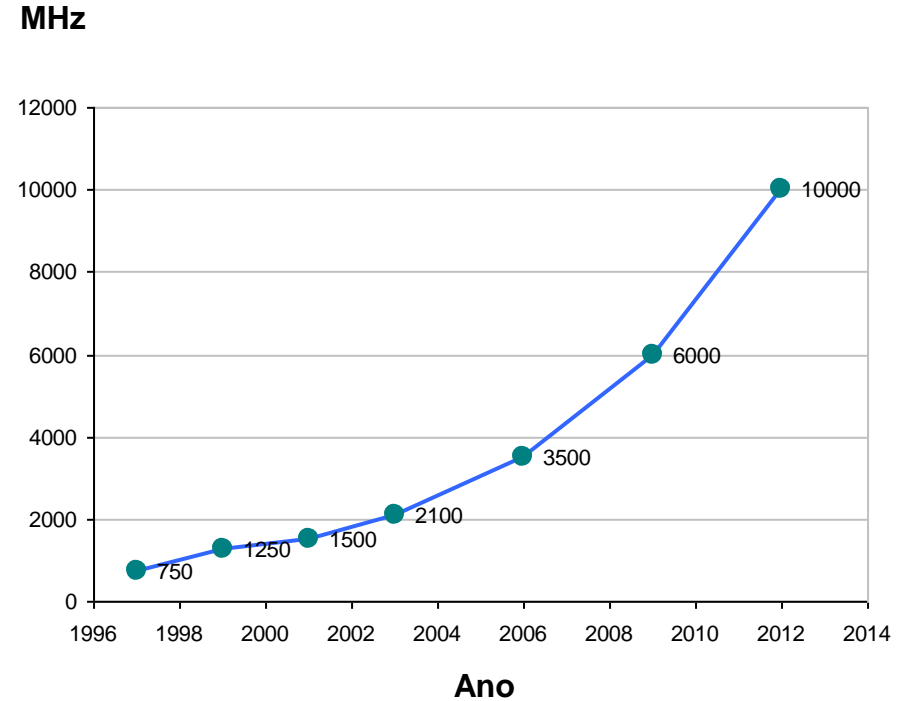
Memória secundária [Endres & Rombach, 2003]

Evolução Tecnológica

Evolução Exponencial



Evolução da quantidade de memória por circuito integrado [ITRS, 2001]



Evolução da frequência de operação em microprocessadores [ITRS, 2001]

Evolução Exponencial



IBM Real-Time Computer Complex - NASA Manned Spacecraft Center
Década de 1960



Hoje

Exemplo: Indústria Automóvel

SPECTRUM

This Car Runs on Code

By Robert N. Charette



IMAGE: DAIMLER

The avionics system in the F-22 Raptor, the current U.S. Air Force frontline jet fighter, consists of about 1.7 million lines of software code. The F-35 Joint Strike Fighter, scheduled to become operational in 2010, will require about 5.7 million lines of code to operate its onboard systems. And Boeing's new 787 Dreamliner, scheduled to be delivered to customers in 2010, requires about 6.5 million lines of software code to operate its avionics and onboard support systems.

These are impressive amounts of software, yet if you bought a premium-class automobile recently, "it probably contains close to 100 million lines of software code," says Manfred Broy, a professor of informatics at Technical University, Munich, and a leading expert on software in cars. All that software executes on 70 to 100 microprocessor-based electronic control units (ECUs) networked throughout the body of your car.

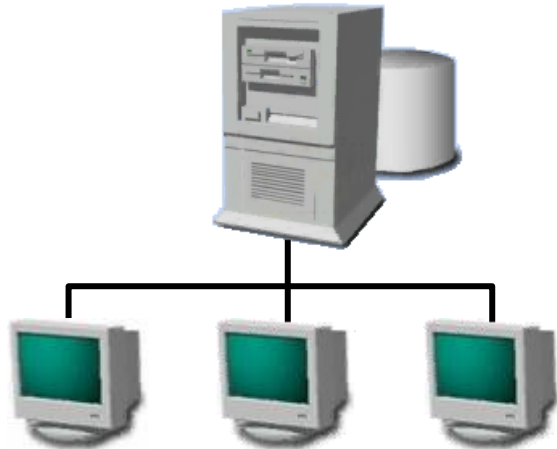
Evolução de Paradigmas de Computação

A par com a evolução tecnológica, também os paradigmas de computação evoluíram ao longo do tempo

