



Engenharia de Software

Conceitos Básicos

Ritomar Torquato
Instituto Federal do Piauí

Introdução

Engenharia de Software: Por onde começar?



Introdução

O que é Software?



A resposta não é óbvia, segundo Pressman, em 1970 menos de 1% dos profissionais poderiam definir o que é Software.

Introdução

Software

substantivo masculino

1. conjunto de componentes lógicos de um computador ou sistema de processamento de dados; programa, rotina ou conjunto de instruções que controlam o funcionamento de um computador; suporte lógico.
2. todo programa armazenado em discos ou circuitos integrados de computador, esp. destinado a uso com equipamento audiovisual.

Introdução

O que é Engenharia?



Introdução

Engenharia

substantivo feminino

1. aplicação de métodos científicos ou empíricos à utilização dos recursos da natureza em benefício do ser humano.
2. *eng* formação, ciência e ofício de engenheiro. "e. civil, naval"
3. *p.ext.* conjunto de atividades e funções de um engenheiro, que vão da concepção e do planejamento até a responsabilidade pela construção e pelo controle dos equipamentos de uma instalação técnica ou industrial.
4. *eng* projeto e manufatura de produtos complexos. "e. de software"

Introdução

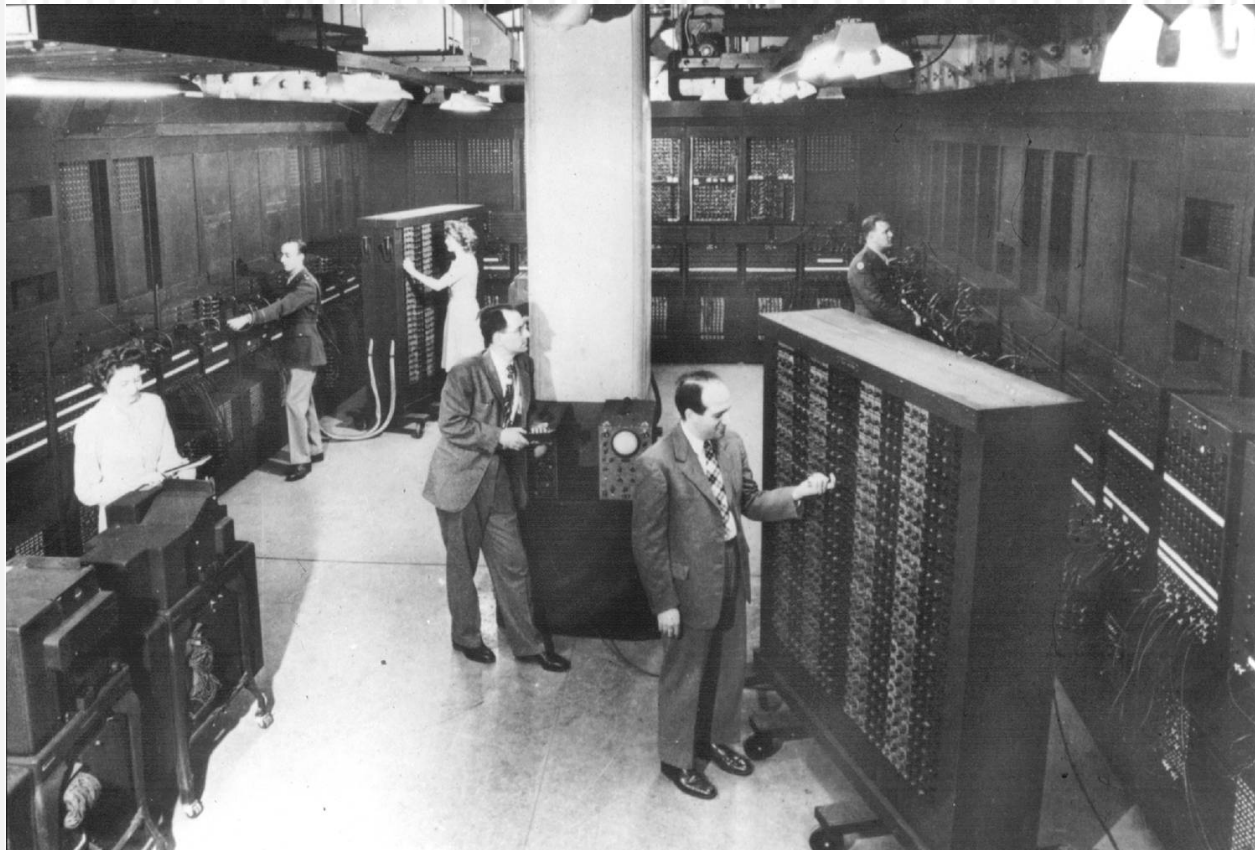
O que é Engenharia de Software?

