

Разработка ПО.

**Модели
жизненного цикла ПО.**

Методологии разработки ПО

- **Разработка ПО** — процесс, направленный на создание и поддержание работоспособности, качества и надежности ПО

Проблемы:

- новизна решений,
новизна платформ и инструментов
- высокая сложность систем
- ...

-
- Менее предсказуем результат
 - Хуже поддается планированию
 - До сих пор в большей степени творчество, чем ремесло

Процессы организации разработки ПО

- **текущий процесс** - осуществляется некоторой командой в рамках определенного проекта, имеет большое количество особенностей и индивидуальностей
- **конкретный процесс** – спланированный в определенном проекте процесс разработки (роли и обязанности в команде, рабочие продукты, порядок участия в их разработке членов команды, ...)
- **стандартный процесс** – единые правила, применяемые в компании по разработке ПО для организации текущих процессов
- **универсальный процесс -?**

Идеальный процесс разработки ПО

НЕ СУЩЕСТВУЕТ!!!!

На сегодняшний день не существует **универсального процесса** разработки ПО – набора методик, правил и предписаний, подходящих для ПО любого вида, для любых компаний, для команд любого состава, компетенций, национальности,...

Исследование особенностей процессов разработки ПО на основе моделей жизненного цикла

- **Жизненный цикл (ЖЦ) ПО** — период времени, с момента принятия решения о создании ПО и до момента его изъятия из эксплуатации

за это время реализуется

множество различных процессов, видов деятельности, методов, методик и шагов

- **Фаза (стадия) ЖЦ ПО** — часть процесса создания ПО, ограниченная определенными временными рамками и заканчивающаяся выпуском продукта этой фазы (моделей, программных компонентов, документации)



Вид деятельности — это определенный тип работы, выполняемый в процессе разработки ПО

Стандарт ISO 12207

Systems and software engineering — Software life cycle processes

Содержание стандарта:

- ▶ 23 процесса разработки;
- ▶ 95 родов деятельности по разработке (англ. *activity*);
- ▶ 325 заданий (англ. *task*);
- ▶ 224 результатов выполнения процессов (англ. *outcome*).

NB. Стандарт определяет *составляющие* процессов разработки ПО, но не *последовательность* их выполнения.

