



Reengenharia de software

Elton Morais

OBJETIVOS

1

Compreender por que a reengenharia é, algumas vezes, uma opção eficaz, em termos de custos, para evolução do sistema de software;

2

Compreender as atividades, como engenharia reversa e reestruturação de programa, na reengenharia de software;

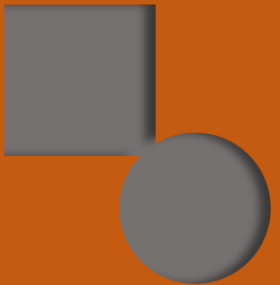
3

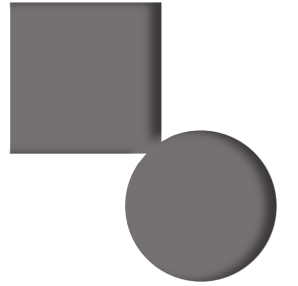
Compreender as diferenças entre reengenharia de software e de dados e por que a reengenharia é dispendiosa e demorada.

INTRODUÇÃO

Muitas empresas dependem de sistemas legados e, por consequência, precisam mantê-los em operação.

Entre as estratégias de evolução de software estão a manutenção, a substituição, a evolução de arquitetura e a **reengenharia de software**.





REENGENHARIA DE SOFTWARE

Reorganização e modificação do software com objetivo de torna-lo mais fácil de manter.

REENGENHARIA DE SOFTWARE

A reengenharia pode envolver **redocumentar**, **organizar** e **reestruturar** o sistema, traduzir o sistema para uma **linguagem de programação mais moderna** e modificar e atualizar a estrutura e os valores dos dados do sistema.

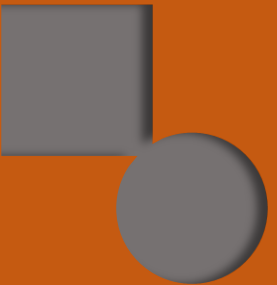
A funcionalidade do software não é modificada e, normalmente, a arquitetura do sistema também permanece a mesma.



REENGENHARIA DE SOFTWARE

A reengenharia é uma forma de obter **reuso** de software e entender os conceitos do domínio da aplicação sendo modificada.

Facilita a compreensão de sistemas cuja **documentação** de análise e projeto normalmente é **inexistente** ou **desatualizada**.



VANTAGENS DA REENGENHARIA DE SOFTWARE



Riscos Reduzidos

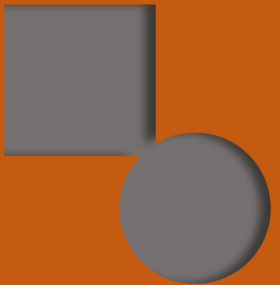


Custos Reduzidos

REENGENHARIA DE SOFTWARE

A reengenharia também está associada ao processo de negócios;

A principal diferença entre a reengenharia e o novo desenvolvimento é o ponto de partida para o desenvolvimento.





