КСПТ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ПРОГРАММНЫЕ ТЕХНОЛОГИИ

Реверс-инжиниринг программного обеспечения

Технологии разработки программного обеспечения

2014

Реверс-инжиниринг

- Реверс-инжиниринг ПО (обратный инжиниринг, обратная разработка) – процесс восстановления структуры, внутреннего устройства программы с целью понимания его принципа её работы

Цели проведения реверс-инжиниринга

- Извлечение спецификации
- Извлечение моделей
  - Выявление архитектуры программы
  - Получение алгоритмов работы
  - Извлечение моделей
- Понимание работы программы
  - program understanding
  - деобфускация
- Восстановление исходного кода
- …

Результаты проведения реверс-инжиниринга

- Диаграммы классов
- Диаграммы компонентов
- Диаграммы модулей
- Диаграммы состояний
- Диаграммы последовательностей
- Схемы алгоритмов
- Модели данных
- И т.п.

Предмет анализа реверс-инжиниринга

- Исходный код ПО
- Бинарный код ПО
- Байт-код
- Программная документация

Методы проведения реверс-инжиниринга

- Статический анализ программ
  - Обратная трассировка
- Динамический анализ программ
  - Анализ трасс исполнения
- Комбинированный анализ

Проблемы проведения реверс-инжиниринга

- Оптимизация кода
  - Машинно-зависимая оптимизация
  - Машинно-независимая оптимизация
  - Распараллеливание
- Обфускация кода
- В общем случае задача реверс-инжиниринга неразрешима!

Применение реверс инжиниринга

- Понимание legacy кода
- Подготовка к реинжинирингу
- Использование части legacy кода
- Копирование функциональности ПО без нарушения авторских прав *

- (*) Реверс-инжиниринг может быть запрещен производителем

Реверс-инжиниринг и качество ПО

- Функциональность
- Надежность
- Практичность
- Эффективность
- **Сопровождаемость**
  - **Анализируемость**
  - **Изменяемость**
  - **Стабильность**
  - **Тестируемость**
- Мобильность

