

---

UNIP – Universidade Paulista  
Campus Tatuapé – SP  
Ciência da Computação  
Engenharia de Software

Reengenharia de Software

Prof. Marcelo Nogueira

---

---

---

---

---

---

---

---

## Reengenharia de Software

---

- Reorganizar e modificar os sistemas de software existentes para aumentar o seu ciclo de vida.

---

---

---

---

---

---

---

---

## Objetivos

---

- Compreender por que a reengenharia é uma opção eficaz, em termos de custos, para a evolução do sistema de software
- Compreender as atividades, como engenharia reversa e reestruturação de programa, que podem estar envolvidas no processo de reengenharia de software
- Compreender as diferenças entre reengenharia de software e de dados e por que a reengenharia é um processo dispendioso e demorado

---

---

---

---

---

---

---

---

## Tópicos

---

- Tradução de código-fonte
- Engenharia reversa
- Melhoria de estrutura de programa
- Modularização de programa
- Reengenharia de dados

---

---

---

---

---

---

---

---

## Reengenharia de sistema

---

- Reestruturar ou reescrever todo ou parte de um sistema legado sem alterar a sua funcionalidade
- Aplicável quando alguns, mas nem todos os sub-sistemas de um sistema maior exigem manutenção freqüente
- Reengenharia envolve um esforço adicional para torná-los mais fáceis de manter. O sistema pode ser reestruturado e redocumentado

---

---

---

---

---

---

---

---

## Quando aplicar reengenharia

---

- Quando as mudanças de sistema estão confinadas a uma parte do sistema, então faz a reengenharia dessa a parte
- Quando o suporte de hardware ou software se torna obsoleto
- Quando ferramentas de apoio à reestruturação estão disponíveis

---

---

---

---

---

---

---

---

## Vantagens da reengenharia

- **Riscos reduzidos**
  - Existe um alto risco em desenvolver um software novo que seja essencial. Podem ser cometidos erros na especificação do sistema e ocorrer problemas de desenvolvimento
- **Custos reduzidos**
  - O custo da reengenharia é significativamente menor do que os custos de desenvolvimento de um novo software

---

---

---

---

---

---

---

---

## Reengenharia de processo de negócio

- Se ocupa de reprojeter processos de negócios, a fim de reduzir o número de atividades redundantes e melhorar a eficiência do processo
- Em geral, depende da introdução ou do aumento do apoio baseado em computadores para o processo
- Muitas vezes, a reengenharia de processo é um fator direcionador da evolução de software, uma vez que os sistemas legados podem incorporar dependências implícitas dos processos existentes

---

---

---

---

---

---

---

---

## Atividades no processo de reengenharia

- **Tradução de código-fonte**
  - O programa é convertido de uma linguagem de programação antiga para uma versão mais moderna da mesma linguagem ou para um linguagem diferente
- **Engenharia reversa**
  - O programa é analisado e as informações são extraídas dele, a fim de ajudar a documentar sua organização e funcionalidade
- **Melhoria de estrutura do programa**
  - A estrutura de controle do programa é analisada e modificada, a fim de torná-la mais fácil de ser lida e compreendida
- **Modularização de programa**
  - As partes relacionadas do programa são agrupadas e, quando for apropriado, a redundância é removida
- **Reengenharia de dados**
  - Os dados processados pelo programa são modificados, a fim de refletir as mudanças feitas nele