Engenharia direta

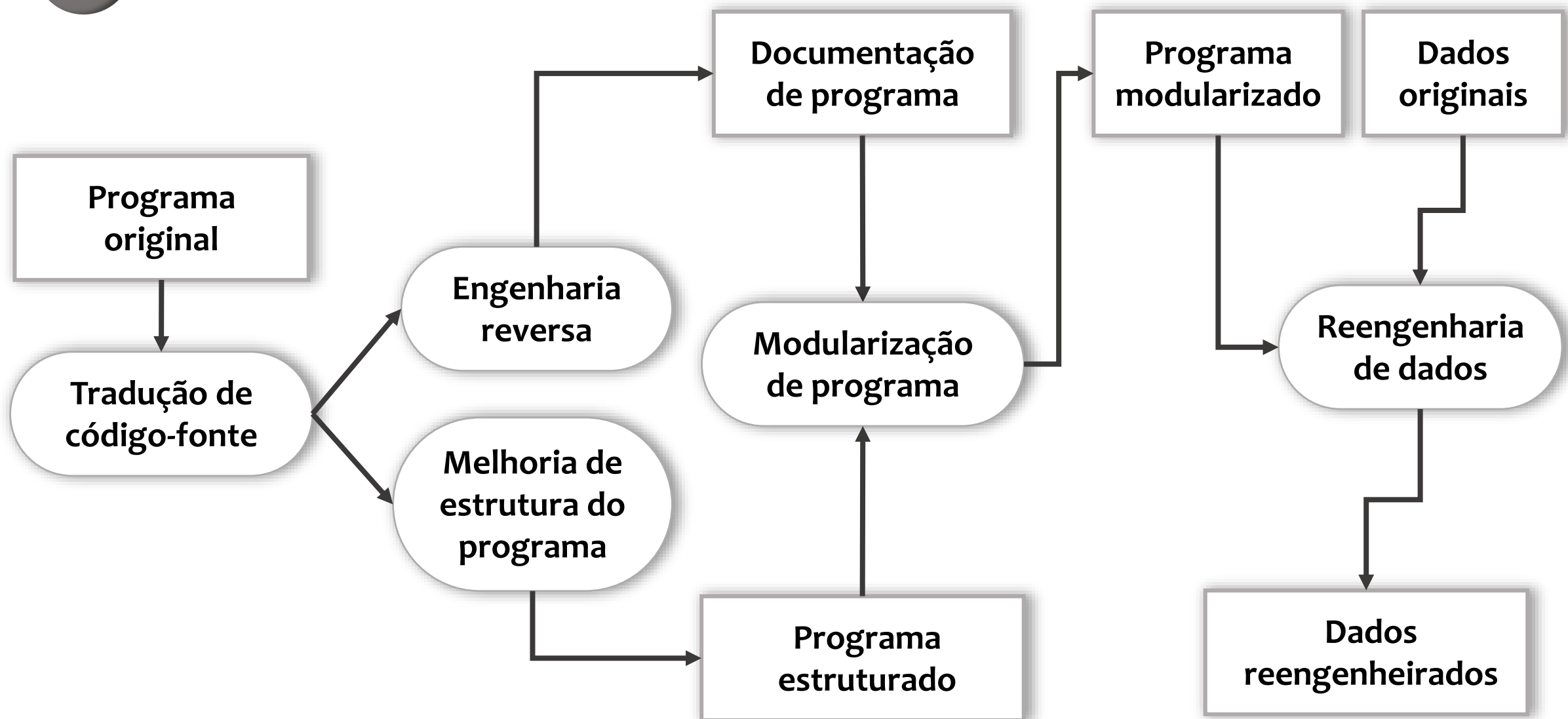


Reengenharia de software

PROCESSO DE REENGENHARIA



Processo de reengenharia



REENGENHARIA DE SOFTWARE

Os principais fatores que afetam os custos da reengenharia são:



A qualidade do software que deve passar pela reengenharia.



O apoio às ferramentas disponíveis para a reengenharia



A extensão da conversão de dados requerida



A disponibilidade de pessoal habilitado

Reestruturação automatizada
do programa

Reestruturação do programa e
dos dados



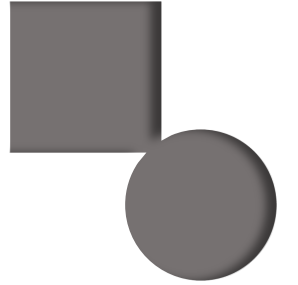
Conversão automatizada de
código-fonte

Reestruturação automatizada
com modificações manuais

Reestruturação mais
mudanças de arquitetura



Aumento dos custos



TRADUÇÃO DE CÓDIGO-FONTE

O processo mais simples da reengenharia de software.

TRADUÇÃO DE CÓDIGO-FONTE

A tradução do programa é o meio mais simples de proceder à reengenharia de software. Trata-se da **tradução automática** de um código-fonte de uma linguagem para o código-fonte de outra linguagem.

TRADUÇÃO DE CÓDIGO-FONTE

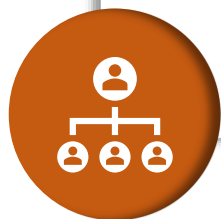


Atualização da
plataforma de
hardware.



Escassez de
pessoal
habilitado.

