

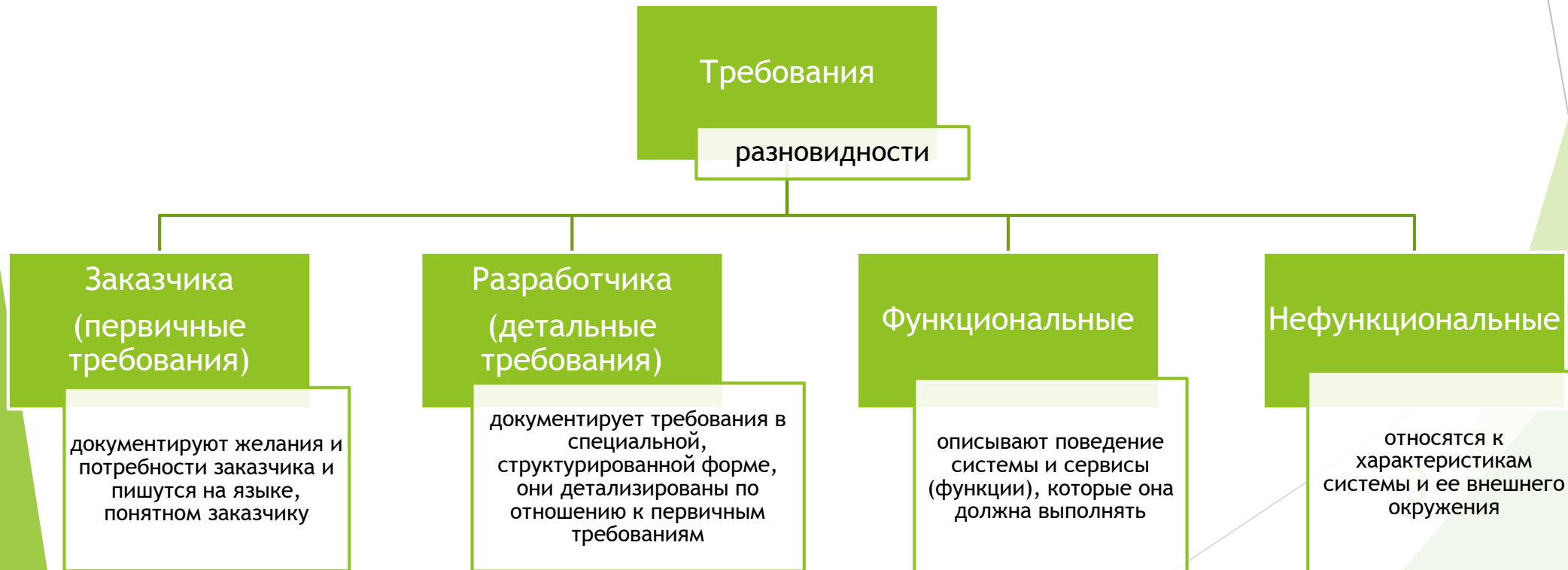
Программная инженерия. Формирование и анализ требований

Формирование и анализ требований

Виды требований к ПО

Требования (requirements) называют описание функциональных возможностей и ограничений, накладываемых на создаваемую программную систему.

Требования выражают *что* система должна делать, а не *как* добиться выполнения этих функций.



Формирование требований

Цель этой работы - сформировать требования заказчика.

Требования заказчика представляется в такой форме, что они понятны любому пользователю, однако приводит к проблемам:

- ▶ витиеватость стиля изложения;
- ▶ смешение и объединение требований.

Шаги процесса формирования требований:

- ❑ Определение представлений заказчика
- ❑ Проведение опроса представителей заказчика
- ❑ Документирование результатов опроса
- ❑ Проверка требований

Анализ требований

Формирование требований является лишь начальной фазой работы с требованиями.

Анализ требований рассматривает требования заказчика как исходные данные, на выходе анализа - требования разработчика, называемые детальными требованиями.

В ходе анализа ищется ответ на вопрос: «Что должна будущая система?»

