

# ОБЗОР ТЕХНОЛОГИЙ ПРОЕКТИРОВАНИЯ ПРИЛОЖЕНИЙ ИНФОРМАЦИОННЫХ СИСТЕМ

Подготовил: Исак Баглан

IT:MDS-19.659



# ЧТО ТАКОЕ СИСТЕМА?

- **Система** - сущность, которая в результате взаимодействия ее частей (компонентов) может поддерживать свое существование и функционировать как единое целое. В этом определении важно отметить, что поведение системы зависит не от природы свойств ее образующих частей, а от того, как эти части соединены между собой.
- Из определения системы следует, что системы функционируют как целое, что порождает у них свойства, отличающиеся от свойств составляющих ее частей. Эти свойства известны как *эмерджентные (возникающие)*. Разделив систему на компоненты, мы никогда не обнаружим ее существенных свойств, они проявляются только в результате действия целостной системы. Эмерджентные свойства являются важными в понимании сущности анализа и синтеза. Разбиение системы на составляющие части для познания принципов ее функционирования называется *анализом*. С помощью анализа мы получаем знания, однако теряем возможность понимания свойств. Дополнением анализа является *синтез* — создание целого из частей. С помощью синтеза мы обретаем *понимание*.



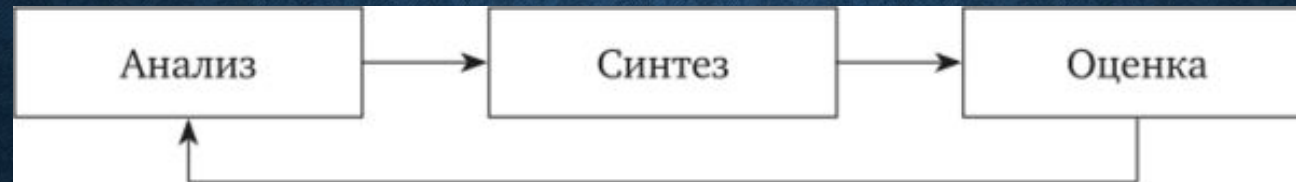
# СИСТЕМЫ БЫВАЮТ ...

Принято выделять *простые* и *сложные системы*. Сложность может проявляться двумя различными путями: статическим и динамическим. *Статическая сложность* (детальная) определяется в процессе детализации как количество рассматриваемых элементов. *Динамическая сложность* зависит от отношений между элементами. Информационные системы относятся к классу динамически сложных. Таким образом, информационная система состоит из множества элементов или подсистем, которые находятся в разных состояниях и могут изменяться в зависимости от изменения других частей.



# ПРОЕКТИРОВАНИЕ — ПРОЦЕСС, КОТОРЫЙ ДАЕТ НАЧАЛО ИЗМЕНЕНИЯМ В ИСКУССТВЕННОЙ СРЕДЕ.

Проектировщик должен предвидеть конечный результат осуществления своего проекта и определить меры, необходимые для достижения этого результата. Проектирование представляется как итерационный цикл. Каждая итерация отличается большей детализацией и меньшей общностью.



В процессе проектирования выделяют следующие фазы:

- 1) *дивергенция* — расширение границ проектной ситуации с целью обеспечения более обширного пространства поиска решения;
- 2) *трансформация* — стадия создания принципов и концепций (исследование структуры проблемы);
- 3) *конвергенция* — охватывает традиционное проектирование (кодирование, отладка, проработка деталей).



# ТРУДНОСТИ ПРОЕКТИРОВАНИЯ:

- • предположение о конечном результате проектирования приходится делать еще до того, как исследованы средства его достижения;
- • часто случается, что в ходе исследования событий в обратном порядке (от конечного результата) обнаруживаются непредвиденные трудности или открываются новые, более благоприятные возможности.



# *ОСНОВНЫМИ ЗАДАЧАМИ МЕТОДОЛОГИИ ЯВЛЯЮТСЯ СЛЕДУЮЩИЕ:*

- • обеспечение создания ИС, отвечающих предъявляемым к ним требованиям;
- • создание системы с заданным качеством в заданные сроки и в рамках бюджета;
- • поддержка сопровождения, модификации и наращивания ИС;
- • создание ИС, отвечающих требованиям открытости, переносимости и масштабируемости.



# КЛАССИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

- Классическое проектирование ИС берет свое начало в 1970-х гг. Одно из первых направлений получило название *каскадной схемы* проектирования. Она широко использовалась при проектировании ИС и включала следующие стадии проекта: разработка требований, проектирование, реализация, тестирование, ввод в действие (внедрение). Основной особенностью данной методики является последовательная организация работ при разбиении структуры ИС на заранее определенный ряд подсистем: организационное, методическое, информационное, программное и аппаратное обеспечения. В западной литературе такая схема организации работ получила название «водопадной модели» (*waterfall model*) и включала дополнительно итерационные процедуры уточнения требований к системе и рассмотрения вариантов проектных решений. Каскадная модель предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.



# КЛАССИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Основными недостатками каскадной схемы проектирования являются запаздывание получения конечных результатов и низкая эффективность.





# ИТЕРАЦИОННАЯ МОДЕЛЬ

- Дальнейшим развитием каскадной модели явилась *итерационная (поэтапная модель)* с промежуточным контролем. Разработка ИС ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах. Время жизни каждого из этапов растягивается на весь период разработки.



# ИТЕРАЦИОННАЯ ИЛИ ПОЭТАПНАЯ МОДЕЛЬ:



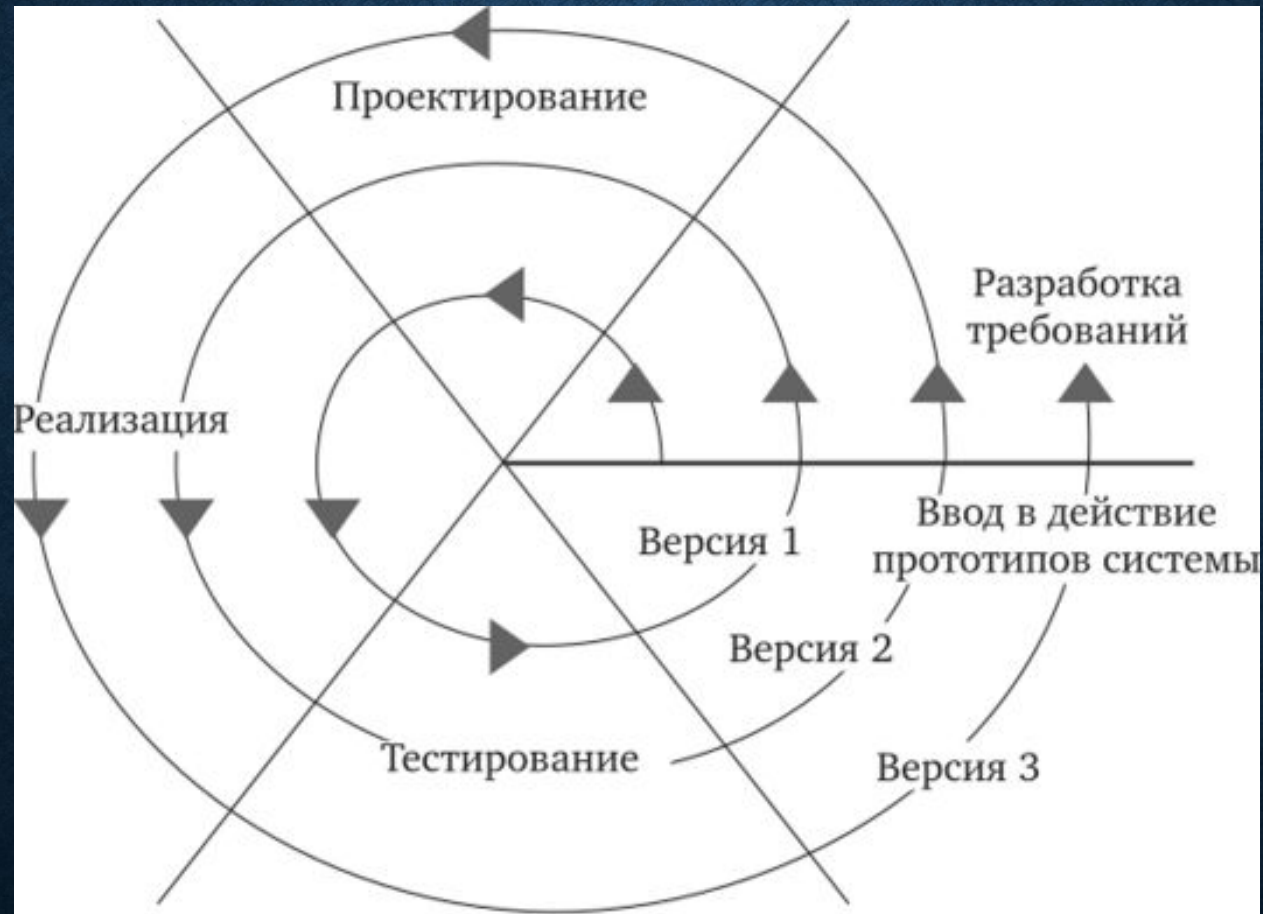


# СПИРАЛЬНЫЙ МЕТОД РАЗРАБОТКИ

- В процессе совершенствования разработки ИС появилась *схема непрерывной (спиральной) разработки*, использовавшаяся при реализации больших проектов фирмы *IBM* в 1970—1980-х гг.
- Характерной особенностью данной методики стал непрерывный спиральный процесс разработки ИС с планируемыми точками передачи в эксплуатацию новых версий и новых функциональных подсистем. На каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки — анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования).



