



# МЕТОДЫ И СРЕДСТВА ИСИТ 2024



# ЛЕКЦИЯ 2

Ключевые определения, документация SRS..



# СОДЕРЖАНИЕ ЛЕКЦИИ

- ◆ Определение терминов технология проектирования и методы (методология) проектирования
- ◆ Определение спецификации
- ◆ Типы спецификаций
- ◆ Ключевая спецификация SRS и ее компоненты

# ВВОДНЫЕ ОПРЕДЕЛЕНИЯ КУРСА

- ◆ Технология - совокупность методов и инструментов для достижения желаемого результата; в широком смысле - применение научного знания для решения практических задач.
- ◆ Методология - система наиболее общих принципов, положений и методов, составляющий основу для проектирования информационных систем и технологий.

# ВВОДНЫЕ ОПРЕДЕЛЕНИЯ КУРСА

- ◆ Информационная система (ИС) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию (ISO/IEC 2382:2015).
- ◆ Информационные технологии (ИТ) — ресурсы, необходимые для сбора, обработки, хранения и распространения информации (ISO/IEC 38500:2015).

# ТРЕБОВАНИЯ К ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ

- ◆ технология должна поддерживать полный ЖЦ ПО;
- ◆ технология должна обеспечивать гарантированное достижение целей разработки ИС;
- ◆ технология должна обеспечивать возможность выполнения крупных проектов в виде подсистем;
- ◆ технология должна обеспечивать возможность ведения работ по проектированию отдельных подсистем небольшими группами;
- ◆ технология должна обеспечивать минимальное время получения работоспособной ИС.

# ТРЕБОВАНИЯ К ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ

- ◆ технология должна предусматривать возможность управления конфигурацией проекта;
- ◆ технология должна обеспечивать независимость выполняемых проектных решений от средств реализации ИС;
- ◆ технология должна быть поддержана комплексом согласованных CASE-средств.

# ЗАДАЧИ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ ИС и ИТ

- ◆ обеспечивать создание ИС и ИТ любой сложности и масштаба, отвечающих целям и задачам объекта автоматизации;
- ◆ гарантировать создание системы с заданным качеством в заданные сроки и в рамках установленного бюджета проекта;
- ◆ поддерживать удобную дисциплину сопровождения, модификации и наращивания системы;
- ◆ обеспечивать создание ИС и ИТ, отвечающих требованиям открытости, переносимости и масштабируемости;
- ◆ обеспечивать преемственность разработки.



# КОМПОНЕНТЫ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ ИС и ИТ

- ◆ набор моделей (типов моделей) для описания требований к ИС, проектных и программных решений;
- ◆ метод применения набора моделей для построения ИС;
- ◆ процесс организации проектных работ.

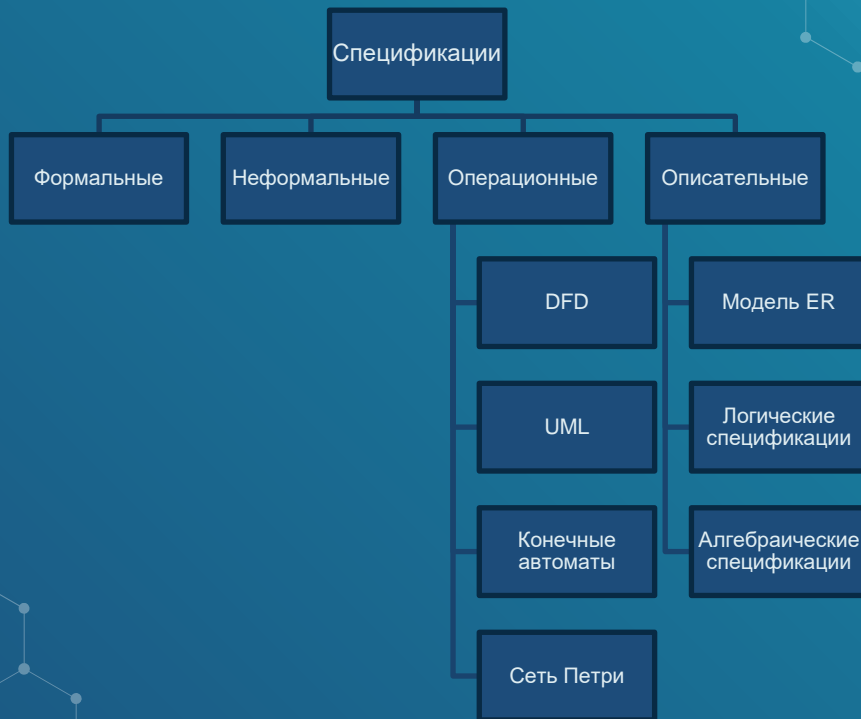
# СРЕДСТВА СПЕЦИФИКАЦИИ

*Спецификация* — документ, определяющий в полной, точной, проверяемой форме требования, дизайн, поведение или другие свойства системы либо компонента, и, зачастую, процедуры, для определения, выполняются ли эти требования.

*Основные свойства спецификации:*

- ◆ точность, однозначность.
- ◆ понятность, ясность, читабельность.
- ◆ полнота описания задачи.