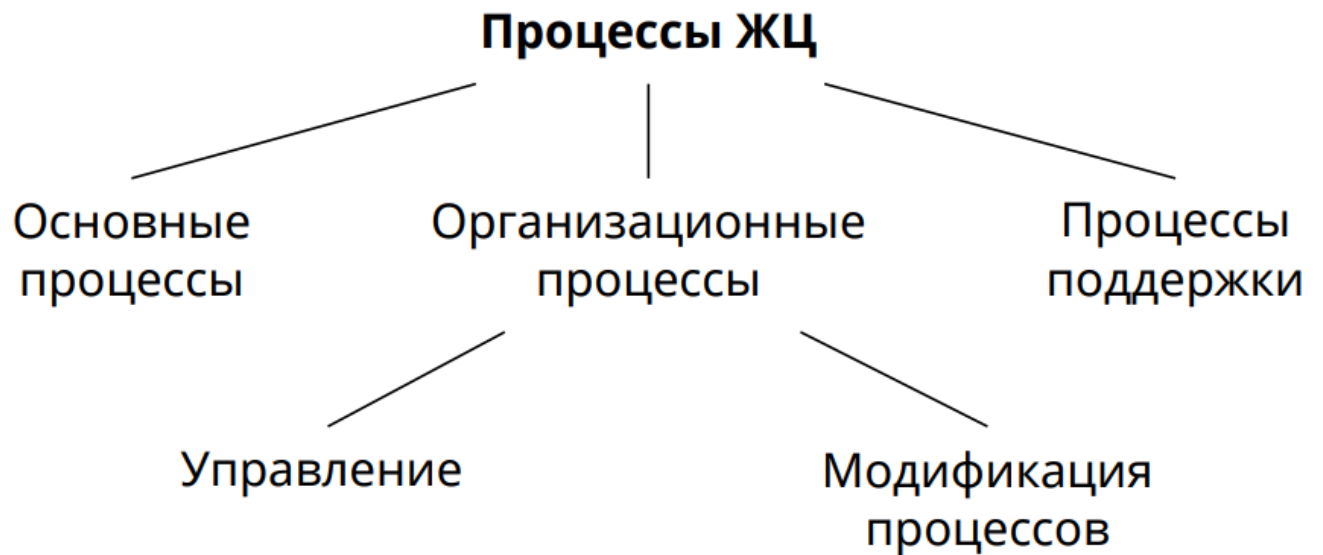
Структура процессов ЖЦ



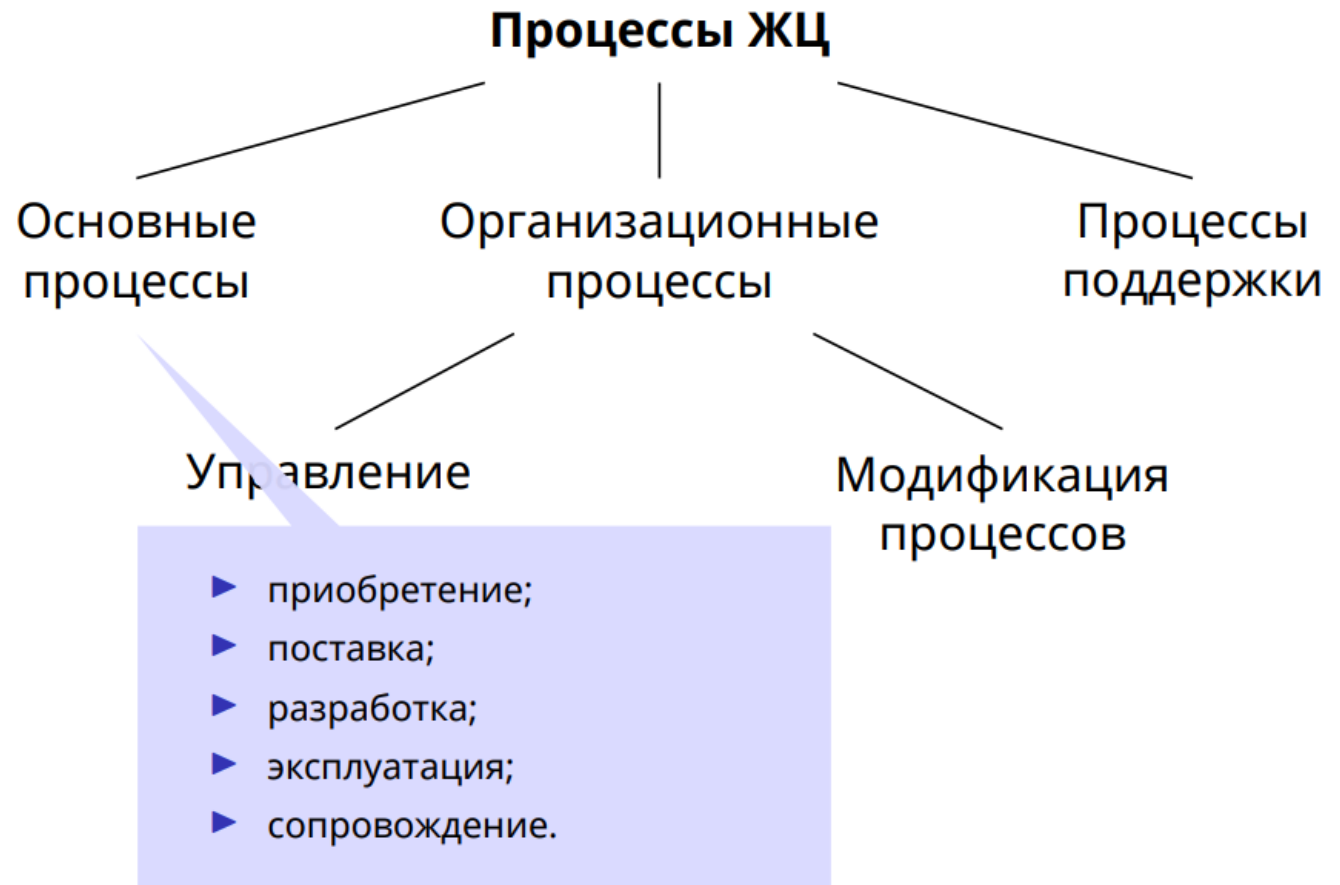
Определение

Процесс — набор взаимосвязанных действий, преобразующих поданную на вход информацию.