# СПИРАЛЬНЫЙ МЕТОД





# ПО СТЕПЕНИ АВТОМАТИЗАЦИИ МЕТОДЫ ПРОЕКТИРОВАНИЯ РАЗДЕЛЯЮТСЯ:

- на *ручное*, при котором проектирование компонентов ИС осуществляется без использования специальных инструментальных программных средств, а программирование — на алгоритмических языках;
- • *компьютерное*, при котором производится генерация или конфигурирование (настройка) проектных решений на основе использования специальных инструментальных программных средств.



# ОСНОВНЫЕ КЛАССЫ ПРОЕКТИРОВАНИЯ

- Каноническая: В основе *канонического проектирования* лежит каскадная модель жизненного цикла ИС.
- Индустриальная: *автоматизированное* (использование CASE-технологий) и *типовое* (параметрически-ориентированное или модельно-ориентированное) проектирование.



# ВЫБОР ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ОСУЩЕСТВЛЯЕТСЯ С УЧЕТОМ СЛЕДУЮЩИХ ТРЕБОВАНИЙ:

1. Возможность обеспечения соответствия создаваемого с помощью конкретной технологии проекта требованиям заказчика; (Требования заказчика)
2. Способность выбираемой технологии обеспечивать минимальные трудовые и стоимостные затраты на проектирование и сопровождение проекта; (Затраты финансовые и трудовые)
3. Создание условий для повышения производительности труда проектировщика; (Производительность труда)
4. Обеспечение надежности процесса проектирования и эксплуатации проекта; (Надежность и безопасность)
5. Простота ведения проектной документации. (Простота)



# **CASE-ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ**

CASE – это акроним Computer-Aided System / Software Engineering, объединяющий методы автоматизации самого процесса проектирования информационных систем. CASE средства поддерживают или автоматизируют работы на всех этапах жизненного цикла системы.

Целью CASE-технологии является ускорение процесса создания систем и повышение их качества.

CASE-средства не обязательно дают немедленный эффект, он может быть получен только спустя какое-то время.

*Программа, как конечный продукт проектирования, содержит не только идеи программиста, но также и идеи системного аналитика и требования пользователей.*



## Архитектура CASE-средства:



**Репозиторий** (энциклопедия, словарь данных) является ядром системы и представляет собой специализированную базу данных, на основе которой выполняется синхронизация всех описаний (диаграмм) и координация усилий всех разработчиков. Репозиторий содержит информацию об объектах проектируемой ИС и взаимосвязях между ними.

**Графический редактор** диаграмм позволяет создавать различные описания системы в виде рисунков.

**Верификатор диаграмм** контролирует правильность диаграмм и синхронизирует диаграммы между собой.

**Документатор проекта** документирует проект в виде различных отчетов. Отчеты могут строиться по нескольким признакам, например, по времени, автору, элементам диаграмм, диаграмме или проекту в целом.

**Администратор проекта** автоматизирует выполнение инициализации проекта и мониторинг выполнения проекта.



# ***RAD-ТЕХНОЛОГИИ ПРОТОТИПНОГО СОЗДАНИЯ ПРИЛОЖЕНИЙ.***

***Rapid Application Development (RAD)*** – методы быстрой разработки программ.

Методология RAD не применима:

- 1) Для применения сложных расчетных программ, ОС или др. систем, требующих наличия большого объема уникального кода.
- 2) Для построения приложений в которых отсутствует ярко выраженная интерфейсная часть.
- 3) Для построения приложения, от которых зависит безопасность людей.

***Принципы методологии RAD:***

- 1) Разработка приложений итерациями;
- 2) Необязательность полного завершения работ на каждом из этапов ЖЦ;
- 3) Обязательное вовлечение пользователей в процесс разработки ИС;
- 4) Необходимое применение CASE-средства
- 5) Применение средств управления конфигурацией, которое облегчает внесение изменений в проект и сопровождение готовой системы;
- 6) Необходимое использование генераторов кода;
- 7) Использование прототипов, позволяющих полнее выяснить и удовлетворить потребности конечного пользователя;
- 8) Тестирование и разработка проекта, осуществляемые одновременно с разработкой;
- 9) Ведение разработкой немногочисленной хорошо управляемой командой профессионалов;
- 10) Грамотное руководство разрабатываемой системой, четкое планирование и контроль выполнения работ.



• БЛАГОДАРЮ ЗА  
ВНИМАНИЕ!!!