

# Разработка и анализ требований проектирования ПО

## Анализ и моделирование требований к ПО

Лектор: Петрова О.Б.

СПбГУТ  
2017

# Атрибуты требований

- Полнота описания.
- Корректность.
- Непротиворечивость.
- Осуществимость.
- Необходимость (обоснованность).
- Приоритет.
- Однозначность толкования.
- Проверяемость.
- Статус.

# Полный набор требований ПО

- Вводы системы.
- Выводы системы.
- Функции системы.
- Атрибуты системы.
- Атрибуты системного окружения.



## Контекстная диаграмма, диаграмма Use Case

Не включаются:

- календарные планы,
- выделенные средства,
- тесты.

# Представление вводов и выводов ПО



Контекстная диаграмма



Диаграмма Use Case

# Полнота нефункциональных требований

Пример: время ответа для 90 процентов транзакций будет составлять менее 3 секунд.

# Связь высокоуровневых функций и функциональных требований

## Концепция:

8. Система «Абитуриент» должна предоставлять ежедневную сводку о результатах приема документов от абитуриентов.

## Функциональные требования:

SR8.1 Система получает данные о количестве принятых документов из базы данных текущего года на текущую дату.

SR8.2 Система использует для сравнения данные сводки на последний день приема из базы данных прошлого года.

SR8.3 Система рисует график в системе координат, ось x которого — ось времени, где каждая точка — очередная дата, ось y — количество поданных заявлений.

SR8.4 Диапазон значений по оси x определяется сроком приема документов (Бизнес-требование \_\_), диапазон значений по оси y — от 0 до максимального значения количества принятых документов за предыдущий год\*1,2.

SR8.5 Каждая точка графика представляет собой количество документов, принятых от начала приема документов до конкретной даты.

SR8.6 В одних осях строятся два графика: на текущий год от начала приема до текущей даты и предыдущий год на весь период приема документов.

# Дочерние требования

SR8.1 Система получает данные о количестве принятых документов из базы данных текущего года на текущую дату.

SR8.2 Система использует для сравнения данные сводки на последний день приема из базы данных прошлого года.

SR8.3 Система рисует график зависимости количества принятых документов за период от даты начала приема документов.

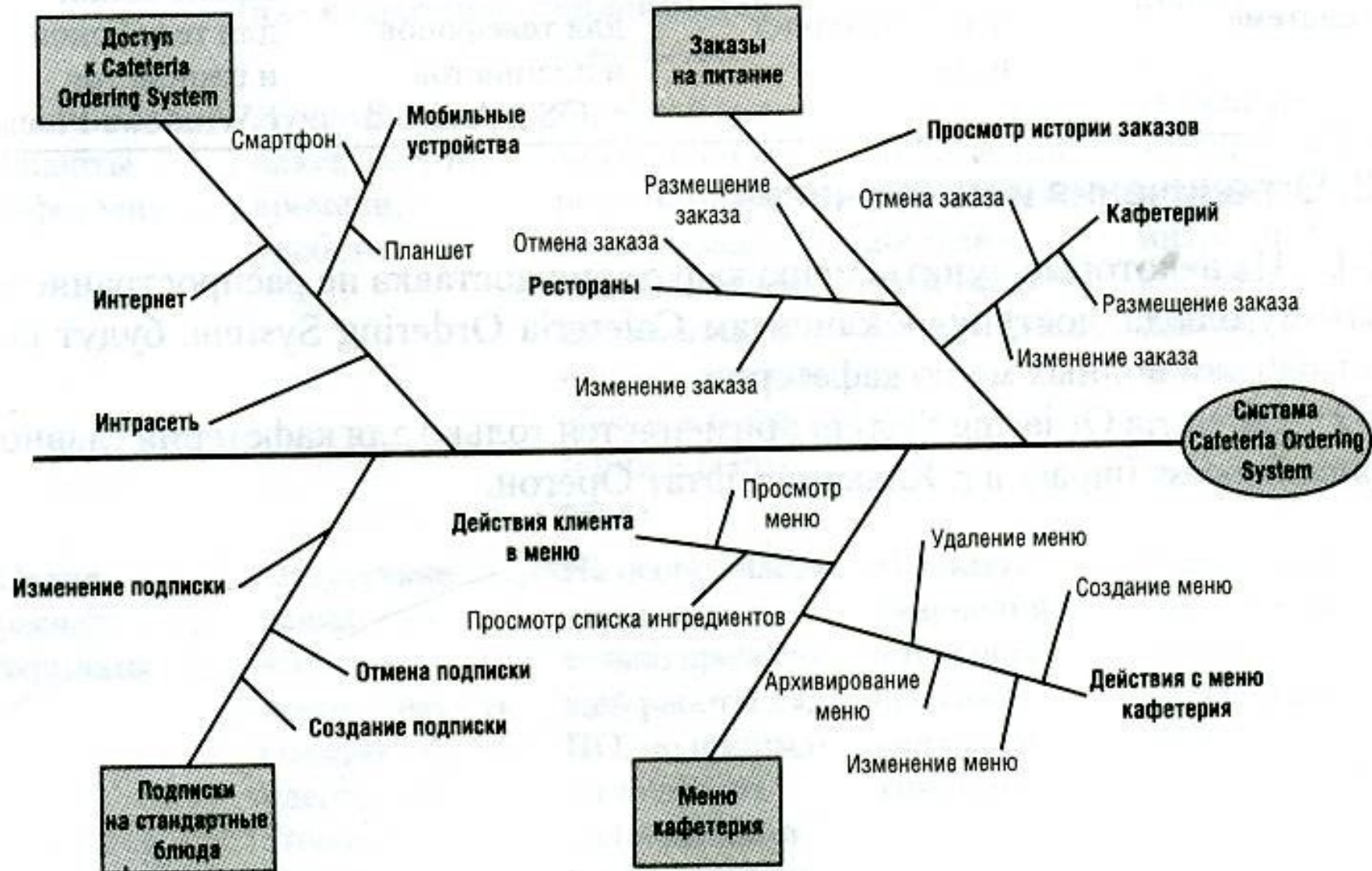
SR8.31 Ось x представляет собой ось времени, где каждая точка — очередная дата, ось y — количество поданных заявлений.