# КЛАССИФИКАЦИЯ СПЕЦИФИКАЦИИ



# СПЕЦИФИКАЦИИ SRS

*Software Requirement Specification (спецификации требований к ПО)* – структурированный набор требований (функциональность, производительность, конструктивные ограничения и атрибуты) к программному обеспечению и его внешним интерфейсам (IEEE 1012:2004). Предназначен для того, чтобы установить базу для соглашения между заказчиком и разработчиком (или подрядчиками) о том, как должен функционировать программный продукт.

# СТРУКТУРА SRS. ВВЕДЕНИЕ.

**Цели.** В данной секции дается общее описание ИС или ИТ, функционал которой будет описываться в SRS.

**Соглашения о терминах.** Описанию подлежат все непонятные технические слова или термины которые встречаются в SRS. Описание непонятого слова не может содержать другое непонятное слово. Предполагаемая аудитория.

**Масштаб проекта.**

**Ссылки на источники.** Ссылки на литературу в которой можно найти основания использованных технологий и фактов.

# СТРУКТУРА SRS. ОБЩЕЕ ОПИСАНИЕ.

***Видение продукта.***

***Функциональность продукта.*** Описываются части функционала на высоком уровне. Тут желательно разместить DFD-диаграмму которая покажет общее взаимодействие системы.

***Классы и характеристики пользователей.***

***Среда функционирования продукта (операционная среда).*** Описывается окружение в котором будет работать продукт. ОС, версии компиляторов, базы данных, сервера, софт, железо и другие вещи которые необходимы для работы ПО или ИТ.

***Рамки, ограничения, правила и стандарты.***

- ◇ стандарты обмена данными
- ◇ ограничения накладываемые операционной средой
- ◇ ограничения которые могут быть наложены бизнес-логикой проекта

***Документация пользователя.***

# СТРУКТУРА SRS. ФУНКЦИОНАЛЬНОСТЬ СИСТЕМЫ.

**Функциональный блок X.** Для каждой функции проекта создается уникальное название и идентификатор. Например, `server.html.editor`.

- ◇ *описание и приоритет*, функция продукта описывается детально. Для чего она? Что должна делать? Какой у нее приоритет выполнения
- ◇ *причинно-следственные связи*, триггер запуска функции продукта. Когда она запускается и как себя ведет при запуске? Например, HTML редактор показывается при нажатии пользователя на ссылку меню «Открыть HTML редактор».
- ◇ *функциональные требования*, подробное и детальное описание функции. Описываем все: как работает, как реагирует на ошибки, что должно проверять, как отображать данные, как и куда что сохраняет и т.д.

# СТРУКТУРА SRS. ТРЕБОВАНИЯ К ВНЕШНИМ ИНТЕРФЕЙСАМ.

Описание того как система будет взаимодействовать с внешним миром. Какие API входят в состав системы, как получить те или иные данные и т.п.

Подразделы служат для детализации требований:

- ◇ *интерфейсы пользователя*
- ◇ *программные интерфейсы*
- ◇ *интерфейсы оборудования*
- ◇ *интерфейсы связи и коммуникации*



# СТРУКТУРА SRS. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.

**Требования к производительности.** Требования к производительности. Допустим база данных проекта должна выдерживать 1000 запросов в секунду и т.п.

**Требования к сохранности (данных).**

**Требования к качеству программного обеспечения.** Требования к качеству кода. Какие тесты использовать? Какие метрики использовать для определения качества кода? Сколько кода должно быть покрыто тестами?

**Требования к безопасности системы.** Требования по безопасности. Если это HTML редактор, через которые можно изменять что-то на сайте, то это может быть нечто вроде: через HTML редактор не должно быть возможности поставить shell на сервер и т.п.

**Требования по интеллектуальной собственности.**

# САМОСТОЯТЕЛЬНОЕ ЗАДАНИЕ (ТЕСТЫ)

- ◆ Выполнить Тест 1. Дедлайн – 23 февраля.
- ◆ Google Classroom для групп БСБО-04,05,06,07,08,09: enkfxui.
- ◆ Выполнить Тест 1. Дедлайн – 26 февраля.
- ◆ Google Classroom для групп БСБО-10,11,12: wdqcim2.
- ◆ Google Classroom для групп БСБО-1,2,3,13,14: zrhmuks.
- ◆ Обратите внимание на то, чтобы в Google Classroom ваш псевдоним содержал вашу фамилию кириллицей (в крайнем случае - латиницей).

# САМОСТОЯТЕЛЬНОЕ ЗАДАНИЕ (КНИГИ)

- ◆ По учебнику Технологии разработки ПО: Глава 4 Управление требованиями, стр. 104-121.  
<http://msuniversity.ru/d/16>
- ◆ Образец спецификаций на примере Технического задания на разработку ПО, <http://msuniversity.ru/d/16>
- ◆ Образец спецификации SRS, <http://msuniversity.ru/d/16>

# СПАСИБО!

ВАШИ ВОПРОСЫ,  
ПОЖАЛУЙСТА?