Типичные шаги анализа требований:

- ❑ Организация первичных требований
 - ❑ по режиму
 - ❑ по категориям пользователей
 - ❑ по объектам
 - ❑ по свойствам
 - ❑ по стимулам
 - ❑ по откликам
 - ❑ по иерархии функций

Анализ требований

- ❑ Преобразование первичных требований в детальные требования
 - ❑ первичная оценка
 - ❑ обеспечение прослеживаемости требования
 - ❑ обеспечение тестируемости требования
 - ❑ анализ однозначности толкования требований
 - ❑ назначение приоритета требований
 - ❑ проверка полноты требования
 - ❑ проверка согласованности требования с другими требованиями
 - ❑ требование зависимости в спецификацию анализа
 - ❑ описание и обоснование желаемых характеристик детального требования

Анализ требований

- ❑ Аттестация детальных требований
 - ❑ проверка правильности требований
 - ❑ проверка на непротиворечивость
 - ❑ проверка на полноту
 - ❑ проверка на выполнимость

В ходе аттестации применяются следующие методы:

- ❑ Совместные проверки требований
- ❑ Макетирование
- ❑ Генерация тестов
- ❑ Автоматизированная проверка непротиворечивости



Спецификация требований

Спецификация требований - это документ, являющийся официальным предписанием для разработчиков ПО. Он содержит описание требований заказчика (первичные требования) и разработчика (детальные требования).

Многие организации разрабатывают стандарты документирования требований. Стандарт утверждает, что качественно составленная спецификация (SRS) приносит следующие выгоды:

- ❑ Создание основ для соглашения между заказчиками и разработчиками по поводу функций, которые должен выполнять программный продукт;
- ❑ Уменьшение объема работ по разработке;
- ❑ Обеспечение основы для валидации и верификации;
- ❑ Облегчение передачи пользователям;
- ❑ Служение в качестве основы для расширения.

Структура SRS

Стандарт предлагает следующую структуру спецификации требований к программному обеспечению (SRS):

- ❑ Введение
 - ❑ Назначение (назначение спецификации, аудитория)
 - ❑ Область действия (название ПО, его задачи, применение)
 - ❑ Определения и сокращения (терминология)
 - ❑ Краткий обзор (характеристика всех разделов)
- ❑ Полное описание
 - ❑ Перспектива изделия
 - ❑ Функции изделия
 - ❑ Характеристики пользователя
 - ❑ Ограничения
 - ❑ Допущения и зависимости
 - ❑ Распределение требований

Структура SRS

- ❑ Конкретные требования (охватывают функциональные, нефункциональные и интерфейсные требования)

Это наиболее значимая часть документа, описывает детальные требования, организованное выбранным способом.

- ❑ Приложения (оценка себестоимости, форма ввода-вывода, проблемы)
- ❑ Алфавитный указатель.

Управление требованиями

В ходе управления требованиями нужно решить ряд вопросов:

- ❑ Распознавание и учет требований;
- ❑ Управление внесенными изменениями;
- ❑ Стратегия трассировки;
- ❑ Хранение требований;
- ❑ Реализация цикла изменения требований;
- ❑ Управление трассировкой.

Шаги процесса управления изменениями:

- ❑ Распознавание проблемы;
- ❑ Анализ изменения;
- ❑ Выполнения изменения.

Классические методы анализа

Структурный анализ

Структурный анализ — один из формализованных методов анализа требований к ПО. Автор этого метода — Том Де Марко (1979). В этом методе программное изделие рассматривается как преобразователь информационного потока данных. Основной элемент структурного анализа — диаграмма потоков данных.

Диаграмма потоков данных (ПДД) — графическое средство для изображения информационного потока и преобразований, которым подвергаются данные при движении от входа к выходу системы.