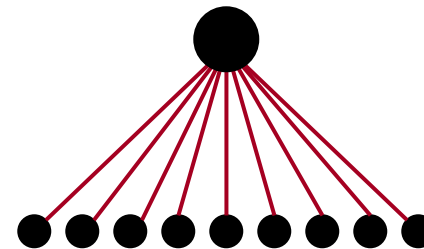
- \approx 1970: Computação centralizada com arquitecturas monolíticas, aplicações executadas num servidor (*mainframe*), sendo a interacção com o utilizador realizada através de terminais alfanuméricos
- \approx 1980: Computação descentralizada, computadores pessoais pouco dispendiosos puderam ser dedicados a utilizadores específicos, computadores menos dispendiosos ao nível departamental
- \approx 1990: Computação em rede, com aumentos significativos das capacidades de processamento e de armazenamento de dados, a par da utilização em larga escala da Internet, meios de comunicação e processamento heterogéneos, sendo a interoperabilidade um dos principais requisitos
- \approx 2000: Evolução e massificação da computação em rede

Evolução de Paradigmas de Computação

Computação Centralizada



Organização hierárquica simples

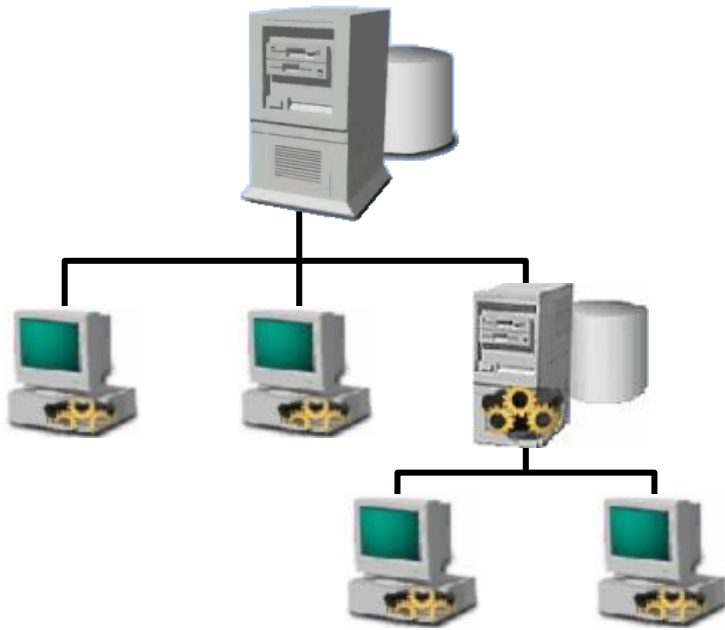


Principais Características:

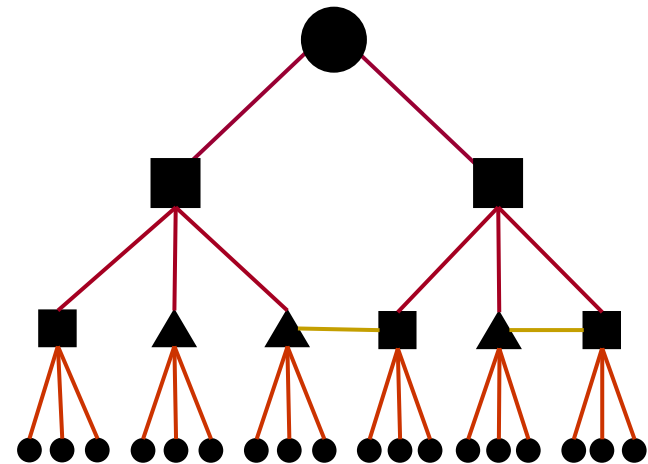
- Um computador muitos utilizadores
- Computadores acessíveis apenas a grandes organizações
- Interação com utilizador através de terminais alfanuméricos
- Arquitectura aplicacional monolítica

