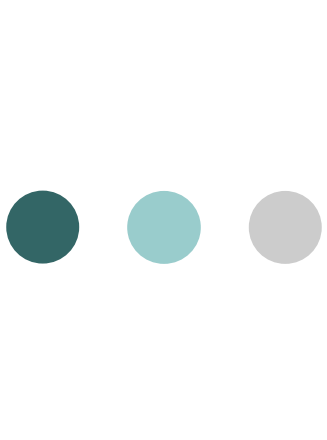


Reengineering

Доц. д-р. Ася Стоянва-Дойчева

Съдържание

- Наследени системи
- Поддръжка на софтуерните системи
- Reengineering
  - Reverse engineering
  - Forward engineering
- Цели на reverse engineering-а и на reengineering – а
- Процеса на reverse engineering
- Техники за reengineering
- Заключение

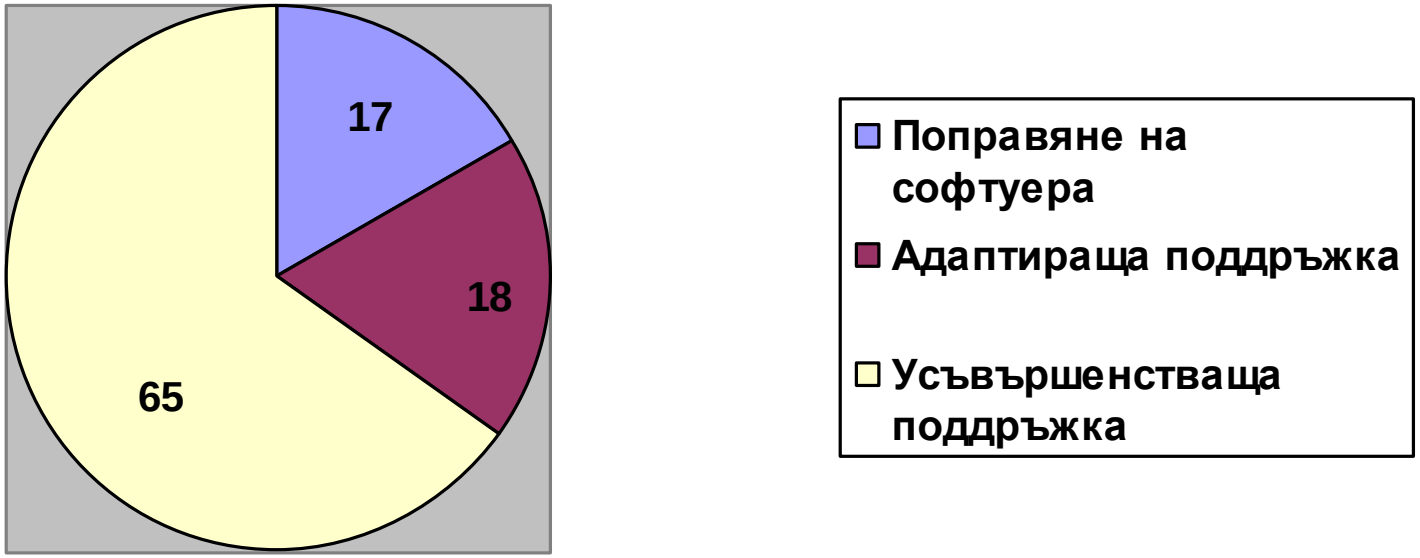
Наследени системи

- Legacy systems.
- Проблеми:
  - Разработчиците на системата ги няма (напусна ли са, пенсионирали са се и др.)
  - Използват се остарели методи
  - Правени са допълнителни кръпки и модификации
  - Липсва или има остаряла документация

Поддръжка на софтуера

- Модифициране на софтуерен продукт след поява на дефект
- Адаптация на продукта към нова среда
- Усъвършенстване на продукта – изпълнение на нови функционалности или нефункционални изисквания.

Поддръжка на софтуера



Reengineering

- **Def: Reengineering –** е изследване и промяна на системи за пресъздаването им в нова форма и последващо изпълнение на новата форма.

Цели на Reengineering

- Разделяне на монолитни системи на части, които могат да бъдат използвани отделно.
- Преминаване към друга платформа
- Подобряване на поддръжката, преносимостта и др.
- Използване на нови технологии – нови езици за програмиране, стандарти, библиотеки и др.