RAZÕES PARA TRADUÇÃO

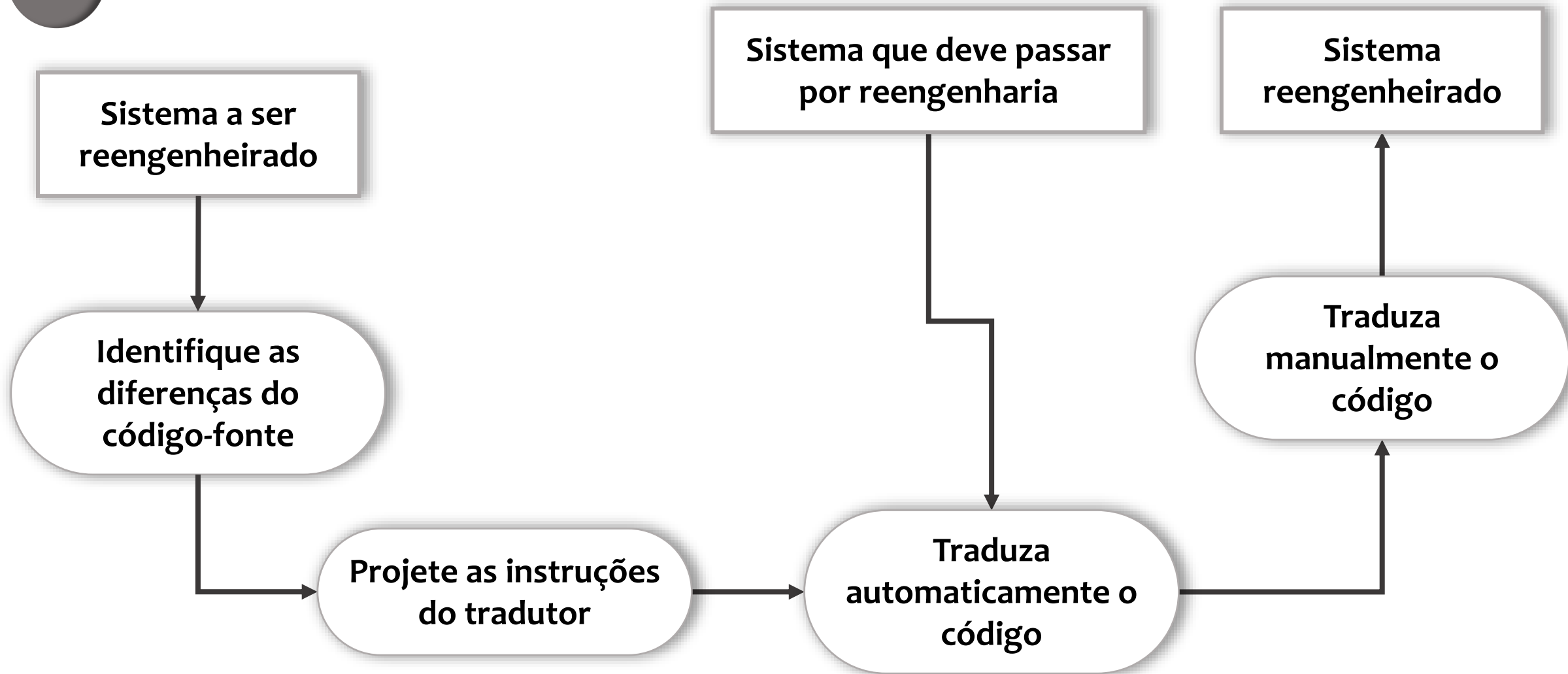


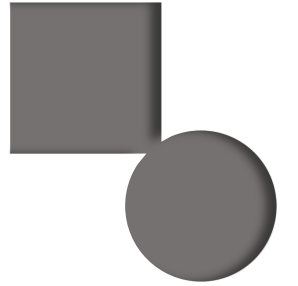
Mudanças na
política
organizacional



Falta de suporte
ao software.

Processo de tradução de programa.





ENGENHARIA REVERSA

Processo de análise de software com o objetivo de recuperar o projeto e a especificação.

ENGENHARIA REVERSA

A engenharia reversa é o processo de analisar o software com o objetivo de recuperar o projeto e sua especificação.

A engenharia reversa não é o mesmo que a reengenharia.