Engenharia de Software é uma área da computação voltada à especificação, desenvolvimento e manutenção de sistemas de **software**, com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade.



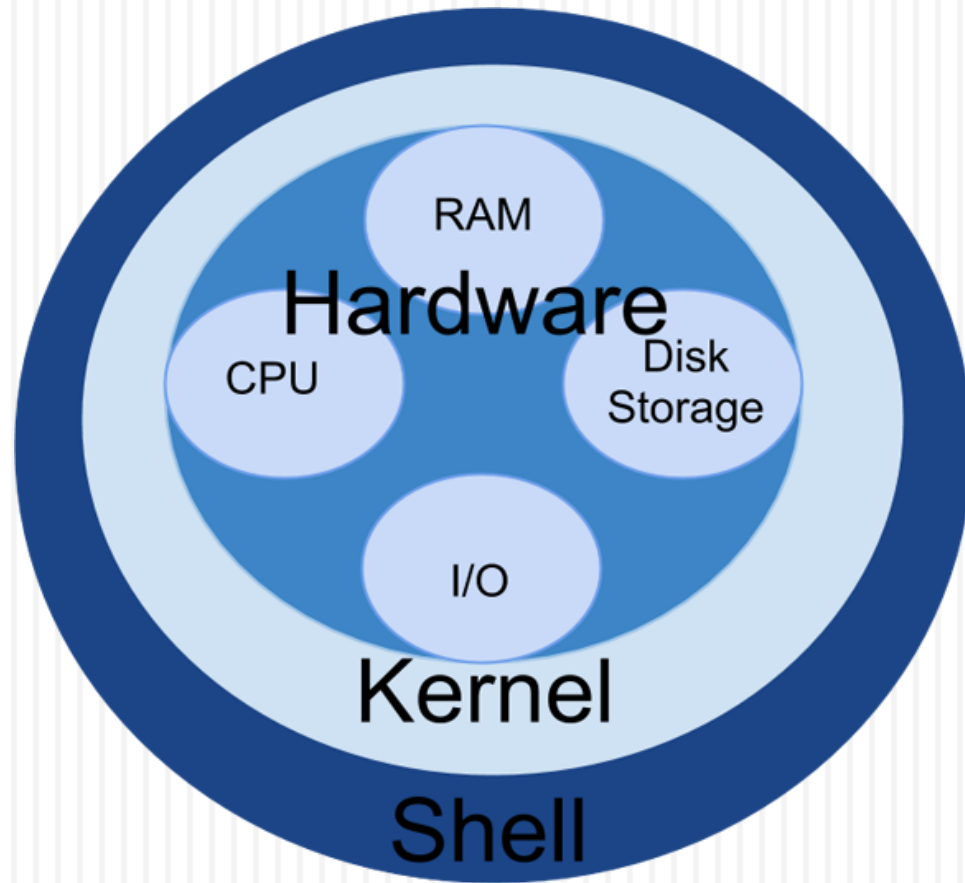
Engenharia de software – Wikipédia, a enciclopédia livre
https://pt.wikipedia.org/wiki/Engenharia_de_software

