

Manutenção de Software



Estatísticas sobre manutenção

90% do custo total [Erlikh 2000]

60% do esforço total [Pressman 1997]

70 bilhões de dólares anuais
nos EUA [Edelstein 1993]

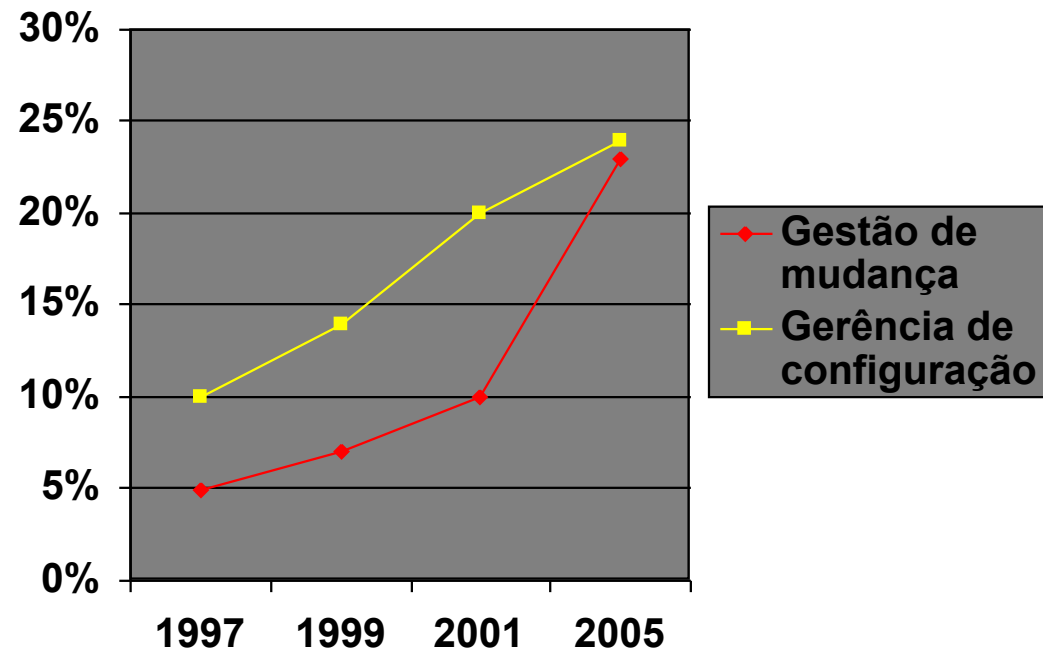
250 bilhões de linhas de código em
manutenção [Somerville 2000]

Cenário mundial

- Somente 21% do esforço de manutenção é despendido em correções de defeitos (Bennett and Rajlich 2000)
- Manutenção é necessária mesmo em sistema com qualidade
 - Adaptação para novas regulamentações legais
 - Novos requisitos
 - Torná-lo mais simples de compreender e manter

Cenário brasileiro

- Manutenção no Brasil [MCT 2006]
 - 2500 empresas entrevistadas
 - 95% de confiabilidade
 - Erro máximo de 5,5%

