Reengenharia de Software

Profa. Márcia Häfele Islabão Franco

Introdução

- A manutenção de software pode ser responsável por mais de 60% de todo o esforço despendido por uma equipe de desenvolvimento e a porcentagem continua a crescer à medida que mais softwares são produzidos.

- Outra razão para o problema de manutenção de software é a mobilidade da equipe. É provável que a equipe que desenvolveu o projeto original, ou que modificaram o sistema, não esteja mais envolvida no projeto.

Introdução

- A manutenção de software pode ser definida pela identificação de quatro diferentes atividades:
  - manutenção corretiva: correção de falhas.
  - manutenção adaptativa: adaptar o software a um ambiente operacional diferente.
  - manutenção perfectiva (ou de melhoria): adicionar ou modificar funcionalidades do sistema.
  - manutenção preventiva: prevenir possíveis erros.
- Apenas 20% de todo o trabalho de manutenção é gasto “consertando erros”. Os restantes 80% são gastos adaptando sistemas existentes a modificações no seu ambiente externo, fazendo melhorias solicitas por usuários e submetendo uma aplicação a reengenharia, para uso futuro.

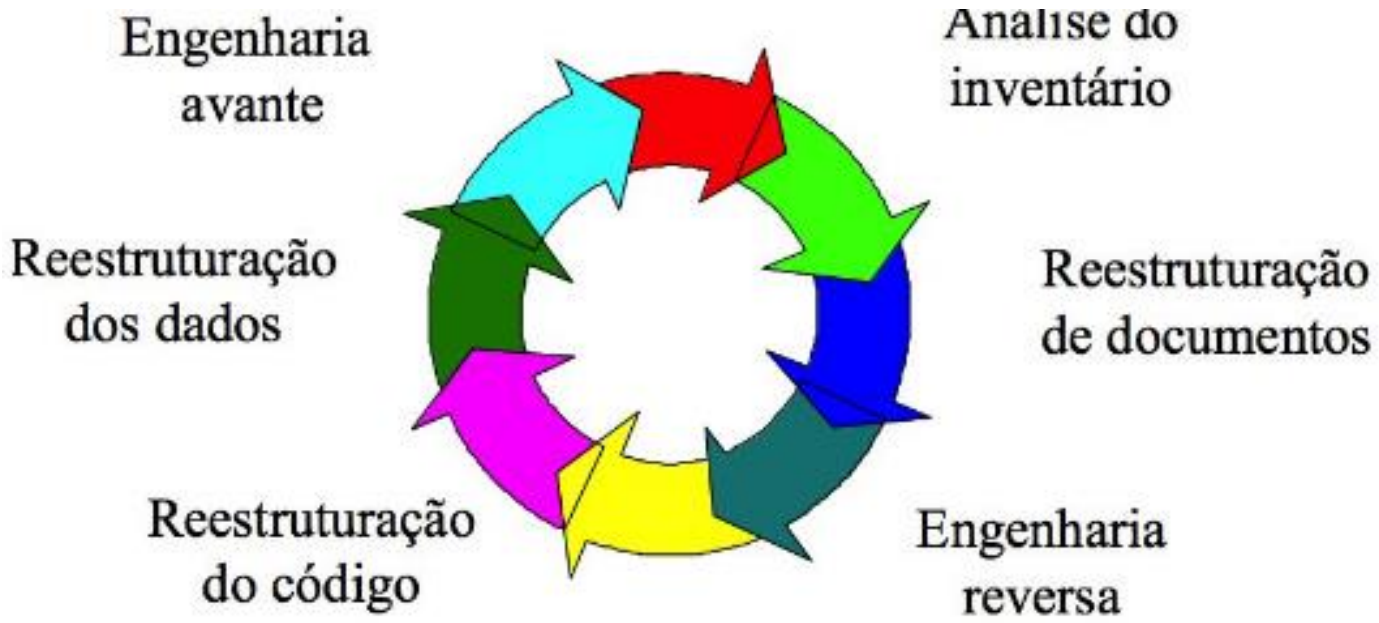
Modelo de Processo de Reengenharia de Software

- A reengenharia leva tempo, tem um custo significativo, e absorve recursos que poderiam, por outro lado, ser usados em necessidades imediatas.