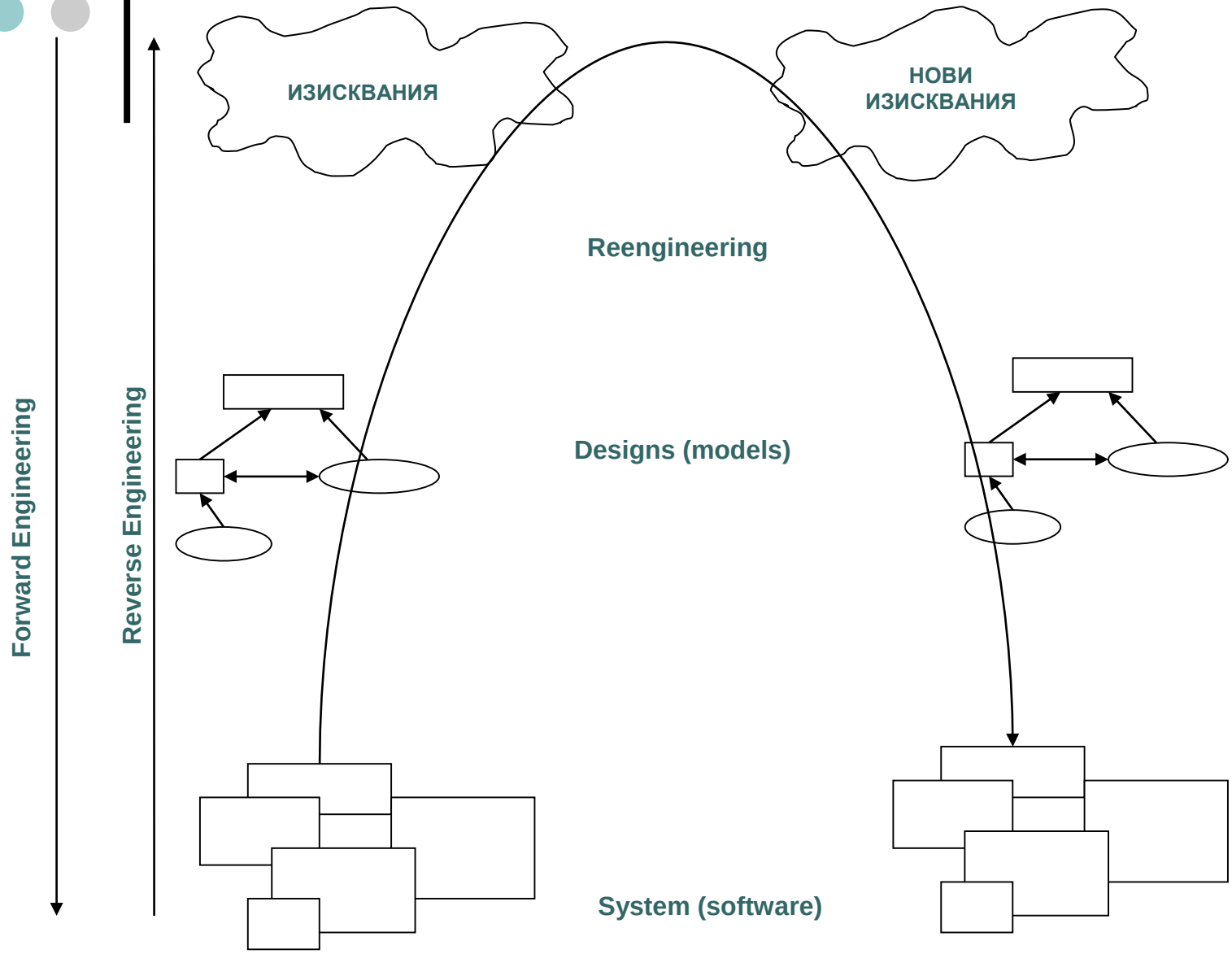
Forward Engineering

- **Def: Forward Engineering –** е традиционния процес на преминаване от високо абстрактно и логически ниво и изпълним независим проект към физическо изпълнение на системата.

Reverse Engineering

- **Def: Reverse Engineering** – е процес на анализиране на системи за:
  - Идентифициране на компонентите на системата и техните взаимовръзки
  - Представяне на системата в друга форма или в по-високо ниво на абстрактност