Evolução de Paradigmas de Computação

Computação Descentralizada



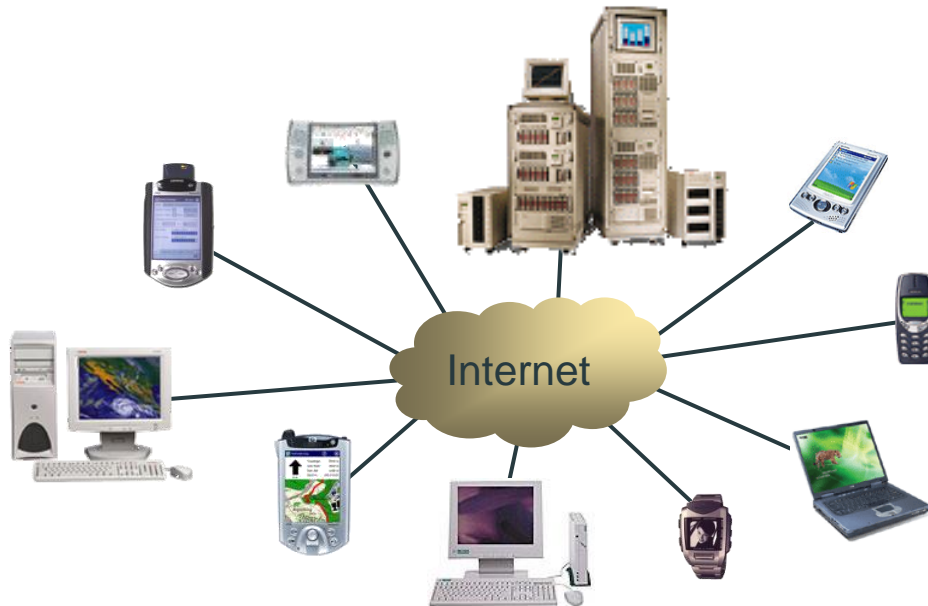
Organização hierárquica multi-nível



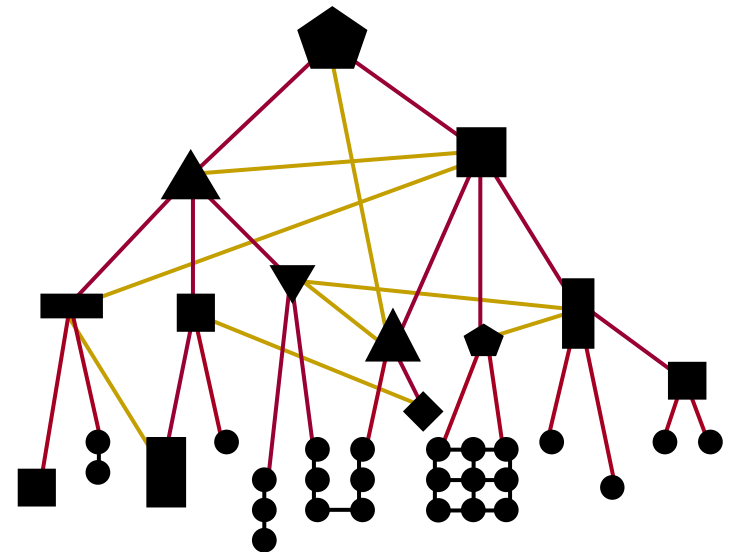
Principais Características:

- Um utilizador um computador
- Computadores menos dispendiosos ao nível departamental
- Interação com utilizador através de interfaces gráficas
- Arquitecturas cliente/servidor

Computação em Rede



Organização híbrida

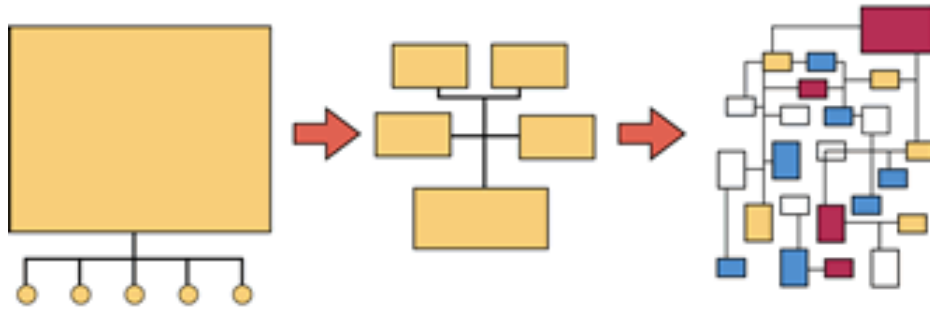


Principais Características:

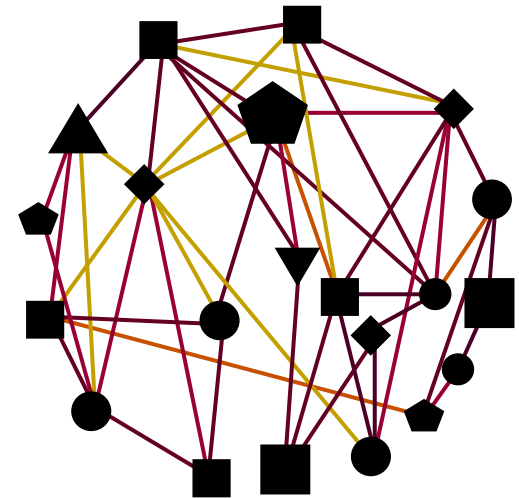
- Um utilizador muitos computadores
- Processamento e dados dispersos por múltiplos dispositivos heterogéneos
- Interacção com o utilizador através interfaces multi-modais
- Arquitecturas multi-camada