Introdução



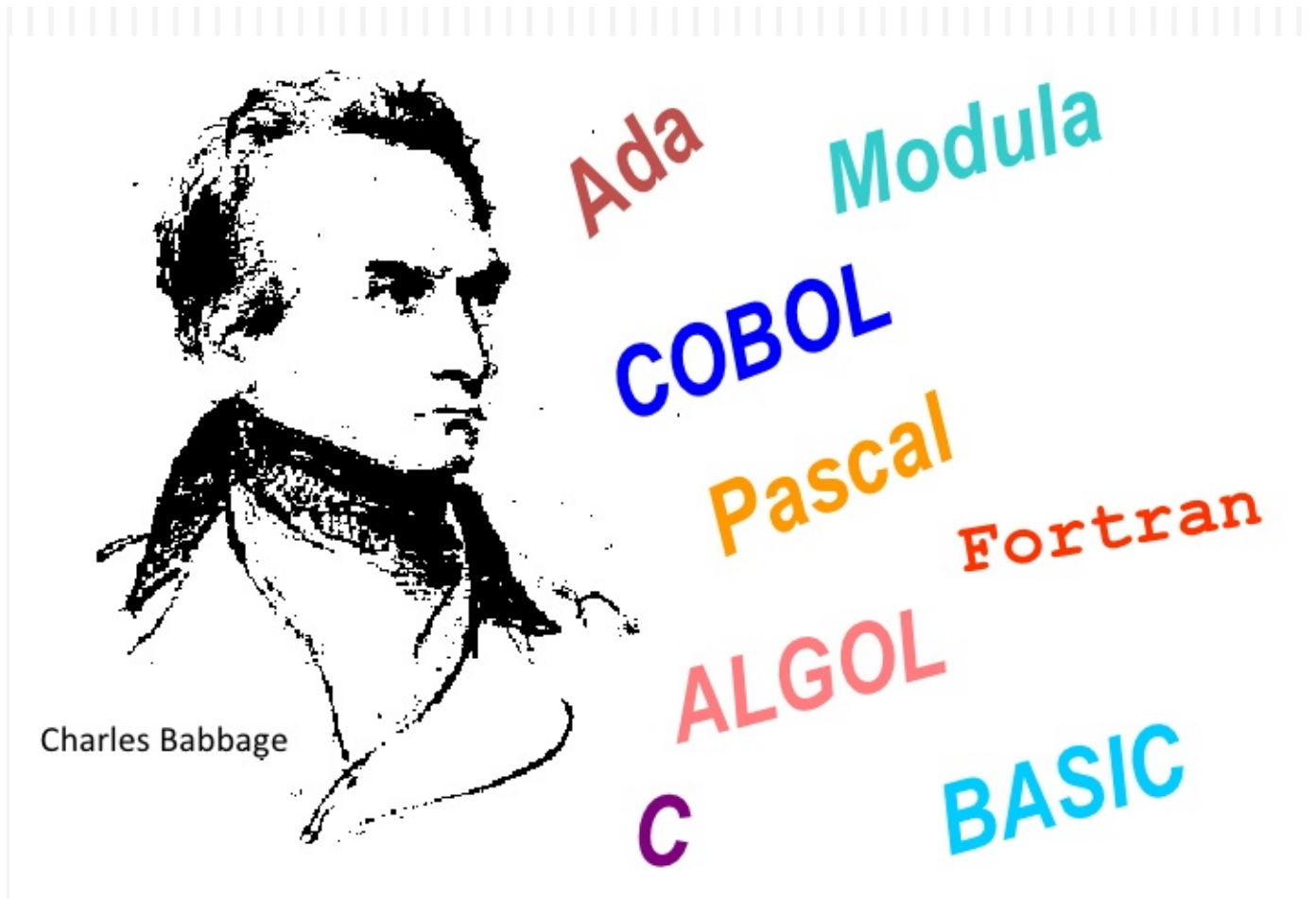
Nos anos 40, quando se iniciou a evolução dos sistemas computadorizados, grande parte dos esforços, e custos, era concentrada no desenvolvimento do hardware.

Introdução



No início dos anos 50, os esforços se voltaram para o desenvolvimento dos sistemas operacionais...

Introdução



...Linguagens de Programação e Compiladores.