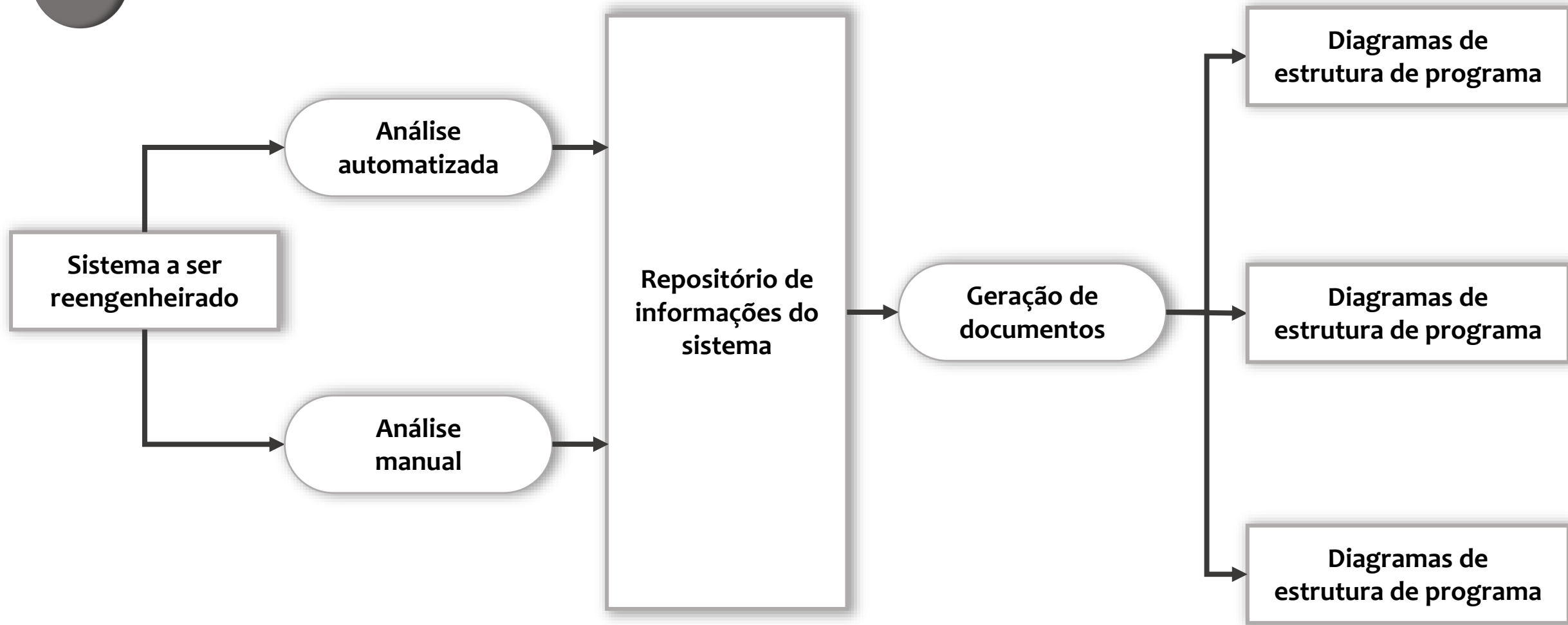
O objetivo da engenharia reversa é derivar o projeto ou a especificação de um sistema a partir de seu código-fonte.

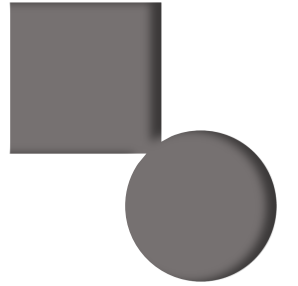
ENGENHARIA REVERSA

A engenharia reversa **pode ser** utilizada durante o processo de reengenharia de software, para recuperar o projeto do programa, afim de compreendê-lo antes de reorganizar sua estrutura.

A engenharia reversa **não precisa** ser sempre seguida da reengenharia.

Processo de tradução de programa.





MELHORIA DE ESTRUTURA DE PROGRAMA

A reestruturação do software modifica o código-fonte e os dados com o intuito de deixar suas modificações mais fáceis.

MELHORIA DE ESTRUTURA DE PROGRAMA

Sistemas legados não são bem estruturados:

- Estruturas de controle complexas, com muitas ramificações;
- Lógica contraintuitiva;
- Manutenções regulares acarretam em códigos intangíveis.

Programa de controle com lógica 'espaguete'

```
Start:  Get (Time-on, Time-off, Time, Setting, Temp, Switch)
        if Switch = off goto off
        if Switch = on goto on
        goto Cntrld
off:    if Heating-status = on goto Sw-off
        goto loop
on:     if Heating-status = off goto Sw-on
        goto loop
Cntrld: if Time = Time-on goto on
        if Time = Time-off goto off
        if Time < Time-on goto Start
        if Time > Time-off goto Start
        if Temp > Setting then goto off
        if Temp < Setting then goto on
Sw-off: Heating-status := off
        goto Switch
Sw-on:  Heating-status := on
Switch: Switch-heating
loop:   goto Start
```


loop

--A declaração GET acha valores para as variáveis dadas a partir do ambiente do sistema.

Get (Time-on, Time-off, Time, Setting, Temp, Switch);

case Switch of

when On => if Heating-status = off then

 Switch-heating; Switch-status := on;

end if;

when Off => if Heating-status = on then

 Switch-heating; Switch-status := off;

end if;

when Controlled =>

if Time >= Time-on and Time <= Time-off then

if Temp > Setting and Heating-status = on then

 Switch-heating; Heating-status = off;

elsif Temp < Setting and Heating-status = off then

 Switch-heating; Heating-status := on;

end if;

end if;

end case;

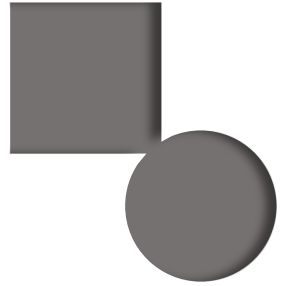
end loop;

MELHORIA DE ESTRUTURA DE PROGRAMA

Dentre os problemas de reestruturação automática, temos: a) perda de comentários; b) perda de documentação; e, c) grandes exigências de computação – pode levar muito tempo para reestruturação de grandes programas.