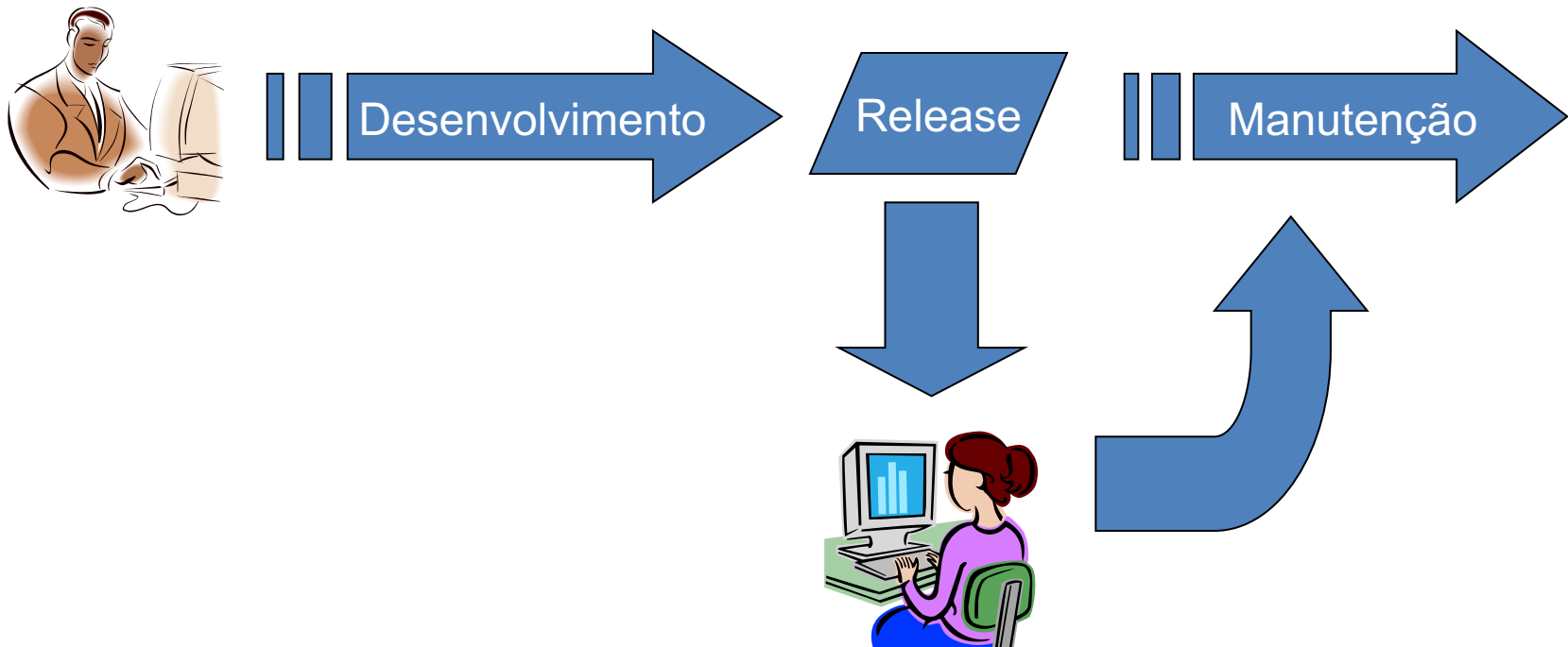


O que é a manutenção?