Инструменты реверс-инжиниринга

КСПТ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ПРОГРАММНЫЕ ТЕХНОЛОГИИ

Реинжиниринг программного обеспечения

Технологии разработки программного обеспечения

2014

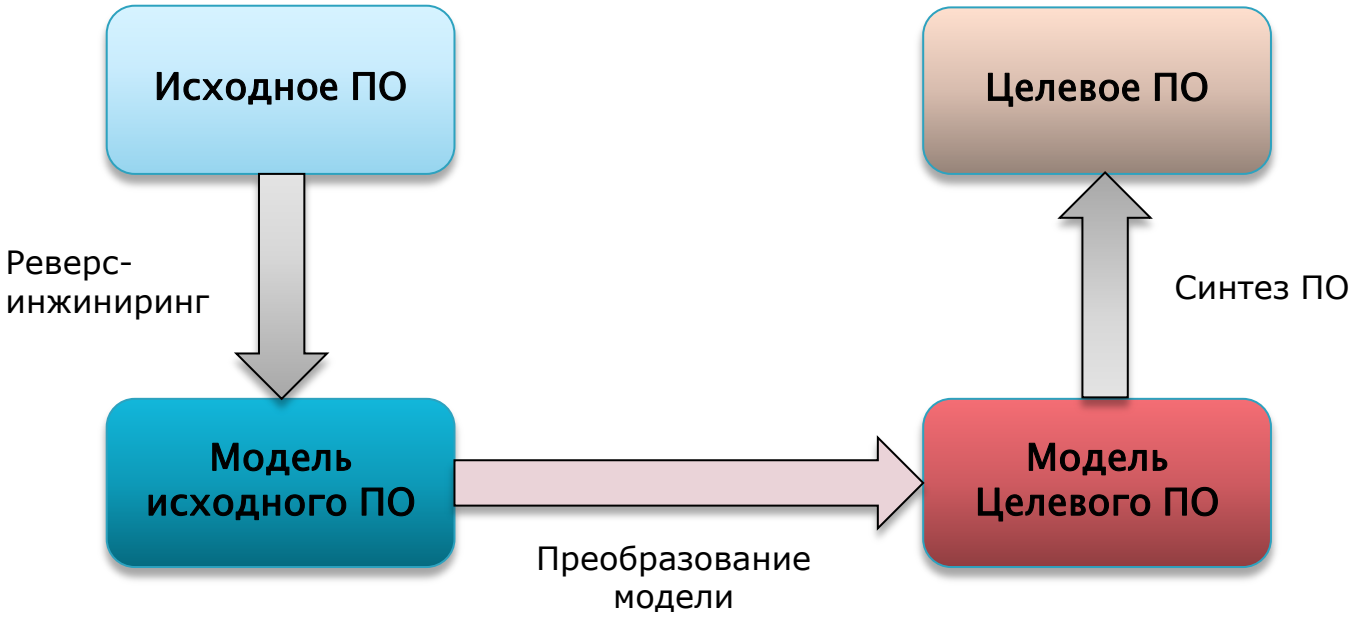
Реинжиниринг ПО

- Термин введен в 1990 году
- Реинжиниринг ПО – процесс создания нового программного обеспечения на основе использования существующего ПО, использующегося для той же или похожей задачи.
- Чаще всего реинжиниринг проводится с использованием реверс-инжиниринга

Виды реинжиниринга

- Эквивалентные преобразования
  - Рефакторинг
  - Машинно-независимая оптимизация
  - Распараллеливание
- Неэквивалентные преобразования
  - Машинно-зависимая оптимизация
  - Обфускация
  - **Модификация ПО**

Схема проведения реинжиниринга



Реинжиниринг и качество ПО

- Функциональность
- Надежность
- Практичность
- Эффективность
- **Сопровождаемость**
  - **Анализируемость**
  - **Изменяемость**
  - **Стабильность**
  - **Тестируемость**
- **Мобильность**
  - **Адаптируемость**
  - **Простота установки (внедрения)**
  - **Соответствие стандартам (подчинение стандартам или соглашениям, относящимся к мобильности)**
  - **Взаимозаменяемость**

Целесообразность проведения реинжиниринга

- Что проще сделать реинжиниринг или провести разработку заново?
- Определяется:
  - Качеством исходного ПО
  - Новыми функциональными требованиями к ПО
  - Нефункциональными требованиями: время, стоимость
  - Квалификацией команды разработчиков
- Реинжиниринг обычно требует более высокой квалификации разработчиков, архитекторов, тестировщиков

Методы и инструменты реинжиниринга

- Системы перезаписи (rewriting systems)
  - Системы перезаписи термов
  - Системы перезаписи строк
  - Преобразователи текстов на основе параметризируемых шаблонов
- Универсальные системы преобразования программ
  - Stratego/XT (strategoxt.org)– язык и программные инструменты для трансформации программ
  - ASF+SDF (www.meta-environment.org/Meta-Environment/ASF+SDF)– интерактивная среда анализа и преобразования программ
  - TXL (txl.ca) – язык программирования (DSL) для поддержки анализа и преобразования исходных текстов
  - DMS Software Reengineering Toolkit (www.semanticdesigns.com) – набор инструментов анализа и преобразования программ

Язык TXL

- TXL – Turing eXtender Language
- Разработан в Queen's University at Kingston, Software Technology Laboratory
- Автор и идеолог – проф. Джеймс Корди
- Программа TXL состоит из двух частей
  - Грамматика языка исходного текста
  - Правил преобразования
- Грамматика языка исходного текста
  - Задается декларативно
  - Аналог расширенных форм Бэкуса–Науэра
- Правила преобразования
  - Функциональный язык
  - Использует рекурсивные правила преобразования, основанные на шаблонах (pattern–matching) и правилах перезаписи (rewriting rules)

Язык TXL

- Примеры задач, которые могут быть решены с помощью TXL:
  - Преобразование языков
    - Java to Python Translator
    - Java to C# Translator
    - HTML to XHTML Converter
  - Преобразование форматов
    - HTML Pretty Printing of Source Code
  - Преобразование программ
    - Исправление «опасных» С-программ, имеющих переполнение буфера