---

---

---

---

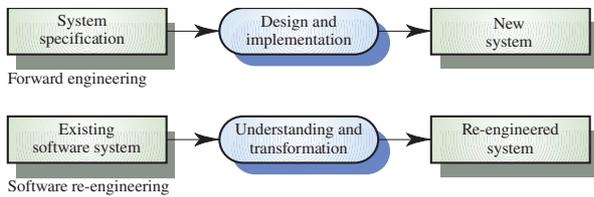
---

---

---

---

## Engenharia direta e reengenharia



©Ian Sommerville 2000

Software Engineering, 6th edition, Chapter 28

Slide 10

---

---

---

---

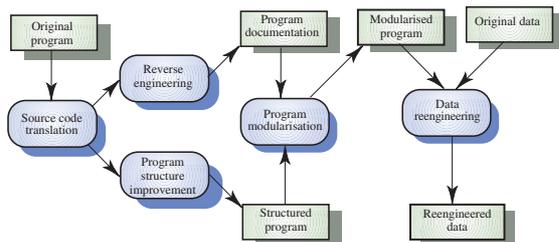
---

---

---

---

## Processo de reengenharia



©Ian Sommerville 2000

Software Engineering, 6th edition, Chapter 28

Slide 11

---

---

---

---

---

---

---

---

## Fatores de custos da reengenharia

- 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

©Ian Sommerville 2000

Software Engineering, 6th edition, Chapter 28

Slide 12

---

---

---

---

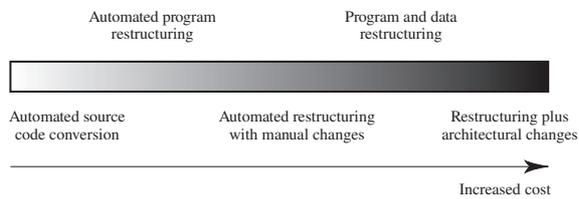
---

---

---

---

## Abordagens da reengenharia



---

---

---

---

---

---

---

---

## Tradução de código-fonte

- Converter um código de uma linguagem (ou versão) para outra, p.e. FORTRAN para C
- Pode ser necessária pelas seguintes razões:
  - Atualização da plataforma de hardware
  - Escassez de pessoal habilitado
  - Mudanças na política organizacional
  - Falta de suporte ao software
- Somente é realista se um tradutor automático estiver disponível

---

---

---

---

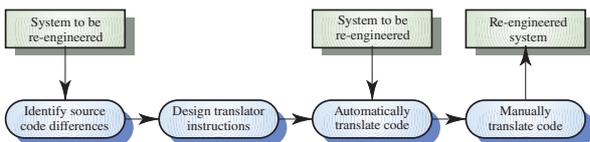
---

---

---

---

## Processo de tradução de programa



---

---

---

---

---

---

---

---

## Engenharia reversa

- É o processo de analisar o software com o objetivo de recuperar seu projeto e sua especificação
- A engenharia reversa pode fazer parte do processo de reengenharia, mas 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; um novo sistema, com manutenção mais fácil
- O processo inicia com uma fase de análise, utilizando-se de ferramentas automatizadas, a fim de descobrir sua estrutura
- As ferramentas para a compreensão do programa (browsers de armazenamento de informações, geradores de referência cruzada, etc.) podem ser utilizadas para apoiar o processo

---

---

---

---

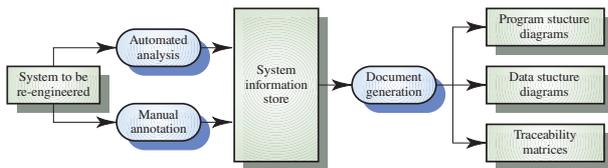
---

---

---

---

## Processo de engenharia reversa



---

---

---

---

---

---

---

---

## Engenharia reversa

- Engenharia reversa é utilizada durante o processo de reengenharia, mas não precisa ser sempre seguida da reengenharia
  - O projeto e a especificação de um sistema existente podem passar por engenharia reversa, de modo que possam servir como uma entrada à especificação de requisitos, para a substituição desse programa.
  - Como alternativa, o projeto e a especificação podem passar por engenharia reversa, de modo que estejam disponíveis para ajudar na manutenção do programa

---

---

---

---

---

---

---

---

## Melhoria de estrutura de programa

- A estrutura de controle dos sistemas legados é complexa, com muitas ramificações incondicionais e a lógica de controle não é intuitiva. Essa estrutura pode ser afetada por manutenções regulares, tornando alguns códigos inatingíveis
- O programa pode ser reestruturado automaticamente para eliminar declarações incondicionais
- Condições complexas podem ser simplificadas, como parte do processo de reestruturação de programa

---

---

---

---

---

---

---

---

## Lógica 'espaguete'

```
Start:  Get (Time-on, Time-off, Time, Setting, Temp, Switch)
        if Switch = off goto off
        if Switch = on goto on
        goto Cntrlid
off:    if Heating-status = on goto Sw-off
        goto loop
on:     if Heating-status = off goto Sw-on
        goto loop
Cntrlid: 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
```