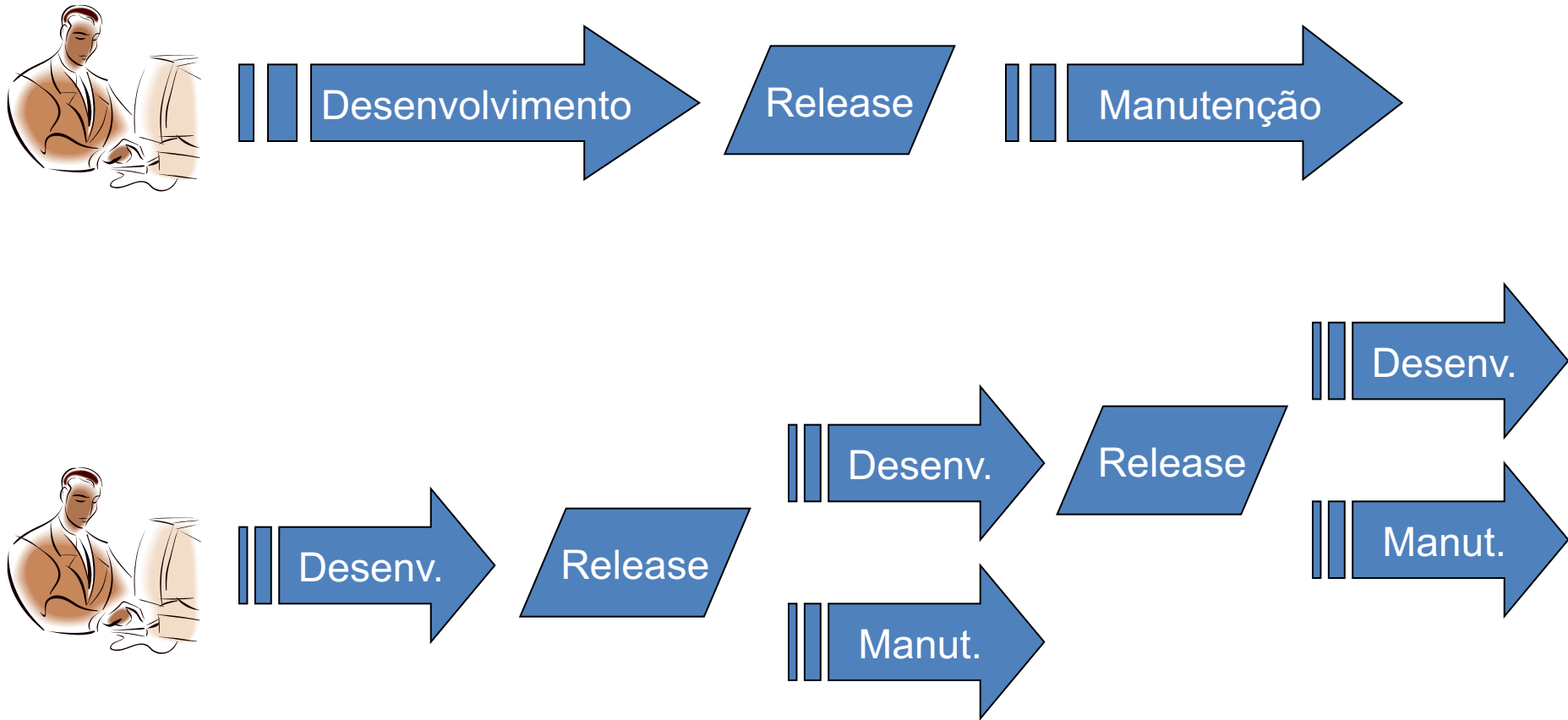
O processo de **modificar um sistema de software** ou componente, **depois da entrega**, para **corrigir falhas, melhorar desempenho ou outros atributos**, ou **adaptar a mudanças no ambiente**.

