

# Инженерия требований к программному обеспечению

Алексей Островский

Физико-технический учебно-научный центр НАН Украины

24 октября 2014 г.

# Требования к ПО

## Определение

**Требования к ПО** — это:

- ▶ свойства системы, необходимые для выполнения предложенных заказчиком функций;
- ▶ ограничения на функционирование системы.

**Инженерия требований:**



## Классификация требований

- ▶ **Пользовательские требования** (англ. *user requirements*) — описание на естественном языке ожидаемой функциональности системы и присущих ей ограничений.  
**Источник:** предлагаются заказчиком ПО.  
**Инструменты:** естественный язык + диаграммы.
- ▶ **Системные требования** (англ. *system requirements*) — детальное описание функциональности системы и ограничений.  
**Источник:** результат совместной работы заказчика и разработчика.  
**Инструменты:** формальные языки, шаблоны, спецификации.

## Классификация требований — пример

**Пример.** Веб-сервис для вычисления чисел Фибоначчи

$$F_i = F_{i-1} + F_{i-2}.$$

**Пользовательское требование:** веб-сервис должен отображать ряд чисел Фибоначчи фрагментами по 100 чисел с помощью HTML-страниц.

**Системные требования:**

- ▶ Веб-сервис должен отображать числа Фибоначчи  $i, \dots, i + 99$  при доступе к веб-сервису с помощью URL вида `http://fib.example.com/fib/i`.
- ▶ Каждая сгенерированная страница должна содержать навигацию для доступа к следующим ста и (если применимо) к предыдущим ста числам Фибоначчи.
- ▶ При попытке доступа к сервису с помощью URL `http://fib.example.com/fib/str`, где *str* не является натуральным числом, должна выдаваться страница оговоренного вида с HTTP-кодом 400.

# Функциональные и нефункциональные требования

**Функциональные требования** — это:

- ▶ определение предоставляемых программным продуктом услуг;
- ▶ описание реакции на различные входные данные;
- ▶ описание поведения системы в различных ситуациях;
- ▶ (необязательно) спецификация запретов.

**Нефункциональные требования** — ограничения на функции, предоставляемые ПП:

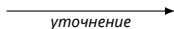
- ▶ временные ограничения;
- ▶ ограничения на процесс разработки;
- ▶ ограничения, связанные со стандартами разработки ПО.

## Связь между требованиями

### Нефункциональное требование

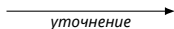
### Функциональное требование

Защита конфиденциальных  
данных



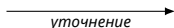
Система авторизации

Ограничение на  
занимаемую память



Периодическое удаление  
лишних данных

Отказоустойчивость



Система резервных  
копий данных

Нефункциональные требования могут в процессе уточнения порождать новые функциональные требования.

# Нефункциональные требования



# Нефункциональные требования





# Нефункциональные требования



# Нефункциональные требования



# Нефункциональные требования



## (Не)функциональные требования — пример

**Пример.** Веб-сервис для вычисления чисел Фибоначчи.

### **Функциональные требования:**

- ▶ требования к отображению информации ([см. выше](#));
- ▶ интерфейс администратора;
- ▶ учет количества посетителей.

### **Нефункциональные требования:**

- ▶ (*usability*) использование адаптивного дизайна для ПК, планшетов и смартфонов;
- ▶ (*производительность*) генерация любой страницы за  $\leq 0,5$  с;
- ▶ (*среда выполнения*) Linux, MySQL, Apache HTTP Server;
- ▶ (*разработка*) использование Python/Django.

# Спецификация требований

## Определение

**Спецификация требований** — запись требований в виде, обеспечивающем их ясность, однозначность, простоту понимания, полноту и непротиворечивость.

<b>Роль</b>	<b>Использование требований</b>
потребители	спецификация и уточнение требований
менеджеры	оценка затрат на систему; планирование процесса разработки
разработчики	детализация характеристик системы
тестеры	разработка тестов для проверки системы
отдел сопровождения	понимание системы и взаимоотношений между ее частями

## Документ спецификации

1. Предварительные замечания (версия документа, основания для ее создания).
2. Вступление (общее назначение системы, ее взаимодействие с другим ПО).
3. Словарь технических терминов.
4. Описание пользовательских требований (+ нефункциональные системные требования).
5. Архитектура системы.
6. Описание системных требований.
7. Системные модели (взаимодействие между компонентами, со средой выполнения и т. п.).
8. Эволюция системы (ожидаемые изменения системы).
9. Приложения.

# Запись требований

## Способы записи:

- ▶ естественный язык  
(подходит для записи пользовательских требований);
- ▶ структурированный язык (таблицы или шаблоны)  
(подходит для спецификации системных требований);
- ▶ язык описания архитектуры  
(используется редко, в основном для спецификации интерфейсов);
- ▶ графическая нотация (напр., UML-диаграммы)  
(подходит для детализации системных требований);
- ▶ математическая спецификация (напр., конечные автоматы)  
(используется для критических требований в области безопасности).