Introdução

```
START: JUMP    LOOP      # jump past sensor and constant locations
RSV:   0000          # read right sensor value here
LSV:   0000          # read left sensor value here
RMP:   0000          # write right motor power level here
LMP:   0000          # write left motor power level here
OFF:   0000          # store motor-off constant here
ON:    0100          # store motor-on constant here
LOOP:  LOAD     1      RSV # load right sensor value into register 1
      LOAD     2      LSV # load left sensor value into register 2
      SUB      1  2  3    # subtract 1 from 2 and store result in 3
      LOAD     1      OFF # load motor-off constant into register 1
      LOAD     2      ON  # load motor-on constant into register 2
      BRANCH   3      RGT # if the left sensor is greater than the
LFT:   STORE    2      RMP # right then turn the right motor on
      STORE    1      LMP # and turn the left motor off
      JUMP     LOOP    # and then jump to beginning of the loop
RGT:   STORE    2      LMP # else turn the left motor on
      STORE    1      RMP # and turn the right motor off
      JUMP     LOOP    # and then jump to beginning of the loop
```

Época de poupar o usuário de conhecer questões relacionadas ao hardware, e concentrar na resolução dos problemas.

Introdução



No início dos anos 60, o uso dos sistemas computacionais tiveram um considerável crescimento.

Introdução

Preço do Hardware

Produtividade e
Qualidade do
Software



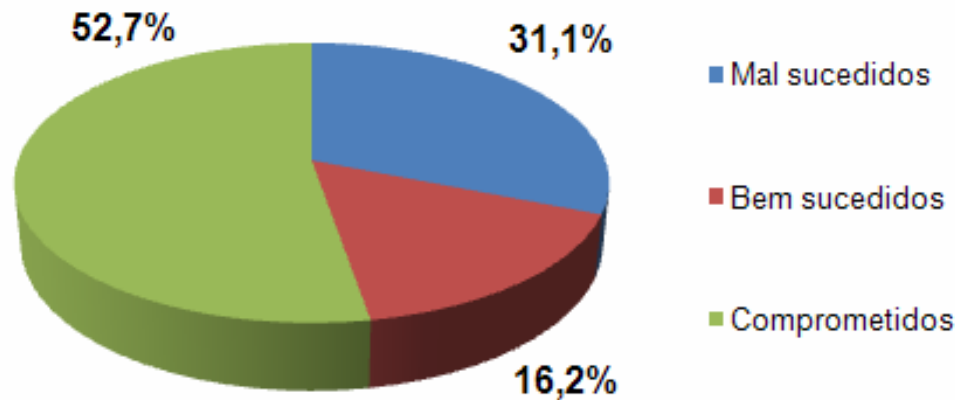
Demanda por
Hardware

Demanda por
Software

Crise do Software

Introdução

Resultado dos Projetos



A **crise do software** foi um termo utilizado nos anos 1970, 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.

Fonte: https://pt.wikipedia.org/wiki/Crise_do_software

Introdução



Apesar da queda dos preços dos equipamentos, o custo de desenvolvimento de software não obedece a esta mesma tendência.

Introdução



O desenvolvimento de software implica grande carga de trabalho, grande número de pessoas num prazo longo de desenvolvimento.

Introdução

□ Sobre a Engenharia de Software:



Faz o Quê - Engenharia de Software - Bloco I



Faz o Quê - Engenharia de Software - Bloco II



Faz o Quê - Engenharia de Software - Bloco III

Introdução



Objetivo: Propor soluções para o desenvolvimento de software, pela transferência de conceitos, técnicas, metodologias e ferramentas de desenvolvimento de software

Principais Aspectos do Software



O software é produto “vendido”, destinado ao uso por pessoas “comuns”.

Principais Aspectos do Software



Os usuários requerem interface e documentação que permitam uso eficiente. O software deve ser testado: os usuários não estarão interessados em detectar e corrigir erros.

Principais Aspectos do Software



- Não é manufaturado no sentido clássico;
- Não se desgasta;
- A maioria é concebido sob medida.

A experiência do programador é adquirida através de tentativa e erro.

Principais Aspectos do Software

A nível industrial, algumas preocupações com o desenvolvimento de software:



- Por que o software demora tanto para ser concluído?
- Por que os custos de produção têm sido tão elevados?
- Por que não é possível detectar todos os erros antes que o software seja entregue ao cliente?
- Por que é tão difícil medir o progresso durante o processo de desenvolvimento de software?

Principais Aspectos do Software

Problemas geradores:



- Falta tempo para coletar dados.
- Projetos baseados em informações vagas sobre as necessidades do cliente.
- Falta de atenção ao teste do software (conceito relativamente recente).
- Falta de atenção à manutenibilidade dos sistemas.