IEEE Std 620.12 1990

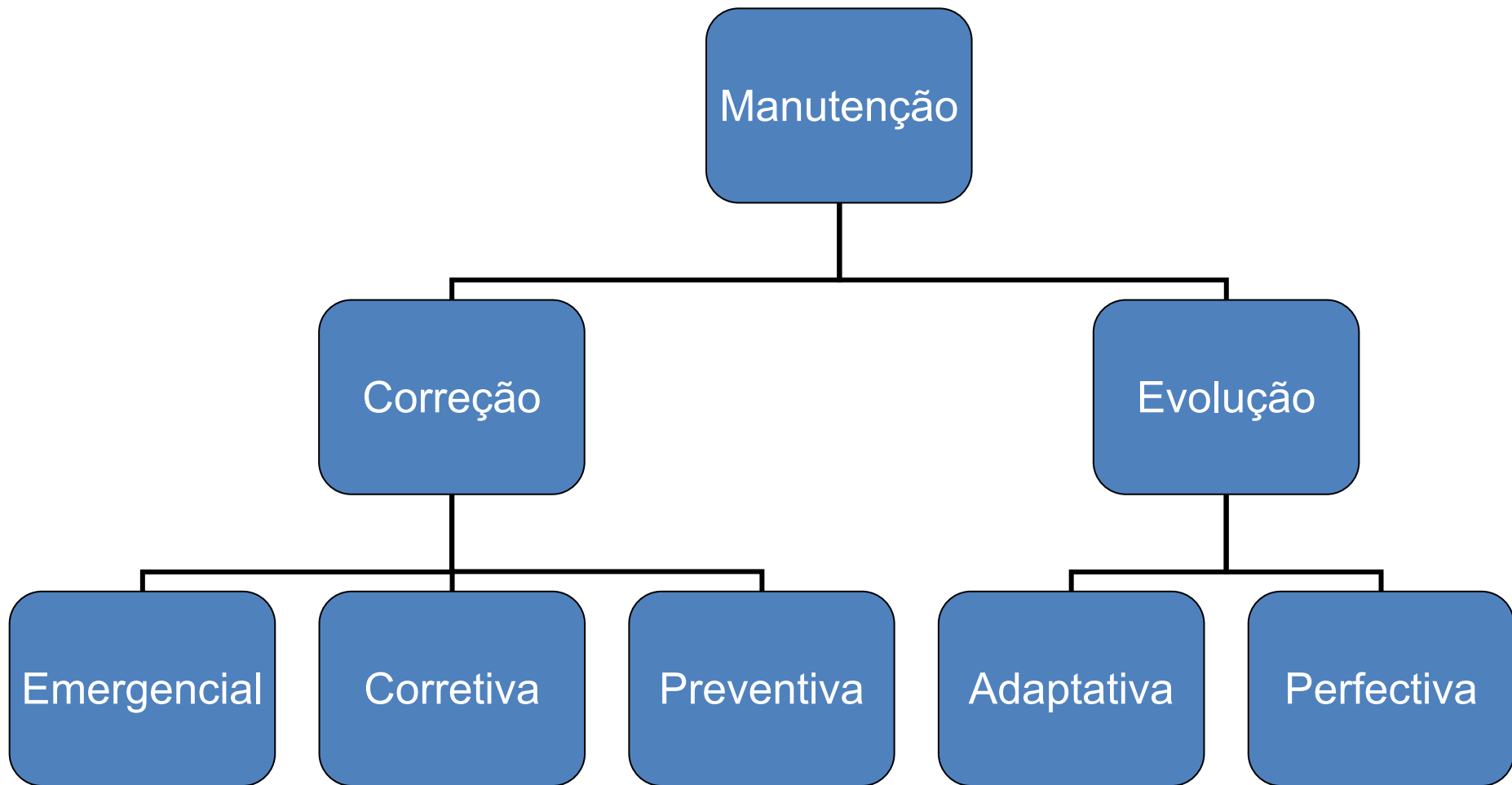
Quando inicia a manutenção?



Quando inicia a manutenção?



Quais são os tipos de manutenção?



Quais são os tipos de manutenção?

- Manutenção emergencial
 - Não programada
 - Mantém temporariamente o sistema funcionando
 - Necessita uma manutenção corretiva posterior
- Manutenção corretiva
 - Reativa
 - Corrige problemas reportados
 - Faz o software voltar a atender aos requisitos

Quais são os tipos de manutenção?

- Manutenção preventiva
 - Pró-ativa
 - Corrige problemas latentes
- Manutenção adaptativa
 - Mantém o software usável após mudanças no ambiente
- Manutenção perfectiva
 - Provê melhorias (funcional e não funcional) para o usuário

Refatoração

- “É uma **mudança na estrutura interna** do software visando torná-lo **mais fácil de entender ou modificar** sem alterar o seu **comportamento externo**” Martin Fowler