- A reengenharia não é conseguida em poucos meses e é uma atividade que absorve recursos de tecnologia.

- Necessário que a empresa tenha um planejamento para a reengenharia de software.

Modelo de Processo de Reengenharia de Software



Modelo de Processo de Reengenharia de Software (Pressman, 2018)

Modelo de Processo de Reengenharia de Software

Cada uma das atividades pode ser revisitada

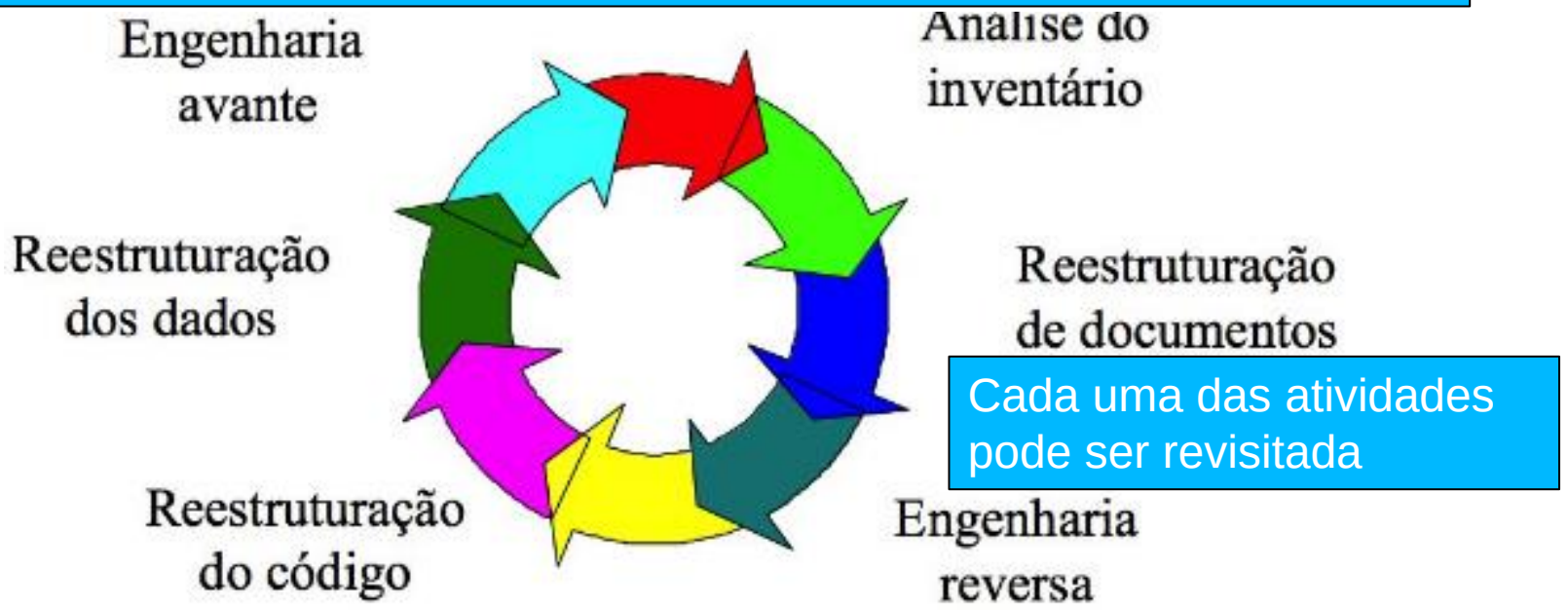
Modelo de Processo de Reengenharia de Software

Cada uma das atividades pode ser revisitada

Para qualquer ciclo particular, o processo pode terminar depois de qualquer uma dessas atividades

Modelo de Processo de Reengenharia de Software

As atividades não precisam ocorrer em uma sequência linear (pode ser que a engenharia reversa tenha que ocorrer antes que a reestruturação de documentos por exemplo)