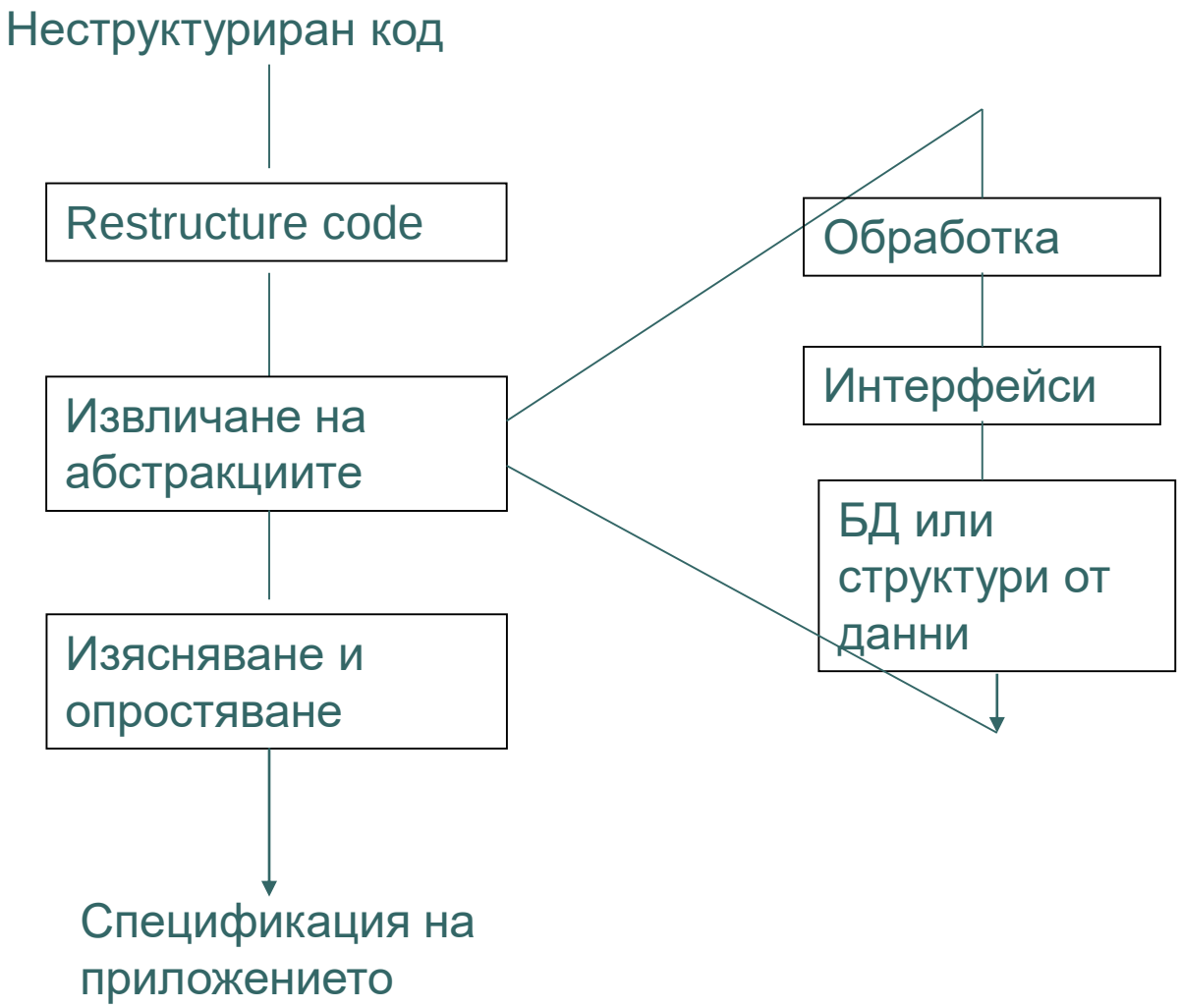
Reengineering



Цели на Reverse Engineering

- Справя се със сложността – на необходимите техники за разбиране на големи и сложни системи
- Генерира алтернативен изглед – автоматично генерира различни изгледи на системата
- Възстановява много информация – извлича какви промени са направени и защо
- Открива странични дефекти
- Идентифицира скрити абстракции в софтуера
- Улеснява повторното използване – определя кандидатите за повторно използване компоненти и артефакти

Процеса на reverse engineering



Техники за reengineering

- Restructuring – е трансформация от една представена форма към друга в същото ниво на абстракция, като се запазва външното поведение на системата.
  - Code Restructuring – кода се променя така, че да реализира същата функционалност, но с по-добро качество
  - Data Restructuring – цели се промяна на структурите от данни, за да се направи архитектурата на данните по-ефективна. Това може да означава например преминаване от един тип БД в друг.

Техники за reengineering

- Refactoring – това е restructuring в обектно-ориентиран контекст.
- Подходящи случаи за използване на refactoring са:
  - Неправилно прилагане на наследяване
  - Липса на наследяване
  - Неуместни операции
  - Нарушено капсулиране
  - Неправилна употреба на класове

Архитектурни проблеми

- Недостатъчна документация – повечето наследени системи страдат от несъществуваща или несъвместима документация
- Дублирана функционалност – cut, paste и edit се използват бързо и лесно, но това изключително много затруднява поддръжката
- Липса на модулиране – силното свързване между модулите затруднява еволюцията.
- Неправилно наследяване – липсата или неправилното наследяване затруднява преносимостта и адаптивността.

Заключение

- Винаги ще имаме legacy (наследени) системи, защото ценните системи надживяват първоначалните си изисквания
- Reverse Engineering техниките ни помагат да възстановим дизайна (проекта) от наследения софтуер.
- Reengineering техниките са необходими за реструктуриране на ценните наследени системи, така че да могат да изпълняват нови изисквания.