Dívida Técnica

Gambiarra

Código duplicado

Alto acoplamento

Baixa coesão

Refatoração

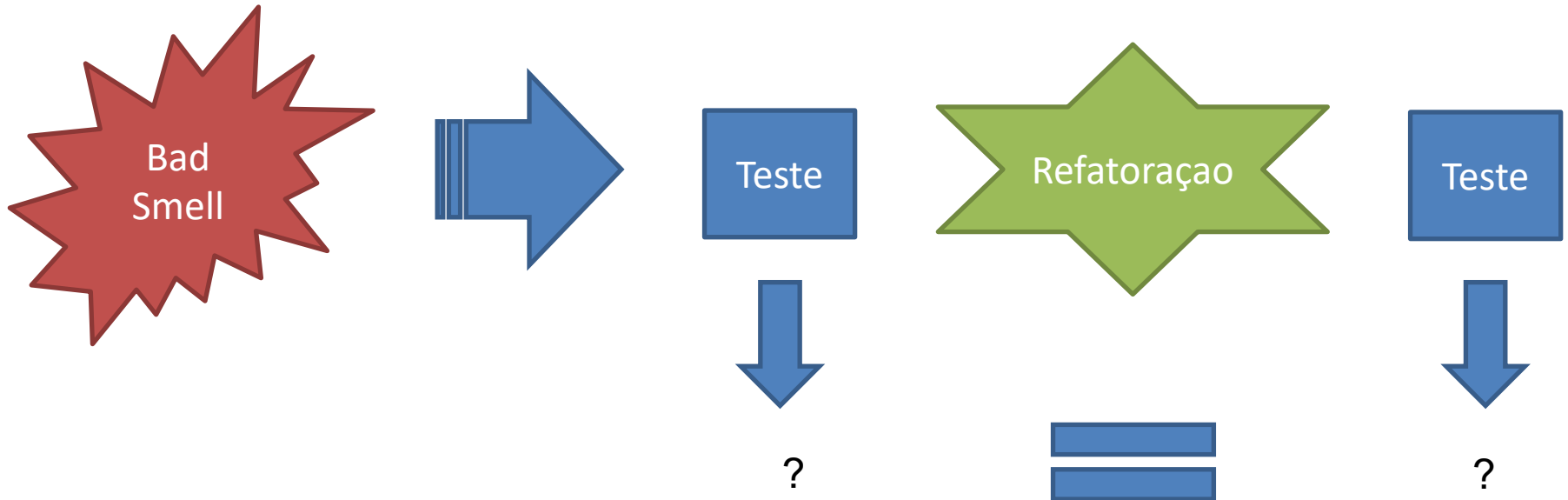
Modularizar

Encapsular

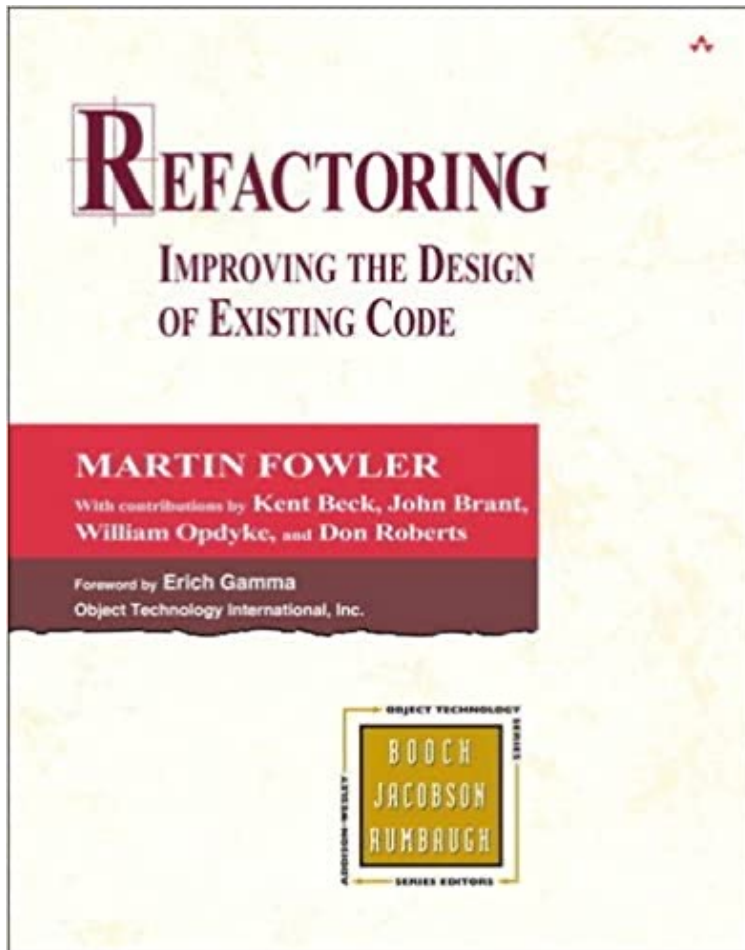
Unificar

Mover

Quando e como refatorar?