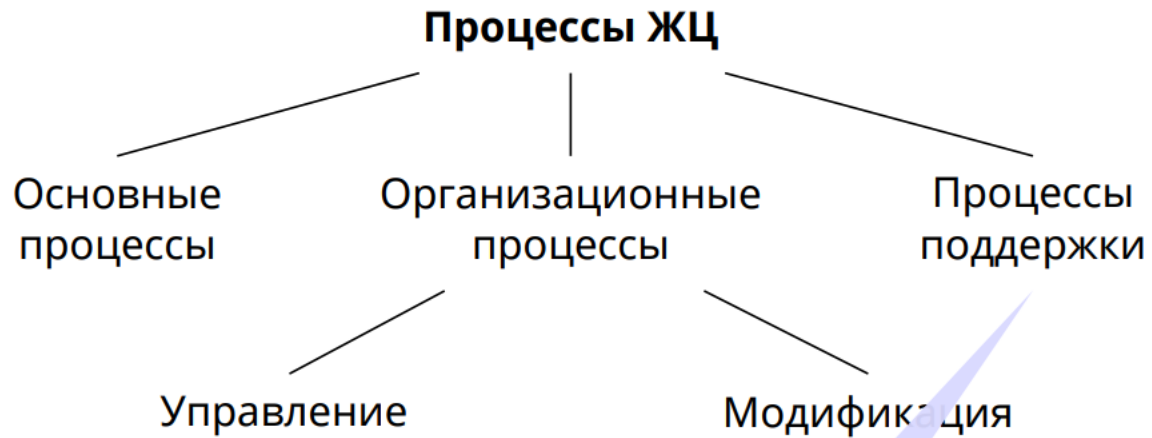
Классификация процессов



Классификация процессов

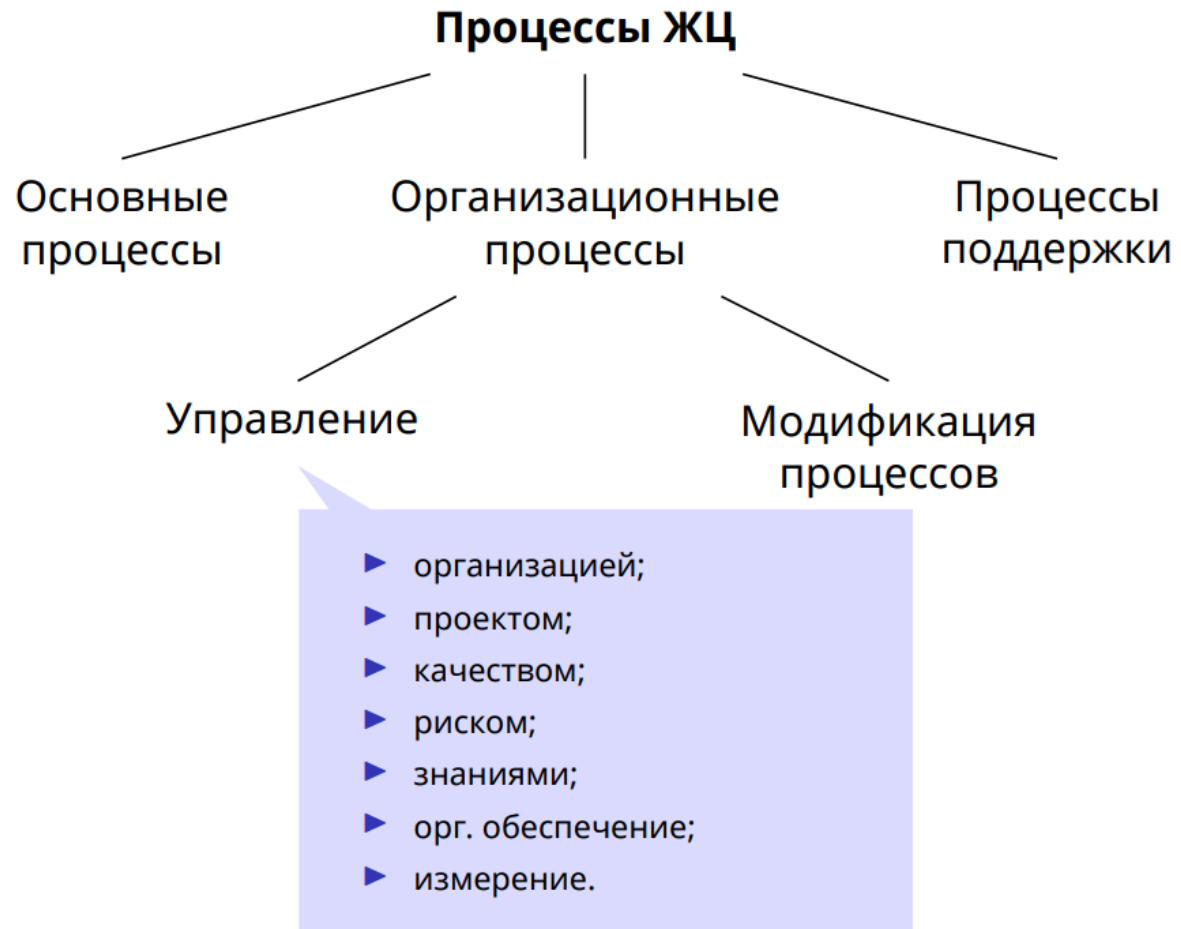


Классификация процессов

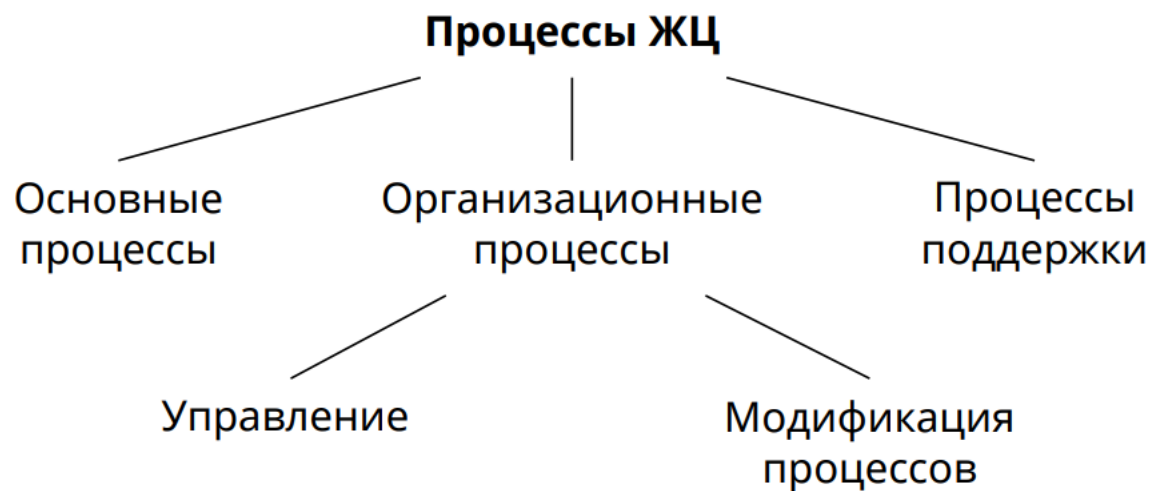


- ▶ документирование;
- ▶ конфигурация;
- ▶ QA;
- ▶ верификация;
- ▶ валидация;
- ▶ аудит;