Para um programa para ser reestruturado, pode observar-se as seguintes **métricas**:

- Índice de falhas;
- Porcentagem de código-fonte modificada por ano; e
- Complexidade do componente.



MODULARIZAÇÃO DE PROGRAMA

Reorganização de um programa, de modo que as partes relacionadas sejam coletadas e consideradas um único módulo.

MODULARIZAÇÃO DE PROGRAMA

A modularização possibilita a remoção de redundâncias em componentes relacionados e otimiza interfaces.

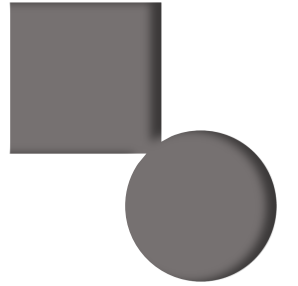
Podem ser criados vários tipos de módulos durante a modularização:

Abstração de
dados

Módulos de
hardware

Módulos
funcionais

Módulos de
apoio ao
processo



REENGENHARIA DE DADOS

É a transformação de dados de um formato para outro.

REENGENHARIA DE DADOS

A reengenharia de dados é o processo de análise e **reorganização** de **estruturas de dados** e, algumas vezes, os valores dos dados em um sistema, para torna-lo mais compreensível.

A reengenharia de dados não será necessária, se a funcionalidade do sistema permanecer inalterada.

REENGENHARIA DE DADOS

Degradação dos dados

Modificações nos dados introduzem erros, valores duplicados podem ser criados e mudanças no ambiente externo podem não estar refletidas nos dados.