Catálogo com 70+ refatorações*



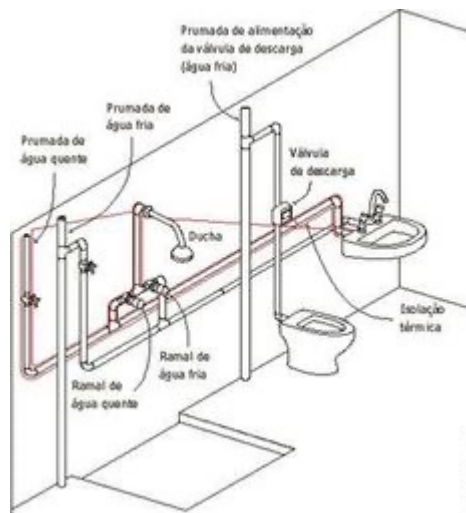
<https://refactoring.com/catalog/>

- Rename Method/Field
- Remove Dead Code
- Encapsulate Field
- Move Method/Field
- Extract Method
- Pull Up Method/Field
- Push Down Method/Field
- Replace Conditional With Polymorphism
- ...

* Disponíveis nas principais IDEs

Documentação

- Fazer documentação e mantê-la atualizada é trabalhoso
- Contudo, documentação atualizada é útil no momento da manutenção



Engenharia reversa

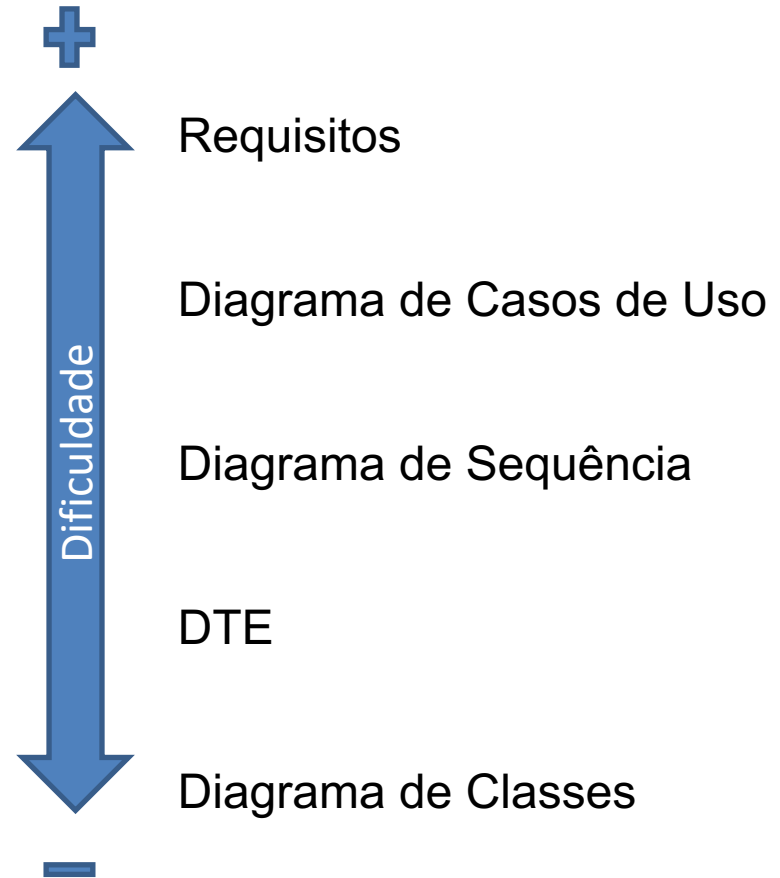
- Sistemas legados usualmente **não têm documentação** atualizada
- A engenharia reversa visa **gerar a documentação** do software a partir do código
- A maioria das **ferramentas CASE** profissionais têm esse recurso, em diferentes níveis de capacidade...