Evolução de Paradigmas de Computação

Computação em Rede - Evolução



Organização em rede



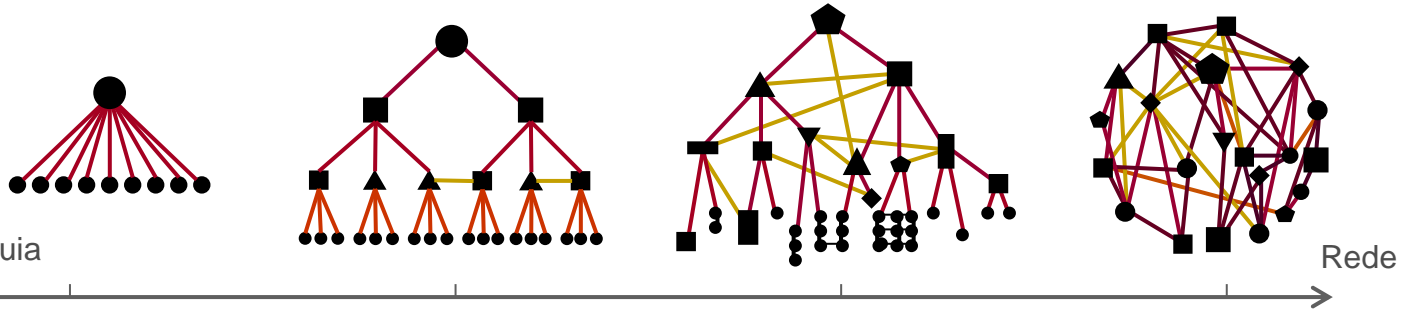
Principais Características:

- Múltiplos elementos heterogêneos organizados em redes de múltiplas dependências
- Estrutura dinâmica que se adapta às condições e necessidades, de modo a tirar partido das múltiplas capacidades heterogêneas disponíveis na rede

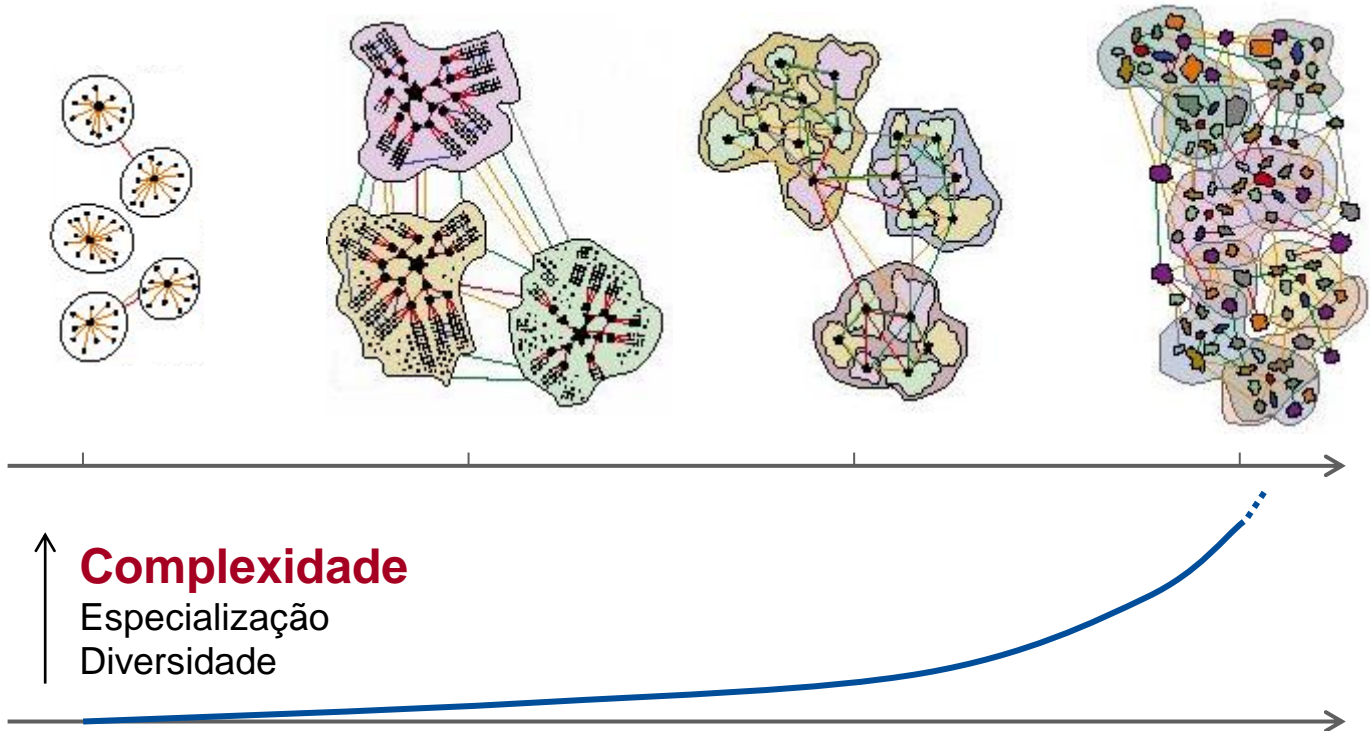
Complexidade Crescente

Tipo de Organização

Hierarquia



Topologia

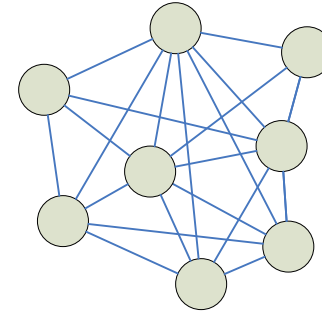
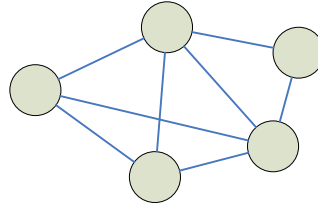
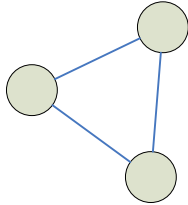