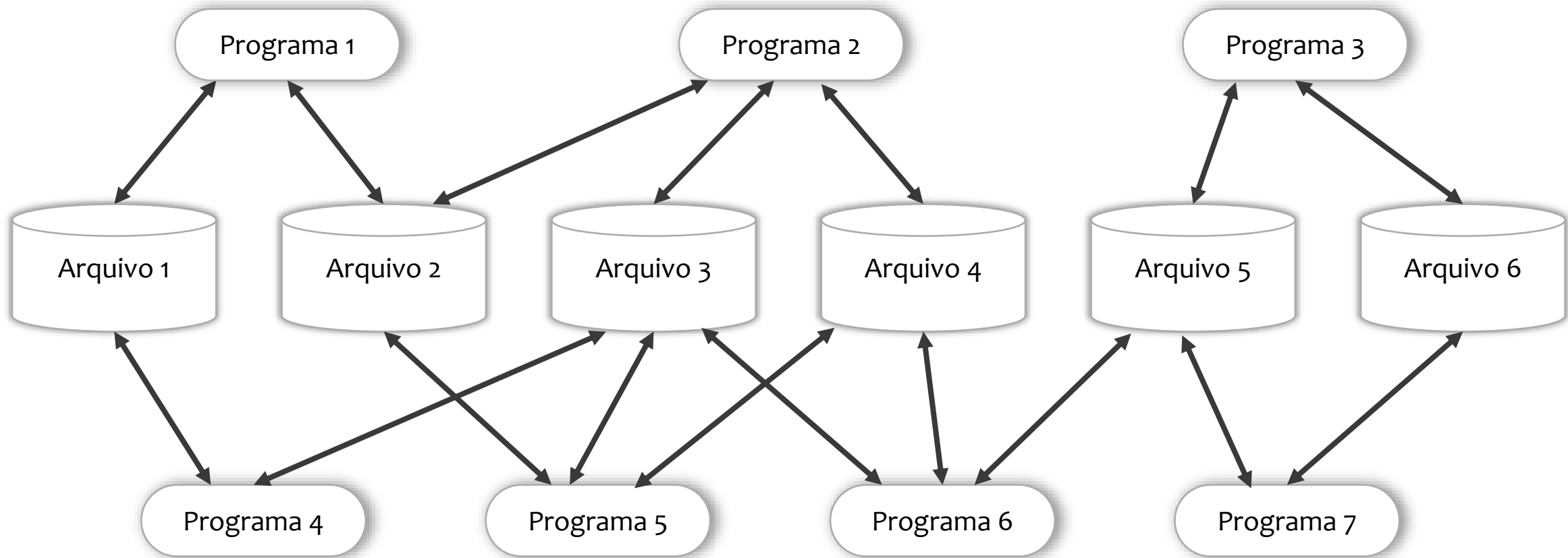
Limites inerentes inseridos no sistema

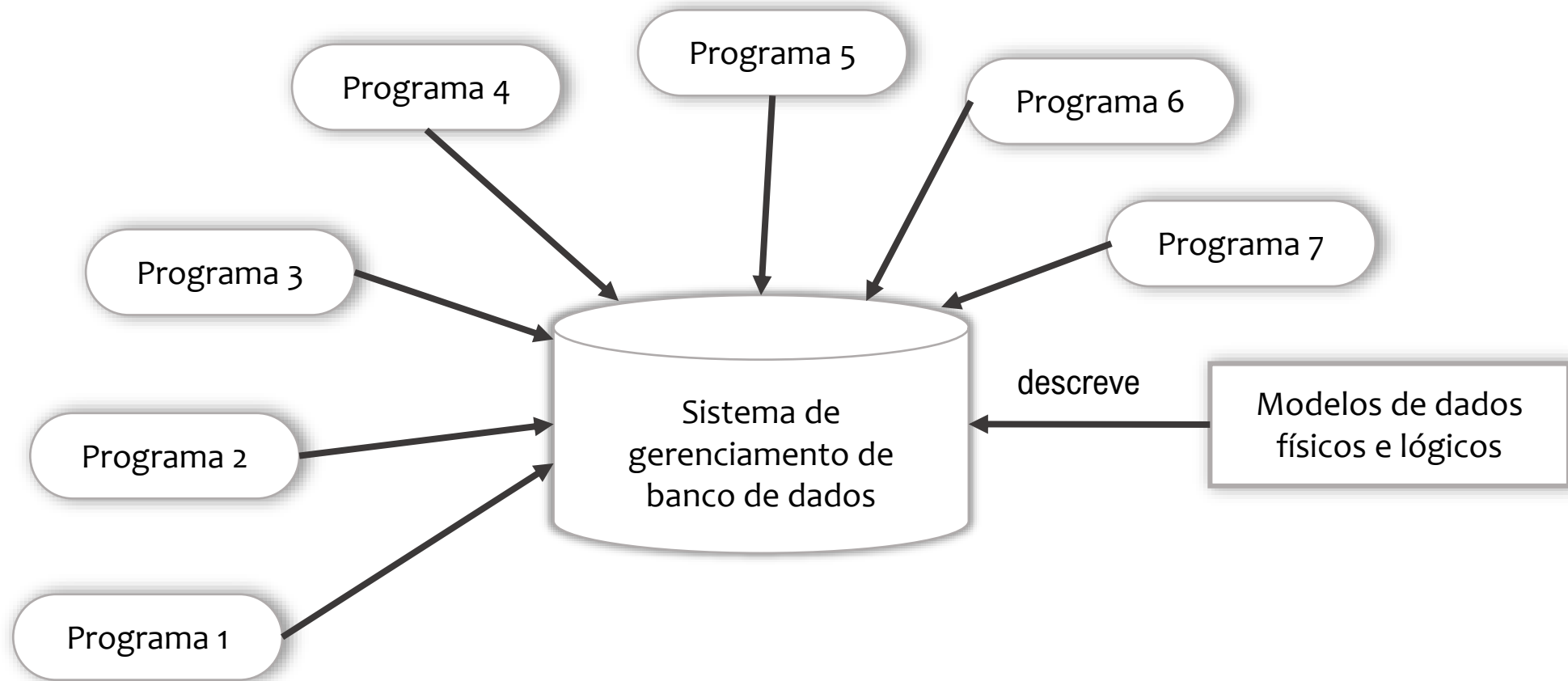
Sistemas legados, possuem limitações de processamento de dados. Atualmente o volume de dados a serem processados é muito maior.

Evolução da arquitetura

A migração de arquitetura centralizada para distribuída requer um sistema de gerenciamento de dados, para permitir acesso de clientes remotos.