Engenharia Reversa

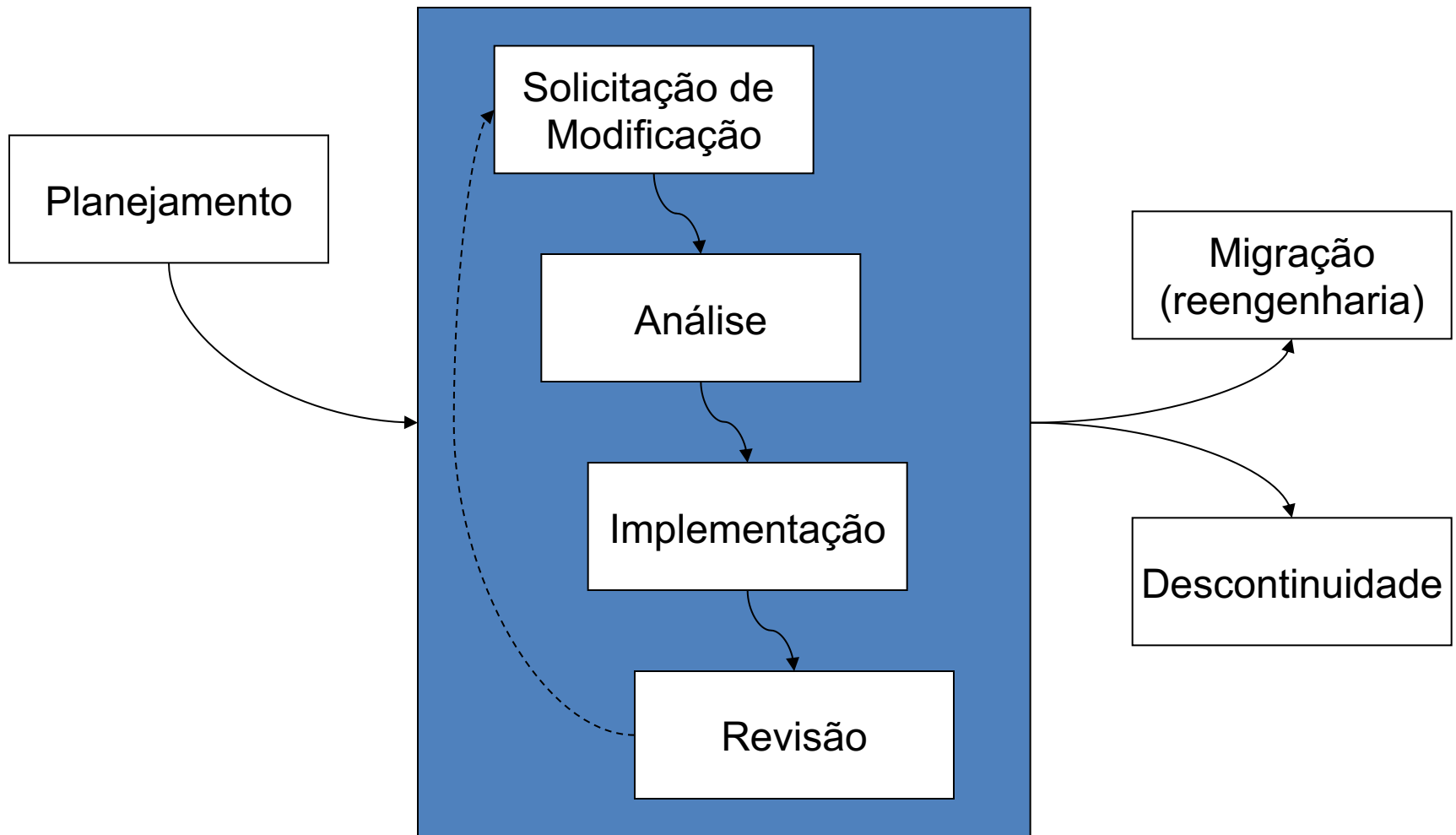
```

17  string sinput;
18  int iLength, iN;
19  double dblTemp;
20  bool again = true;
21
22  while (again) {
23      iN = -1;
24      again = false;
25      getline(cin, sinput);
26      system("cls");
27      stringstream(input) >> dblTemp;
28      iLength = sinput.length();
29      if (iLength < 4) {
30          again = true;
31          continue;
32      } else if (sinput[iLength - 3] != ".") {
33          again = true;
34          continue;
35      } while (++iN < iLength) {
36          if (!isdigit(sinput[iN])) {
37              again = true;
38              continue;
39          }
40      }