---

---

---

---

---

---

---

---

## Lógica estruturada de controle

```
loop
-- The Get statement finds values for the given variables from the system's
-- environment.
Get (Time-on, Time-off, Time, Setting, Temp, Switch) ;
case Switch of
  when On => if Heating-status = off then
    Switch-heating ; Heating-status := on ;
  end if ;
  when Off => if Heating-status = on then
    Switch-heating ; Heating-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 ;
```

---

---

---

---

---

---

---

---

## Simplificação de condição

- Complex condition  
**if not (A > B and (C < D or not ( E > F ) ) )...**
- Simplified condition  
**if (A <= B and (C >= D or E > F)...**

---

---

---

---

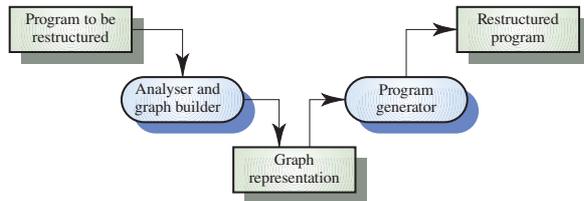
---

---

---

---

## Reestruturação automatizada de programa



---

---

---

---

---

---

---

---

## Problemas com a reestruturação

- Problemas com a reestruturação são:
  - Perda de comentários
  - Perda de documentação
  - Grandes exigências de computação
- Se o programa é dirigido por dados, com componentes estreitamente acoplados, por meio de estrutura de dados compartilhada, a reestruturação de código pode não levar a uma melhoria significativa na facilidade de compreensão
- Se o programa for escrito em um dialeto de linguagem que não seja um padrão, as ferramentas de reestruturação-padrão poderão não funcionar adequadamente e pode ser necessária uma significativa intervenção manual

---

---

---

---

---

---

---

---

## Modularização de programa

- É o processo de reorganizar um programa, de modo que as partes relacionadas sejam coletadas e consideradas um único módulo
- A modularização em geral é realizada manualmente, com a inspeção e a edição do código

---

---

---

---

---

---

---

---

## Tipos de módulos

- **Abstração de dados**
  - São os tipos de dados abstratos, criados a partir da associação de dados com componentes de processamento
- **Módulos de hardware**
  - Esses módulos estão estreitamente relacionados com as abstrações de dados e agrupam todas as funções utilizadas para controlar um determinado dispositivo de hardware
- **Módulos funcionais**
  - São os módulos de programa que agrupam as funções que realizam tarefas semelhantes ou estreitamente relacionadas
- **Módulos de apoio ao processo**
  - São os módulos nos quais são agrupados todas as funções e todos os itens específicos de dados requeridos para apoiar um processo de negócios específicos

---

---

---

---

---

---

---

---

## Recuperação de abstrações de dados

- Muitos sistemas legados se baseiam no uso de tabelas compartilhadas e em áreas de dados comuns
- Fazer modificações nessa áreas é uma tarefa dispendiosa, pois é preciso analisar o impacto das modificações em todo o sistema
- As áreas de dados compartilhados podem ser convertidos para um objeto ou tipos de dados abstratos (ADTs)

---

---

---

---

---

---

---

---

## Recuperação de abstrações de dados

- Etapas envolvidas na conversão de áreas globais:
  - Analisar áreas comuns de dados para identificar suas abstrações lógicas
  - Criar um objeto ou ADT para cada uma dessas abstrações
  - Utilizar um sistema de browsing, ou um gerador de referência cruzada, para encontrar todas as referências aos dados

---

---

---

---

---

---

---

---

## Reengenharia de dados

- Processo de análise e reorganização de estruturas de dados e, algumas vezes, os valores dos dados em um sistema, para torná-lo mais compreensível
- Em princípio, a reengenharia de dados não deverá ser necessária, se a funcionalidade do sistema permanecer inalterada
- Na prática, há uma série de razões pelas quais é preciso modificar os dados, como também os programas, em um sistema legado