Abordagem	Descrição
Limpeza de dados	Os registros e valores de dados são analisados, a fim de melhorar sua qualidade. As duplicações são removidas, as informações redundantes são excluídas e um formato consistente é aplicado a todos os registros. Normalmente, isso não deve requerer quaisquer mudanças nos programas associados.
Extensão de dados	Nesse caso, os dados e programas associados passam pelo processo de reengenharia, a fim de eliminar os limites no processamento de dados. Isso pode exigir mudanças nos programas para aumentar a extensão de campos, modificar limites superiores nas tabelas e assim por diante. Os dados em si podem precisar ser reescritos e limpos, para que reflitam as mudanças no programa.
Migração de dados	Nesse caso, ocorre a migração dos dados para o controle de um moderno sistema de gerenciamento de banco de dados. Os dados podem ser armazenados em arquivos separados ou ser gerenciados por um tipo de sistema de gerenciamento de banco de dados mais antigo.





REENGENHARIA DE DADOS

Principais problemas que podem surgir com os dados em sistema legados:

- Problemas com a denominação dos dados;
- Problemas com a extensão de campos;
- Problemas com organização de registros;
- Literais hard-coded;
- Nenhum dicionário de dados.

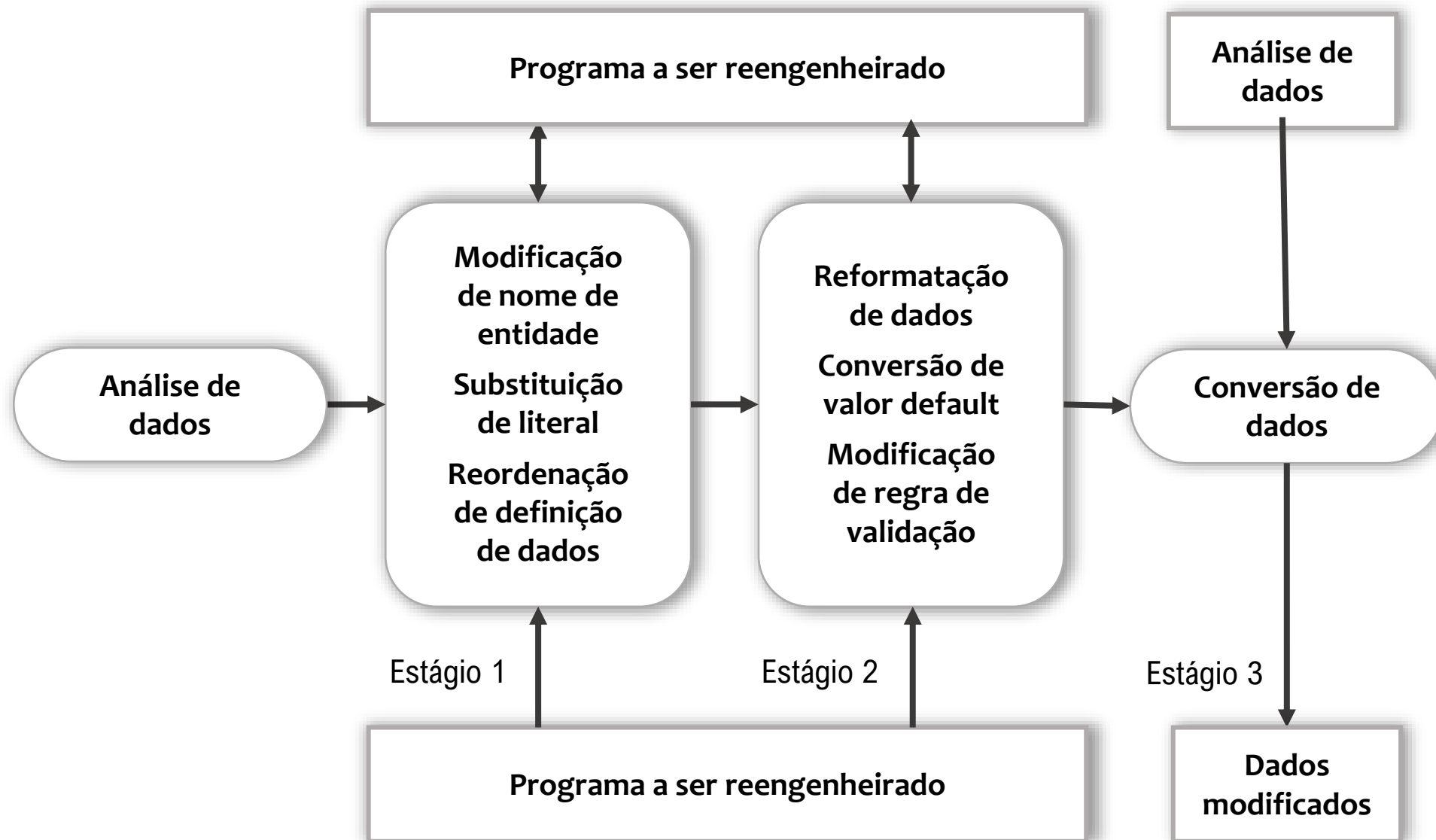
Inconsistência do valor dos dados.