Cada uma das atividades pode ser revisitada

Para qualquer ciclo particular, o processo pode terminar depois de qualquer uma dessas atividades

Análise de Inventário

- Toda empresa deve ter um inventário de todas as aplicações. O inventário pode ser mais do que um modelo de planilha contendo informações que fornece descrição detalhada de cada aplicação ativa (por exemplo, tamanho e idade do software). Ordenando essas informações de acordo com a longevidade, manutenibilidade corrente e outros critérios localmente importantes, aparecem candidatos a reengenharia. Dessa forma, recursos podem então ser alocados para aplicações candidatas a trabalho de reengenharia.
- O inventário deve ser revisitado em um ciclo regular. O estado de aplicações pode mudar em função do tempo e, como resultado, as prioridades para reengenharia se deslocarão.

Reestruturação de Documentos

- Pouca documentação é a marca registrada de muitos sistemas herdados. Mas o que fazer a respeito? Quais são as opções?
  1. Criação de documentação consome tempo demais – Se o sistema funciona, deve-se conviver com o que se tem. Em alguns casos, essa é a abordagem correta. Não é possível recriar documentação para centenas de programas de computador.
  2. A documentação precisa ser atualizada, mas se tem recursos limitados – Neste caso, deve-se utilizar a abordagem “documentar quando tocar”. Pode não ser necessário redocumentar totalmente uma aplicação. Em vez disso, as partes do sistema que estão atualmente sofrendo modificações são totalmente documentadas. Ao longo do tempo, uma coleção de documentação útil e relevante vai evoluir.
  3. O sistema é critico para o negócio e precisa ser totalmente documentado – Uma abordagem inteligente, nesse caso, é limitar a documentação ao mínimo essencial.

Cada uma dessas opções é viável. A equipe deve escolher aquela que é mais adequada a cada caso.

Engenharia Reversa

- Processo de análise de um programa, em um esforço para representá-lo em uma abstração mais alta do que o código-fonte.
- A engenharia reversa é um processo de recuperação de projeto.
- As ferramentas de engenharia reversa extraem informações do projeto de dados, arquitetural e procedimental, para um programa existente.

Reestruturação de Código

- O tipo mais comum de reengenharia.
- Alguns sistemas legados tem uma arquitetura de programa relativamente sólida, mas módulos individuais foram codificados de um modo que se torna difícil entendê-los, testá-los e mantê-los. Nestes casos, o código dos módulos suspeitos pode ser reestruturado.
- Para realizar essa atividade, o código-fonte é analisado usando uma ferramenta de reestruturação. O código resultante reestruturado é revisado e testado para garantir que nenhuma anomalia foi introduzida. A documentação interna do código é atualizada.

Reestruturação de Dados

- Um programa com arquitetura de dados fraca será difícil de adaptar e aperfeiçoar. De fato, para muitas aplicações, a arquitetura de dados está mais relacionada à viabilidade do programa no longo prazo do que o código-fonte propriamente dito.
- Diferentemente da reestruturação de código, que ocorre em um nível relativamente baixo de abstração, a reestruturação de dados é uma atividade de reengenharia de escala plena. Na maioria dos casos, a reestruturação de dados começa com uma atividade de engenharia reversa. A arquitetura de dados atual é dissecada e os modelos de dados necessários são definidos. Objetos de dados e atributos são identificados e as estruturas de dados existentes são revisadas quanto à qualidade.
- Como a arquitetura de dados tem uma forte influência na arquitetura do programa e nos algoritmos que o constituem, as modificações nos dados resultarão invariavelmente tanto em modificações arquiteturais quanto de código.

Engenharia Avante

- Também chamada de renovação ou recomposição.
- Não apenas recupera informações de projeto de software existente, mas usa essa informação para alterar ou reconstituir o sistema existente em um esforço para aperfeiçoar sua qualidade global.
- Na maioria dos casos, o software trabalhado por reengenharia reimplementa a função do sistema existente e também adiciona novas funções e/ou melhora o desempenho geral.

Referências