```

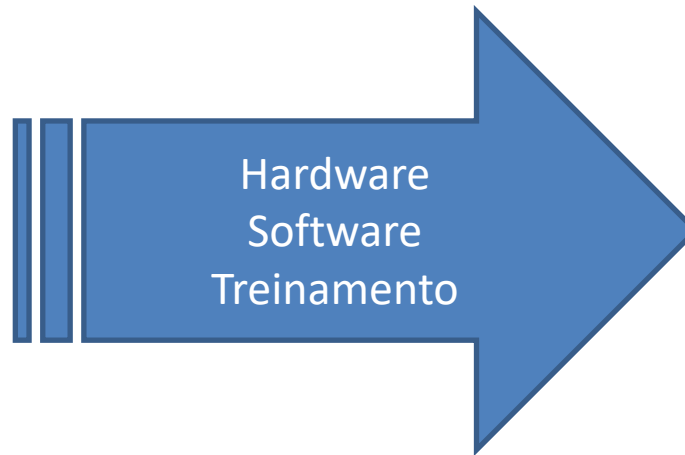


Processo de manutenção



Obsolescência tecnológica

Mainframe
COBOL
NATURAL
DB2
ADABAS
...



PC (Cluster)
Java
Python
Oracle
MySQL
PostgreSQL
...

tempo

- Alto custo para manter a tecnologia antiga
- Dificuldade de encontrar profissionais capacitados
- Dificuldade para adaptar às demandas atuais (móvel, web, ...)

Reengenharia

- Refazer o sistema em uma nova tecnologia
- Possível processo:
 1. Engenharia reversa (caso não tenha documentação disponível)
 2. Engenharia adiante na nova tecnologia
 3. Migração dos dados
 4. Homologação (sistema antigo e novo em paralelo)
- Existem sistemas que automatizam os passos 1 e 2 acima

Principais Referências Bibliográficas

- Bennett, K. H. and V. T. Rajlich (2000). Software maintenance and evolution: a roadmap. International Conference on Software Engineering, The Future of Software Engineering, Limerick, Ireland, ACM Press.
- Edelstein, D. (1993). "Report on the IEEE 1219-1993- Standard for Software Maintenance." ACM SIGSOFT Software Engineering Notes 18(4): 94-95.
- Erlikh, L. (2000). "Leveraging Legacy System Dollars for E-Business." ITProfessional 2(3): 17-23.
- IEEE Std 14764-2006, "Software Engineering – Software Life Cycle Processes – Maintenance".
- MCT (2006). Qualidade e Produtividade no Setor de Software Brasileiro. Brasília, DF, Ministério de Ciência e Tecnologia, Secretaria de Política de Informática.
- Pressman, R. S. (1997). Software Engineering: A Practitioner's Approach, McGraw-Hill.
- Sommerville, I. (2000). Software Engineering, Addison-Wesley.

Manutenção de Software