Principais Aspectos do Software



Mito 01 (de gerenciamento): Se a equipe dispõe de um manual repleto de padrões e procedimentos de desenvolvimento de software, então a equipe está apta a encaminhar bem o desenvolvimento.

É preciso que a equipe aplique efetivamente os conhecimentos do manual. É necessário que o manual reflita a moderna prática de desenvolvimento de software e que este seja exaustivo com relação aos problemas de desenvolvimento que poderão aparecer no percurso.



Principais Aspectos do Software



Mito 02 (de gerenciamento): A equipe tem computadores de última geração, então ela tem as melhores ferramentas para o desenvolvimento de software.

Mais importante do que ter um hardware de última geração é ter ferramentas para a automatização e auxílio do desenvolvimento de software (as ferramentas CASE).



Principais Aspectos do Software



Mito 03 (de gerenciamento): Se o desenvolvimento do software estiver atrasado, basta aumentar a equipe para cumprir o prazo estabelecido.

A introdução de novos profissionais numa equipe em andamento vai requerer treinamento dos novos membros pelas pessoas que já estão envolvidos e, conseqüentemente, causar maiores atrasos no cronograma.



Principais Aspectos do Software



MITO

Mito 04 (do cliente): Uma descrição breve e geral dos requisitos do software é o suficiente para iniciar o seu projeto, maiores detalhes podem ser definidos posteriormente.

O cliente deve definir o mais precisamente os requisitos do software: funções, desempenho, interfaces, restrições de projeto e critérios de validação, etc.



VERDADE

Principais Aspectos do Software

MITO

Mito 05 (do cliente): Os requisitos mudam continuamente durante o seu desenvolvimento, mas isto não representa um problema, uma vez que o software é flexível e poderá suportar facilmente as alterações.

O software é flexível (mais que dos produtos manufaturados). O fator de multiplicação nos custos de desenvolvimento do software devido a alterações nos requisitos cresce em função do estágio de evolução do projeto. Veja a figura 1.1.