Software e Complexidade

*“Complexity is the business we are in
and complexity is what limits us.”*

Frederick Brooks, *The Mythical Man-Month*, 1995

Complexidade

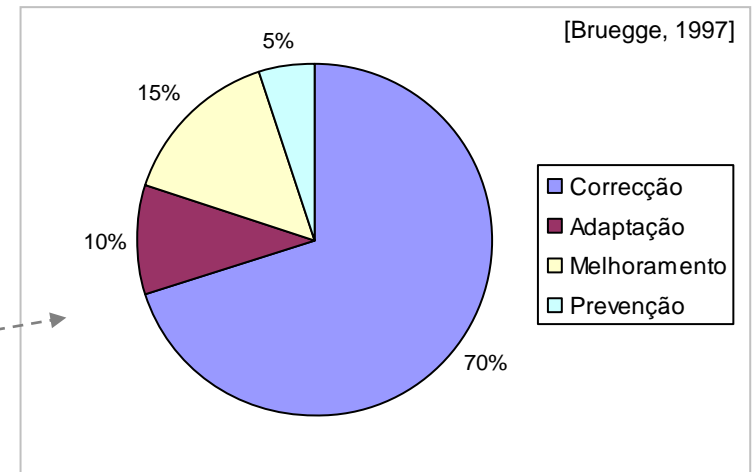
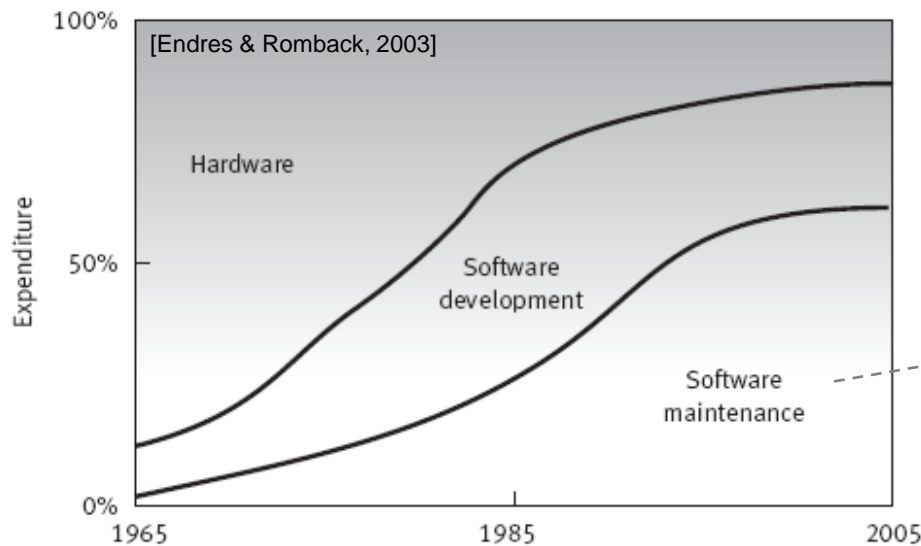


- Um problema de interacção
 - De partes do sistema
 - De elementos de informação
 - De elementos das equipas de desenvolvimento
- Um sistema com duas vezes mais partes é muito mais do que duas vezes mais complexo
 - **Explosão combinatória – Crescimento exponencial**
- Evolução temporal da complexidade

Complexidade Crescente

- Evolução dos sistemas computacionais caracterizada por complexidade crescente!
 - Cada nova geração de computadores permite criar sistemas maiores e mais complexos
 - Aplicações industriais facilmente atingem vários milhões de linhas de código
- Vulnerabilidades críticas:
 - **Robustez:**
 - Crescente número e impacto de falhas
 - **Segurança:**
 - Crescente vulnerabilidade a ataques e intrusões
 - **Manutenção:**
 - Crescentes custos de manutenção
 - **Monitorização e Gestão:**
 - Crescente dificuldade de configuração, monitorização e gestão

Complexidade de Software



- **Problemas latentes**

- Esforço de desenvolvimento (tempo, recursos)
- Gestão e manutenção
- Garantia de qualidade:
 - Crescente vulnerabilidade a ataques e intrusões
 - Crescente número e impacto de falhas

Y16.7 trillion (\$182 billion) Mistake by Deutsche Bank Nearly Sinks Osaka Exchange

[IEEE SPECTRUM, 04-06-2010]