## Запись требований — пример

### Числа Фибоначчи — математический модуль

**Функция:** вычисление ряда чисел Фибоначчи.

**Описание:** вычисляет значение ста последовательных чисел Фибоначчи.

**Вход:**  $i$  — индекс первого числа Фибоначчи, которое нужно вычислить.

**Источник данных:** HTTP-запрос пользователя.

**Выход:** значения чисел Фибоначчи  $F_i, F_{i+1}, \dots, F_{i+99}$ .

**Назначение данных:** цикл обработки HTTP-запроса.

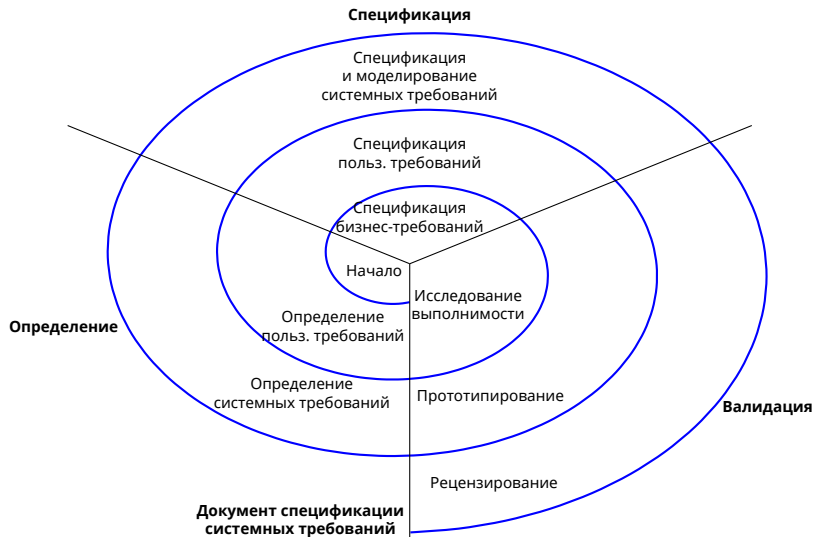
**Действие:** Числа  $F_i$  и  $F_{i+1}$  вычисляются по [формуле](#) быстрого возведения в степень. Оставшиеся числа вычисляются согласно определению чисел Фибоначчи.

**Требования:**  $i$  должно быть целым неотрицательным числом.

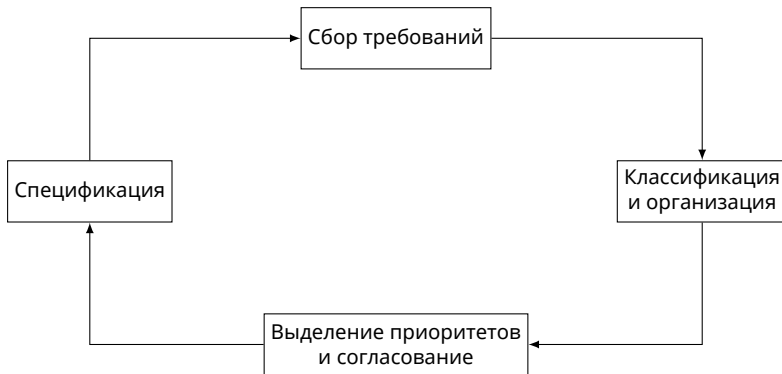
**Побочные эффекты:** нет.



# Процесс инженерии требований

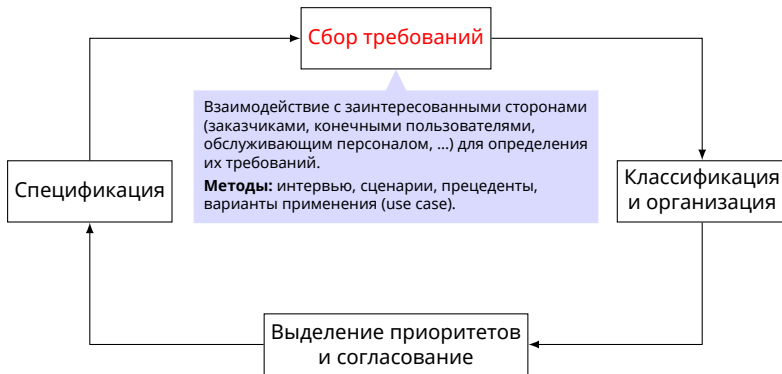


# Определение и анализ требований



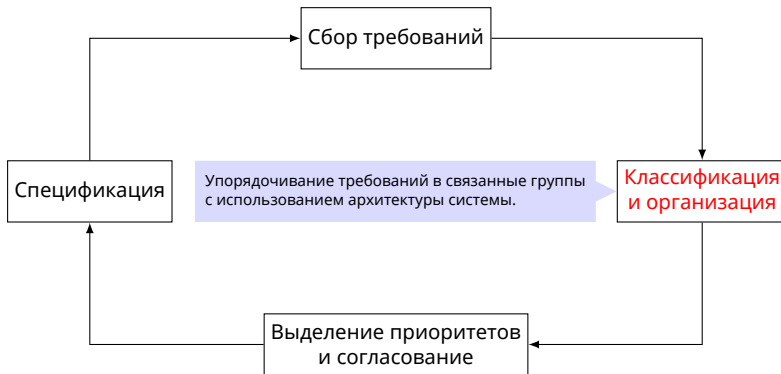
Процессы определения и анализа требований