VERDADE

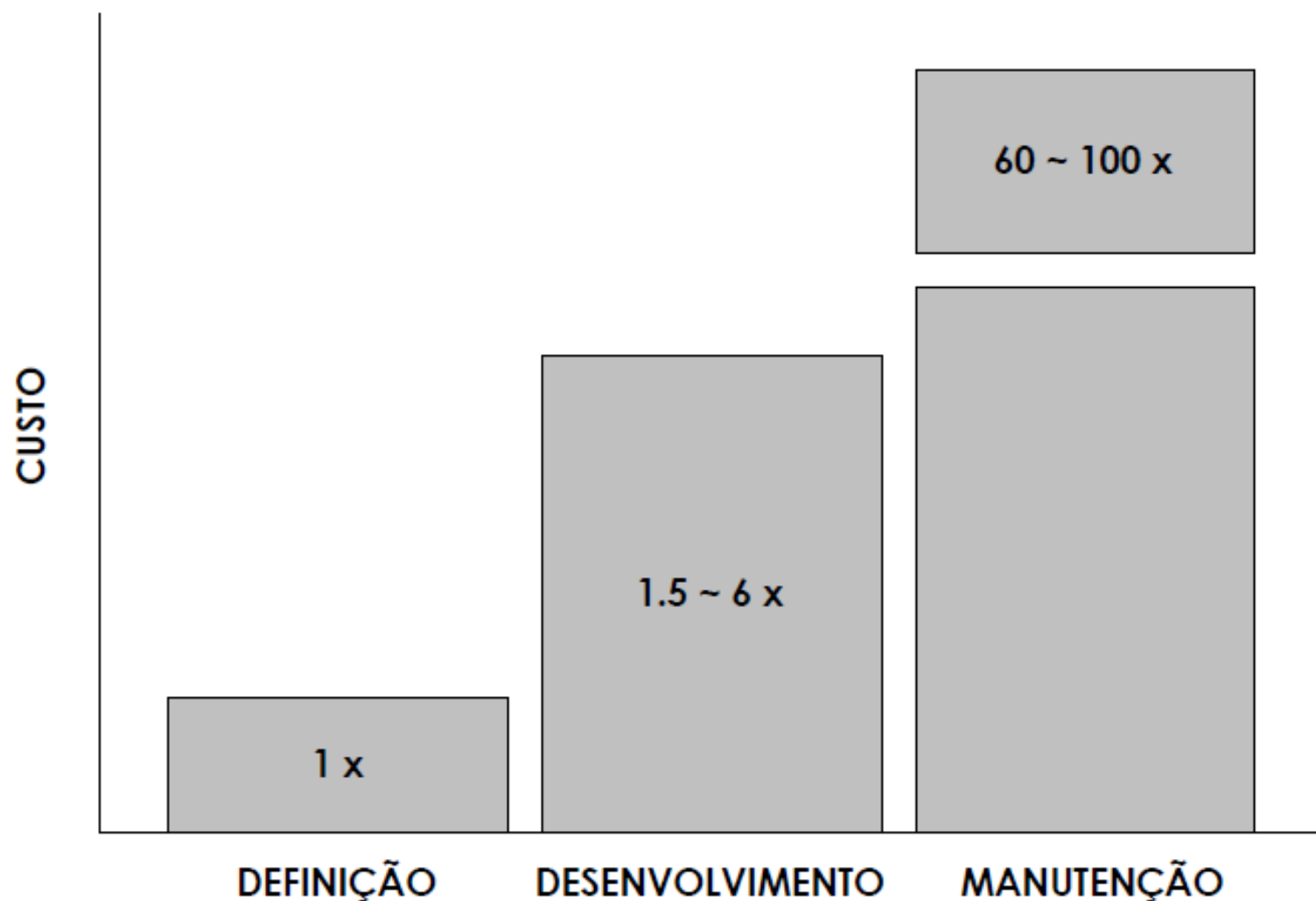


Figura 1.1 - Influência das alterações de requisitos no custo de um sistema.

Principais Aspectos do Software



Mito 06 (do profissional): Após a edição do programa e a sua colocação em funcionamento, o trabalho está terminado.

O que ocorre na realidade é completamente diferente disto. Segundo dados obtidos a partir de experiências anteriores, 50 a 70% do esforço de desenvolvimento de um software é despendido após a sua entrega ao cliente (manutenção).



Principais Aspectos do Software



Mito 07 (do profissional): Enquanto o programa não entrar em funcionamento, é impossível avaliar a sua qualidade.

Na realidade, a preocupação com a garantia do software deve fazer parte de todas as etapas do desenvolvimento, sendo que, ao fim de cada uma destas etapas, os documentos de projeto devem ser revisados observando critérios de qualidade.



Principais Aspectos do Software



Mito 08 (do profissional): O produto a ser entregue no final do projeto é o programa funcionando.

O programa em funcionamento é uma das componentes do software... além do software, um bom projeto deve ser caracterizado pela produção de um conjunto importante de documentos, os quais podem ser identificados com auxílio da figura 1.2.