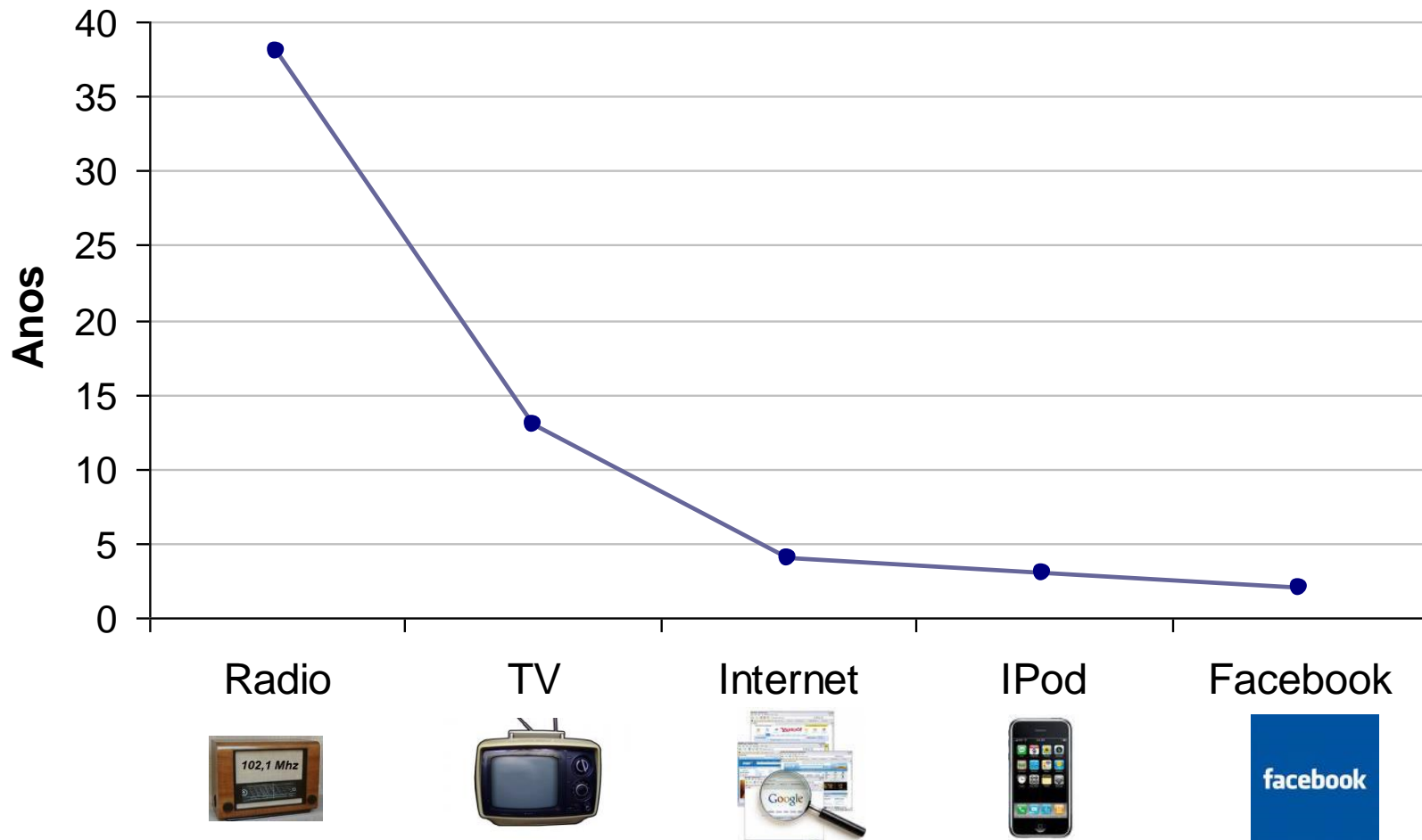
The international stock markets dodged a bullet this week. According to news reports like this one in Reuters, Deutsche Bank's proprietary trading unit in Japan mistakenly placed sell orders of Y16.7 trillion (\$182 billion) at the Osaka Securities Exchange on Tuesday when it first opened.

The reason was a software problem.

The error caused the Tokyo Stock Exchange's Nikkei-225 futures index to lose 110 points or one per cent. If mistake hadn't been caught quickly, it could have caused a repeat of the Dow flash meltdown of a few weeks ago some analysts said.