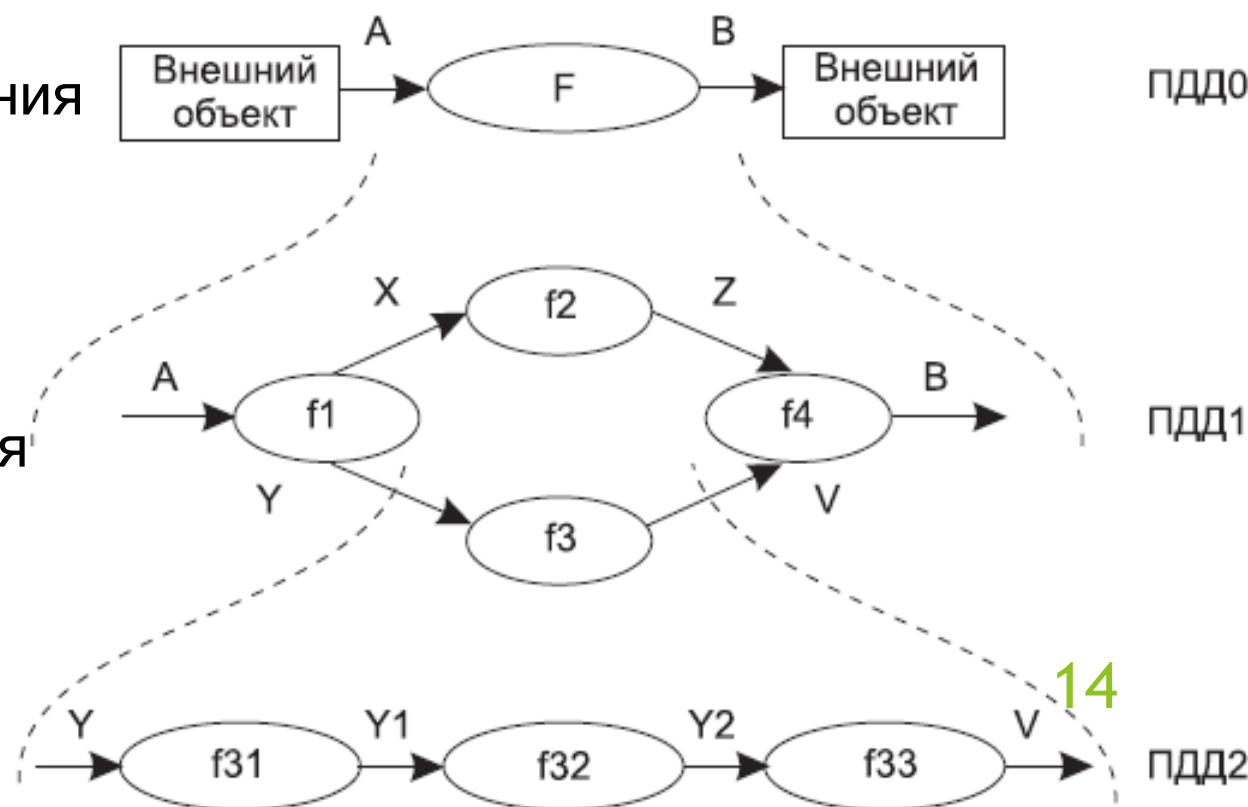
Система взаимосвязанных ПДД

Диаграмма может использоваться для представления программного изделия на любом уровне абстракции. *Диаграмма высшего (нулевого) уровня* представляет систему как единый овал со стрелкой, ее называют **основной** или **контекстной моделью**. Контекстная модель используется для указания внешних связей программного изделия.

Для детализации (уточнения системы) вводится диаграмма 1-го уровня.

Каждый из преобразователей этой диаграммы — подфункция общей системы.

Дальнейшее уточнение приводит к диаграмме 2-го уровня.



Методы анализа, ориентированные на структурных данных

Элементами предметной области для любой системы являются потоки, процессы и структуры данных. При структурном анализе активно работают только с потоками данных и процессами.

Методы, ориентированные на структуры данных, обеспечивают:

- ❑ Определение ключевых информационных объектов и операций.
- ❑ Определение иерархической структуры данных.
- ❑ Компоновку структур данных из типовых конструкций — последовательности, выбора, повторения.
- ❑ Последовательность шагов для превращения иерархической структуры данных в структуру программы.

Метод Варнье-Орра

В методе Варнье-Орра для представления структур применяют диаграммы Варнье. Для построения диаграмм Варнье используют *три базовых элемента: последовательность, выбор, повторение.*

Последовательность $\left\{ \begin{array}{l} a \\ b \\ c \end{array} \right.$

Выбор $\left\{ \begin{array}{l} a \\ \oplus \\ c \end{array} \right.$

Повторение $\left\{ \begin{array}{l} a(1, n), \text{ где } n - \text{ количество повторений} \end{array} \right.$

Метод Джексона

Как и метод Варнье-Орра, метод Джексона появился в период революции структурного программирования. Фактически оба метода решали одинаковую задачу: распространить базовые структуры программирования (последовательность, выбор, повторение) на всю область разработки сложных программных систем. Именно поэтому основные выразительные средства этих методов оказались так похожи друг на друга.

Метод Джексона (1975) включает 6 шагов. Три шага выполняются на этапе анализа, а остальные — на этапе проектирования.

- ❑ Объект-действие
- ❑ Объект-структура
- ❑ Начальное моделирование
- ❑ Доопределение функций
- ❑ Учет системного времени
- ❑ Реализация

СПАСИБО ЗА ВНИМАНИЕ!

**Если остались вопросы, задавайте их
в ЭИОС МТУСИ или пишите на почту**