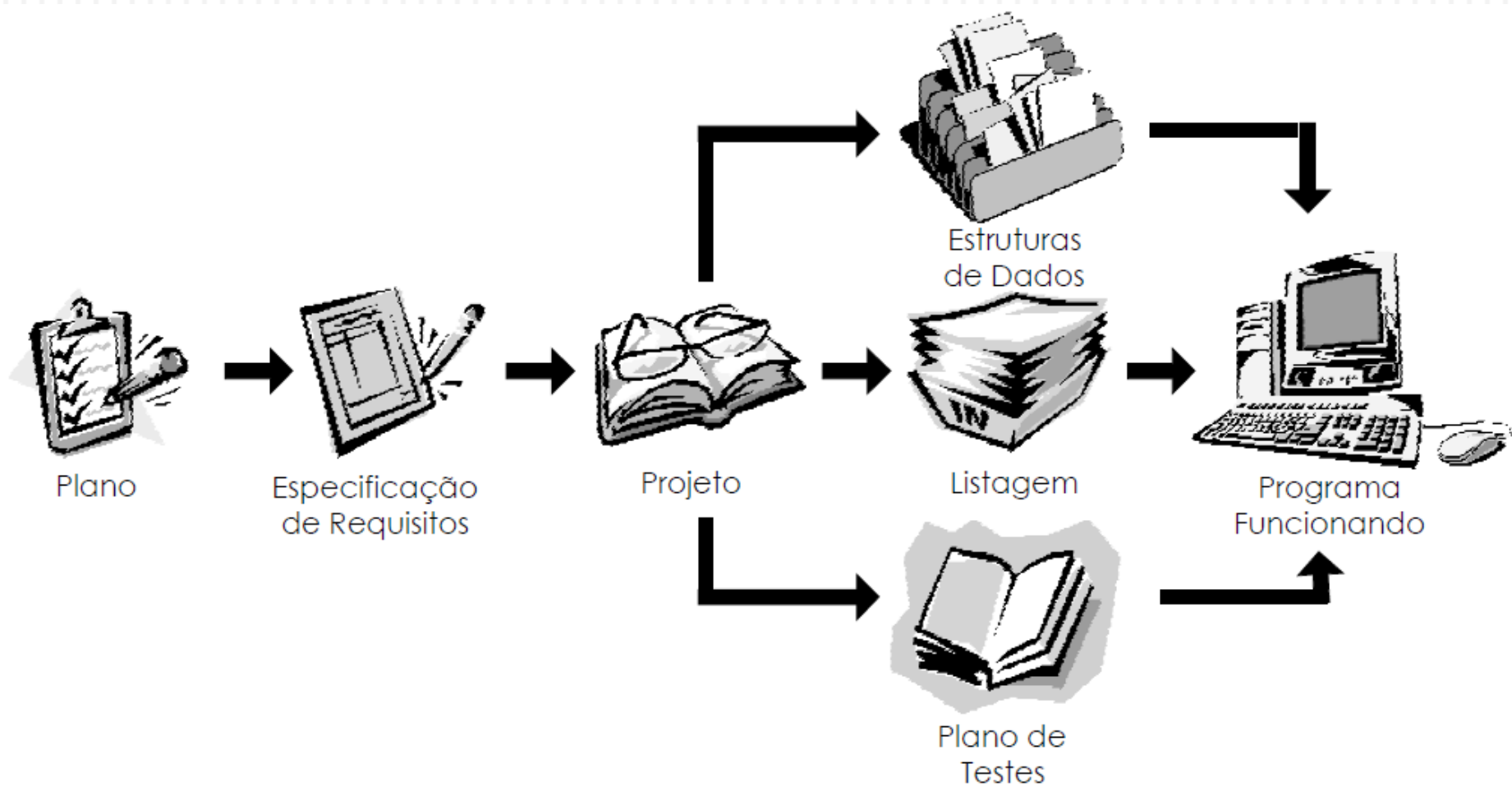
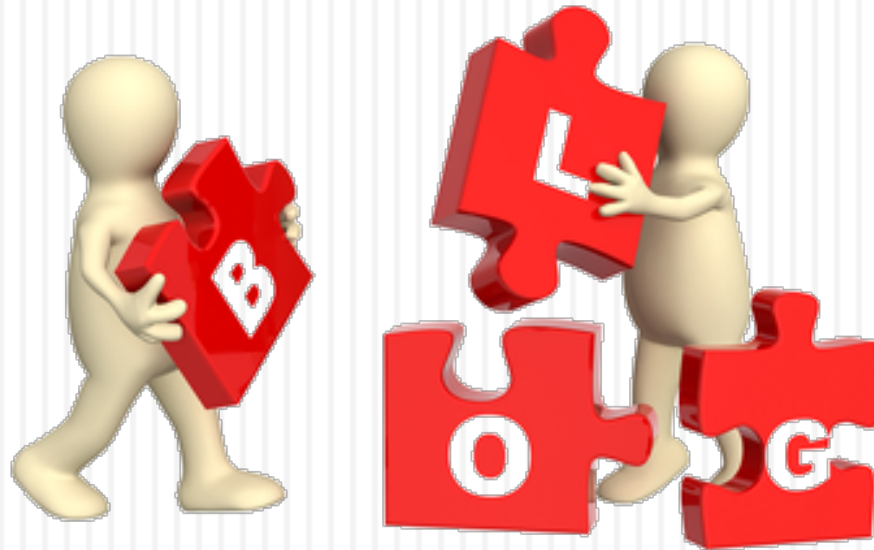


Figura 1.2 - Componentes do software.

Introdução

- Problemas processo de desenvolvimento de software:



Empresa FazSite - Problemas processo de desenvolvimento de software