Mudança Rápida e Contínua

**Tempo de difusão da tecnologia
(50 milhões de utilizadores)**



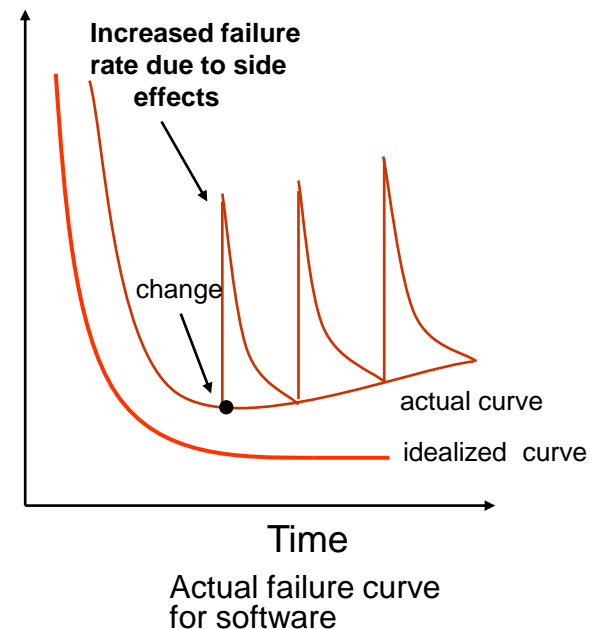
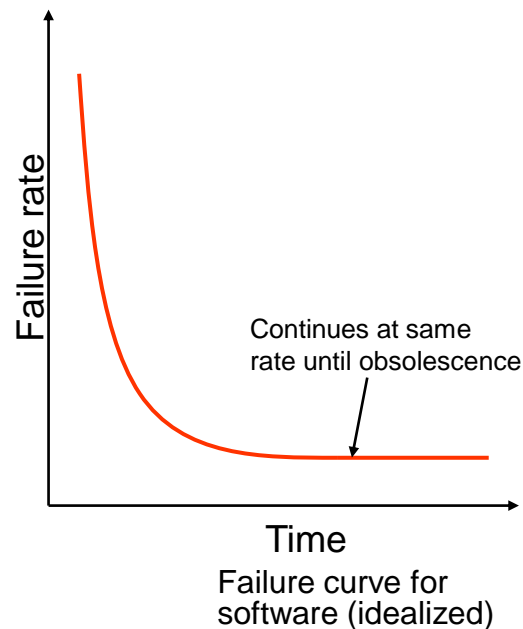
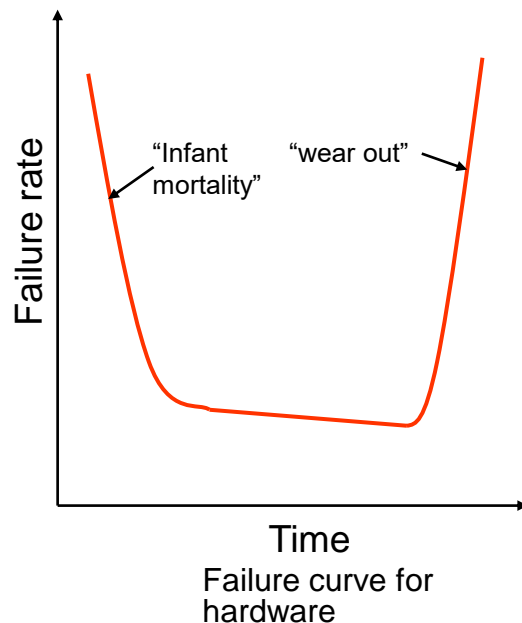
Mudança Rápida e Contínua

- Objectivos, sistemas e pessoas evoluem
- Evolução permanente
- Necessidade de estabilidade para a concretização de sistemas viáveis
- **Necessidade de conciliar mudança com estabilidade**

Características do Software

- O software é desenvolvido e não produzido no sentido clássico
- O processo de desenvolvimento é difícil de gerir
- Os domínios dos problemas a resolver são complexos
- O software oferece uma elevada flexibilidade
- Os sistemas de software possuem dinâmicas internas complexas

Desenvolvimento de hardware vs. software



[Pressman, 2003]

Software: algumas ideias pré-concebidas

Ideia:

- Se um projecto estiver atrasado, o atraso pode ser recuperado pela utilização de mais programadores

Realidade:

- O desenvolvimento de software não é um processo mecânico como a manufactura - ***“adding people to a late software project makes it later”*** [Brooks, 1975]

Ideia:

- Uma descrição geral dos objectivos é suficiente para começar a escrever programas - os detalhes podem ser definidos posteriormente

Realidade:

- Uma especificação inicial deficiente é uma das principais causas de insucesso na realização de software. ***“Quanto mais cedo se começar a programar, mais tarde o programa estará pronto”***

[Pressman, 2003].