---

---

---

---

---

---

---

---

## Razões para modificar os dados

- Degradação dos dados
  - Com o passar do tempo, a qualidade dos dados tende a diminuir. As modificações nos dados introduzem erros, valores duplicados podem ter sido criados e as mudanças no ambiente externo podem não estar refletidas nos dados
- Limites inerentes inseridos no sistema
  - Quando os programas foram originalmente projetados foram incluídas restrições quanto à quantidade de dados, contudo é necessário que os programas processem muito mais dados que o volume originalmente imaginado
- Evolução de arquitetura
  - Se um sistema centralizado migrar para uma arquitetura distribuída, é essencial que o núcleo dessa arquitetura seja um sistema de gerenciamento de dados, que pode ser acessado a partir de clientes remotos

---

---

---

---

---

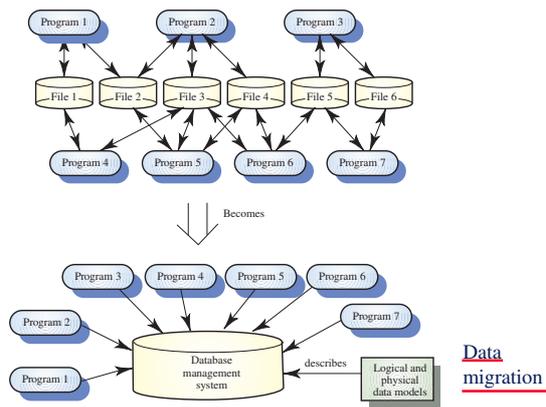
---

---

---

## Abordagens da reengenharia de dados

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	Nessa 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 antigos.



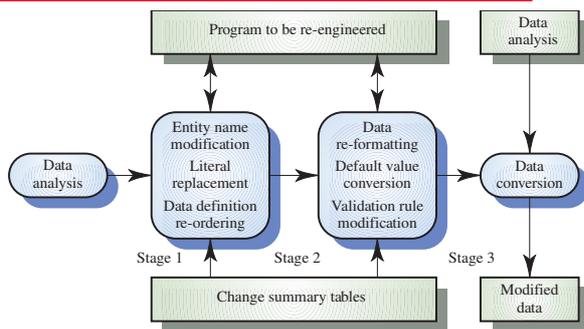
## Problemas com dados

- Problemas com a denominação dos dados
  - Os nomes podem ser obscuros e de difícil compreensão. O mesmo nome pode ser usado em diferentes programas com significados diferentes
- Problemas com e extensão de campos
  - Para o mesmo item podem ser designadas diferentes extensões, em diferentes programas
- Problemas com a organização de registros
  - Os registros que representam a mesma entidade podem ser organizados diferentemente, em diferentes programas
- Código com baixa legibilidade
- Nenhum dicionário de dados

## Inconsistência do valor dos dados

Inconsistência dos dados	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.
Regras de validação inconsistentes	Diferentes programas aplicam diferentes regras de validação da 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 aceitá-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

- O objetivo da reengenharia é melhorar a estrutura do sistema e torná-la mais fácil de ser compreendida e mantida
- O processo de reengenharia inclui tradução de código-fonte, engenharia reversa, melhoria de estrutura de programa, modularização de programa e reengenharia de dados
- A tradução de código-fonte é a conversão automatizada de um programa escrito em uma linguagem de programação para outra linguagem

## Pontos-chave

---

- A engenharia reversa é o processo de derivar um projeto e a especificação de sistema a partir de seu código-fonte
- A melhoria da estrutura de programa envolve substituir construções não estruturadas de controle, como declarações goto, por loops while e declarações condicionais
- A modularização de programa envolve reorganizar o código-fonte de um programa, a fim de agrupar itens relacionados
- A reengenharia de dados pode ser necessária, devido ao gerenciamento inconsistente de dados pelos programas em um sistema legado

---

---

---

---

---

---

---

---

## Bibliografia

---

- SOMMERVILLE, Ian, Engenharia de Software. 6 ed. São Paulo: Addison Wesley, 2003.
- <http://www.comp.lancs.ac.uk/computing/resources/IanS/SE6/Slides/PPT/ch28.ppt>

---

---

---

---

---

---

---

---