# Определение и анализ требований



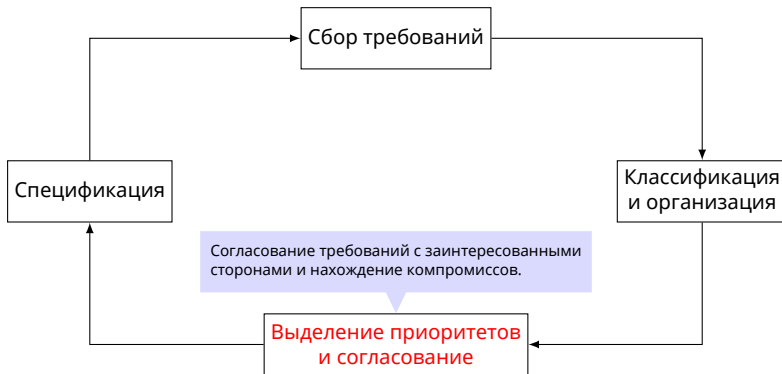
Процессы определения и анализа требований

# Определение и анализ требований



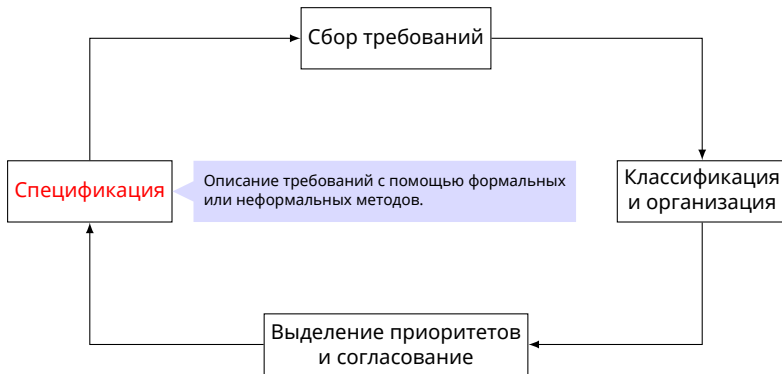
Процессы определения и анализа требований

# Определение и анализ требований



Процессы определения и анализа требований

# Определение и анализ требований



Процессы определения и анализа требований

# Валидация требований

## Проверки:

- ▶ корректность (согласованы ли требования со всеми заинтересованными сторонами?);
- ▶ непротиворечивость (есть ли конфликты между требованиями?);
- ▶ полнота (описывают ли требования все функции системы?);
- ▶ реалистичность (возможно ли реализовать требования?);
- ▶ верифицируемость (существуют ли тесты, проверяющие выполнение требований?).

## Методы валидации:

- ▶ рецензирование;
- ▶ прототипирование;
- ▶ создание тестов.

# Управление требованиями

## Определение

**Управление требованиями** — процесс выявления и контроля изменений в системных требованиях.

### Причины изменений:

- ▶ изменение среды выполнения (новое оборудование, новые приоритеты, изменение регламентирующих документов или законодательства, ...);
- ▶ различие между пониманием системы заказчиком и конечными пользователями;
- ▶ изменение баланса между различными группами пользователей.



# Процесс изменения требований

## Традиционная модель ЖЦ:

1. Анализ проблемы и спецификация изменения. Анализ пересылается заказчику изменения для внесения дополнений или отказа от изменения.
2. Анализ изменения и оценка затрат.
3. Имплементация изменений в общую спецификацию требований, а также в архитектуру и имплементацию системы.

## Agile development:

1. Оценка приоритета изменения.
2. Модификация плана следующего цикла разработки.

## Выводы

1. Требования к ПО определяют его возможности (функциональные тр.) и ограничения на процесс разработки (нефункциональные тр.).
2. Процесс инженерии требований включает в себя анализ выполнимости, выработку и анализ требований, их спецификацию, проверку, а также управление требованиями.
3. Существует несколько инструментов спецификации требований, в частности формальные языки и диаграммы UML.

# Матеріали



Лавріщева К. М.

Програмна інженерія (підручник).

К., 2008. — 319 с.



Sommerville, Ian

Software Engineering.

Pearson, 2011. — 790 p.

Спасибо за внимание!

## Приложение. Вычисление чисел Фибоначчи

$$F_n = F_{n-1} + F_{n-2}, \quad F_0 = 0, F_1 = 1.$$

$F_n, n \geq 1$  можно вычислить за время  $O(\log n)$  с помощью формулы

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

где возведение матрицы в степень выполняется с помощью [быстрого алгоритма](#).

Для вычислений необходима поддержка целых чисел с произвольной разрядностью.

Другой способ — использование формулы

$$F_n = \left[ \frac{1}{\sqrt{5}} \left( \frac{1 + \sqrt{5}}{2} \right)^n \right]$$

(необходима поддержка вещественных чисел с произвольной разрядностью).