SR8.32 Диапазон значений по оси x определяется сроком приема документов (Бизнес-требование \_\_), диапазон значений по оси y — от 0 до максимального значения количества принятых документов за предыдущий год\*1,2.

SR8.33 Каждая точка графика представляет собой количество документов, принятых от начала приема документов до конкретной даты.

SR8.34 В одних осях строятся два графика: на текущий год от начала приема до текущей даты и предыдущий год на весь период приема документов.

# Дерево требований





# Обоснованность требований

Матрица зависимости требований:

Требование	T1	T2	T3	T4
T1	<u>X</u>	<u>X</u>	<u>X</u>	<u>X</u>
T2		<u>X</u>	<u>X</u>	<u>X</u>
T3	Конфликт		<u>X</u>	<u>X</u>
T4		Перекрытие	Перекрытие	<u>X</u>

# Корректность и непротиворечивость

Пример:

- все служащие старше 65 лет в конце календарного года должны получить премию в размере 10000 рублей;
- все служащие, имеющие стаж работы на предприятии 10 лет и более, в конце календарного года должны получить премию 5000 рублей.

Какую премию должен получить сотрудник 65 лет, имеющий стаж работы 15 лет?

# Проверяемость требований

Оцените возможность проверки следующих требований:

- система должна ответить на произвольный запрос в течение 500 миллисекунд;
- цифры на экране, показывающем время, должны хорошо выглядеть;
- система должна быть дружелюбной пользователю;
- система должна экспортировать данные для просмотра в формате, в котором в качестве разделителя используется запятая.

# Оценки разработчиков возможности проверки требований

- система должна отвечать на произвольный запрос в течение 500 миллисекунд:

все зависит от того, что подразумевается по словом произвольный, если число запросов конечно, и можно выявить наиболее сложные из них, проверка возможна;

- цифры на экране, показывающем время, должны хорошо выглядеть:  
красота — дело вкуса, проверить нельзя;

- система должна быть дружелюбной пользователю:

требование неконкретно, в таком виде непроверяемо;

- система должна экспортировать данные для просмотра в формате, в котором в качестве разделителя используется запятая:

надо уточнить некоторые детали, например, что будет, если данные представляют пустое множество, в этом случае требование проверяемо.

# Определение приоритетов

## Методика MoSCoW

Обязательные для выполнения (Must)

Должны быть сделаны (Should)

Могут быть сделаны (Could)

Не нужны (Won't)

## Методика определения по срочности и важности

---

Важные

Неважные

---

Срочные

Высокий приоритет

Не занимайтесь им!

Не срочные

Средний приоритет

Низкий приоритет

# Определение приоритета по полезности функции

Методика вычисления относительной полезности функции для пользователя Quality Function Deployment (QFD)

Оценки выгоды, урона, стоимости, риска — от 1(минимально) до 9(максимально)

Приоритет =

$\% \text{ ценности} / (\% \text{стоимости} * \text{вес\_стоимости} + \% \text{риска} * \text{вес\_риска})$

Относительный вес	1	1			1		1		
Функция	Выгода	Урон	Общая ценность	Ценность %	Стоимость	Стоимость %	Риск	Риск %	Приоритет
Дни рождения	7	7	14	21,21	4	16,67	4	13,79	0,70
Библиотека ссылок	5	3	8	12,12	3	12,50	4	13,79	0,46
Новости	9	9	18	27,27	2	8,33	3	10,34	1,46
Заказ билета	8	3	11	16,67	7	29,17	9	31,03	0,28
Доска почета	9	6	15	22,73	8	33,33	9	31,03	0,35
Итого	38	28	66	100	24	100	29	100	

# Статус требования

Предложено

Одобрено

Реализовано

Проверено

Удалено

Отклонено

# Вопросы

Оцените возможность проверки требования:

- система должна поддерживать до 1000 пользователей одновременно.