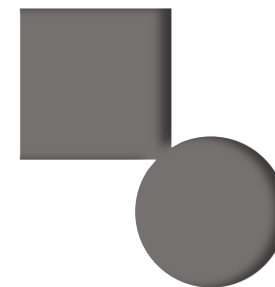
Abordagem	Descrição
Valores default inconsistentes	Diferentes programas atribuem diferentes valores default aos mesmos itens de dados lógicos. Isso causa problemas para outros programas que não sejam os que criaram os dados. Esse problema acontece quando aos valores que faltam é atribuído um valor default válido. Os dados que faltam não poderão ser descobertos.
Unidades inconsistentes	A mesma informação é representada em diferentes unidades, em diferentes programas. Por exemplo, nos Estados Unidos ou no Reino Unido, os dados referentes a peso podem ser representados em libras em programas mais antigos e em quilogramas em sistemas mais recentes. Um problema maior desse tipo surgiu na Europa, com a introdução da moeda única. Os sistemas legados foram escritos para lidar com unidades de moeda nacional, de modo que os dados precisam ser convertidos para o Euro.

Inconsistência do valor dos dados.

Abordagem	Descrição
Regras de validação	Diferentes programas aplicam diferentes regras de validação de dados. Os dados escritos para um programa podem ser rejeitados por outro. Esse é um problema específico para dados de arquivos que podem não ter sido atualizados de acordo com as mudanças nas regras de validação de dados.
Semânticas de representação inconsistentes	Os programas presumem algum significado no modo como os itens são representados. Por exemplo, alguns programas podem presumir que o texto com letra maiúscula representa um endereço. Eles podem utilizar diferentes convenções e, portanto, rejeitar dados que sejam semanticamente válidos.
Manuseio inconsistente de valores negativos	Alguns programas rejeitam valores negativos para entidades que devem ser sempre positivas. Outros, contudo, podem aceita-las como valores negativos ou deixar de reconhecê-las como negativas e convertê-las para um valor positivo.

Processo de reengenharia de dados.





PONTOS-CHAVE

PONTOS-CHAVE

O **objetivo** da reengenharia de sistemas é melhorar a estrutura do sistema e torna-la mais fácil de ser compreendida. O custo da manutenção futura do sistema deve, portanto, ser reduzido.

O **processo** de reengenharia inclui tradução de código-fonte, engenharia reversa, melhoria de estrutura de programa, modularização de programa e reengenharia de dados.

PONTOS-CHAVE

A **tradução de código-fonte** é a conversão automatizada de um programa escrito em uma linguagem de programação para outra linguagem. Ela pode ser necessária quando a linguagem de programação original for obsoleta.

A **engenharia reversa** é o processo de derivar um projeto e a especificação de sistema a partir de seu código-fonte. Ferramentas como os browsers de programa podem ser utilizadas para ajudar no processo.

PONTOS-CHAVE

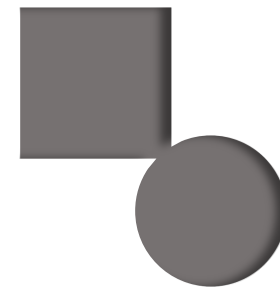
A melhoria da estrutura de programa envolve substituir construções não estruturadas de controle, como declarações *goto*, por *loops while* e declarações condicionais. Esse processo pode ser automatizado.

A **modularização** de programa envolve reorganizar o código-fonte de um programa, a fim de agrupar itens relacionados. Isso os torna mais fáceis de ser compreendidos e modificados.

PONTOS-CHAVE

A reengenharia de dados pode ser necessária, devido ao gerenciamento inconsistente de dados pelos programas em um sistema legado. O objetivo da reengenharia de dados pode ser a reengenharia de todos os programas, a fim de utilizar um banco de dados comum.

Os custos da reengenharia de dados serão aumentados significativamente se os dados existentes tiverem de ser convertidos para um novo formato.



ATIVIDADE SUGERIDA

ATIVIDADE SUGERIDA

Uma empresa estabelece, de maneira rotineira, condições contratuais para os programadores *free lancers* que trabalham na reengenharia de suas aplicações, o que os impede de aceitar contratos com empresas similares. A razão para isso é que a reengenharia, inevitavelmente, revela informações de negócios. Essa é uma atitude razoável por parte da empresa, considerando que eles não têm nenhuma obrigação com a contratante depois que seus contratos terminam?