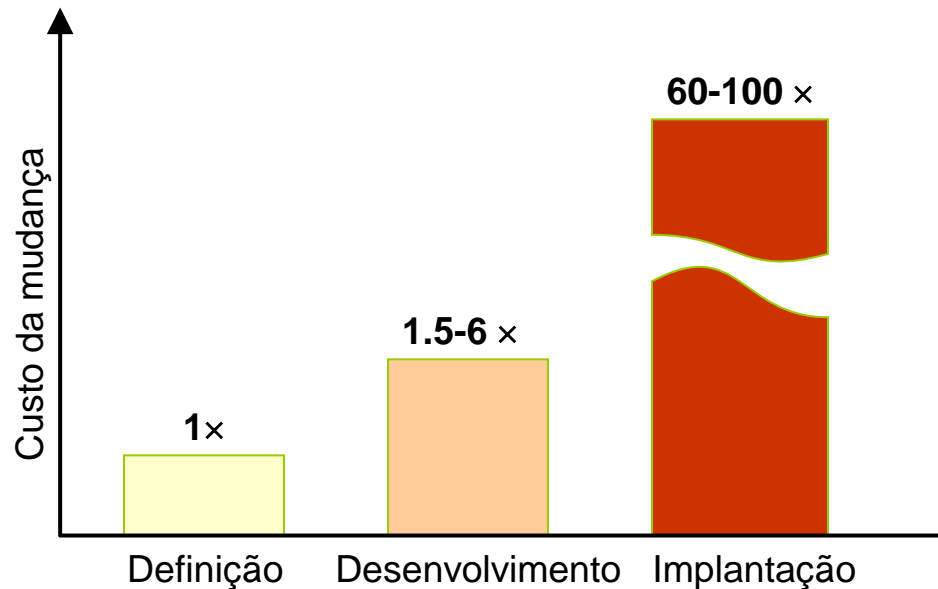
Software: algumas ideias pré-concebidas

Ideia:

- Os requisitos de um programa mudam continuamente, mas essas mudanças podem ser facilmente acomodadas porque o software é flexível

Realidade:

- É verdade que o software é flexível, mas o impacto da mudança varia com a altura em que é introduzida



[Pressman, 2003]

Desenvolvimento de Software

To build software that is ready to meet the challenges of the twenty-first century, you must recognize a few simple realities:

- Software has become deeply embedded in virtually every aspect of our lives. The number of people who have an interest in the features and functions provided by a specific application¹ has grown dramatically. *A concerted effort should be made to understand the problem before a software solution is developed.*
- The information technology requirements demanded by individuals, businesses, and governments grow increasingly complex with each passing year. Large teams of people now create computer programs. Sophisticated software that was once implemented in a predictable, self-contained computing environment is now embedded inside everything from consumer electronics to medical devices to autonomous vehicles. *Design has become a pivotal activity.*
- Individuals, businesses, and governments increasingly rely on software for strategic and tactical decision making as well as day-to-day operations and control. If the software fails, people and major enterprises can experience anything from minor inconvenience to catastrophic consequences. *Software should exhibit high quality.*
- As the perceived value of a specific application grows, the likelihood is that its user base and longevity will also grow. As its user base and time in use increase, demands for adaptation and enhancement will also grow. *Software should be maintainable.*

These simple realities lead to one conclusion: *Software in all its forms and across all its application domains should be engineered.*

[Pressman & Maxim, 2019]

Engenharia de Software

- Surgiu de forma consistente no final da década de 1960, impulsionada pelas falhas crónicas em projectos de software, nomeadamente em termos de prazos e de orçamentos
 - Necessidade de aplicações de grande dimensão e elevada qualidade
 - Aplicações militares
 - O que foi designado como “**A crise do software**”
- O termo surgiu no seguimento da **Conferência da NATO** em Garmisch Partenkirchen (Alemanha), 1968, onde participaram, entre outros, Tony Hoare, Edsger Dijkstra, Alan Perlis e Niklaus Wirth
- **Compreensão de que os princípios da engenharia deveriam ser aplicados ao desenvolvimento de software**

Engenharia de Software

- Desenvolvimento de Software como uma actividade de **engenharia**:
 - Sistemático
 - Quantificável
- **Engenharia de Software**:
 - Aplicação de abordagens sistemáticas, disciplinadas e quantificáveis ao desenvolvimento, operação e manutenção de software [IEEE, 1990]

Engenharia de Software

- Conjunto de metodologias, técnicas e ferramentas para a realização de
 - sistemas informáticos de qualidade
 - de acordo com orçamento definido
 - no prazo estabelecido
- Mesmo perante a complexidade e a mudança

Bibliografia

[Brooks, 1975]

Frederick Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, 1975.

[Bruegge, 1997]

Bernd Bruegge, *Lecture Notes on Software Engineering*, Carnegie Mellon University - School of Computer Science, 1997.

[Endres & Rombach, 2003]

Albert Endres, Dieter Rombach, *A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories*, Addison Wesley, 2003.

[IEEE, 1990]

IEEE 610.12-1990, *Standard Glossary of Software Engineering Terminology*, 1990.

[Leveson & Turner, 1993]

Nancy Leveson, Clark S. Turner, *An Investigation of the Therac-25 Accidents*, IEEE Computer, Vol. 26, No. 7, July 1993, pp. 18-41.

[Pressman, 2003]

Roger Pressman, *Software Engineering: a Practitioner's Approach*, McGraw-Hill, 2003.

[Pressman & Maxim, 2019]

Roger Pressman, Bruce Maxim, *Software Engineering: a Practitioner's Approach*, McGraw-Hill, 2019.

[Schach, 2011]

Stephen Schach, *Object-Oriented and Classical Software Engineering*, McGraw-Hill, 2011.

[Roubine 2000]

O. Roubine - *E-Development*. Rational, 2000.