- ▶ оценка.

Классификация процессов



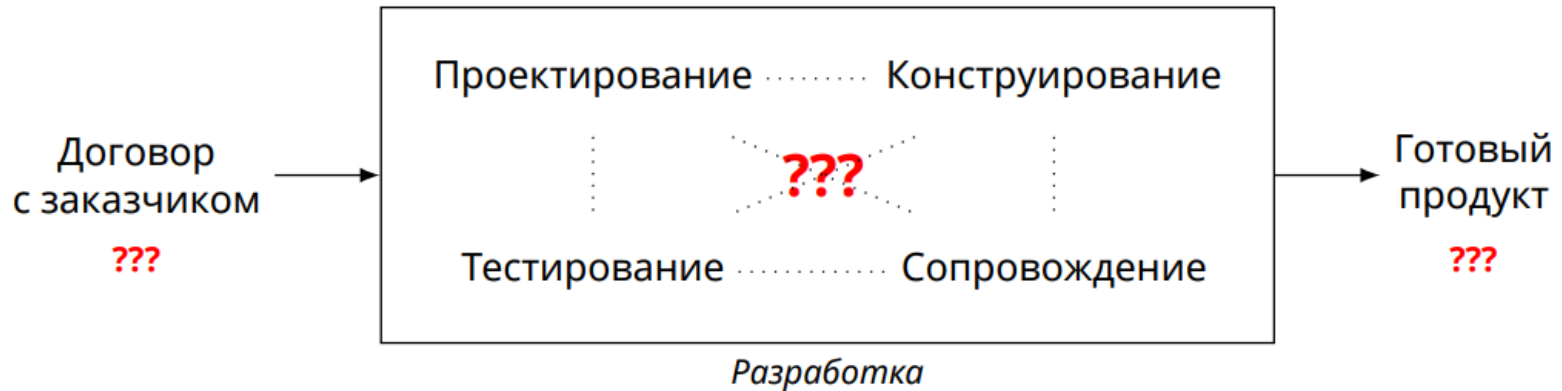
Классификация процессов



- ▶ внедрение;
- ▶ оценка;
- ▶ совершенствование.

Модель жизненного цикла ПО —

структура, определяющая порядок выполнения и взаимосвязи процессов, действий и задач на протяжении жизненного цикла

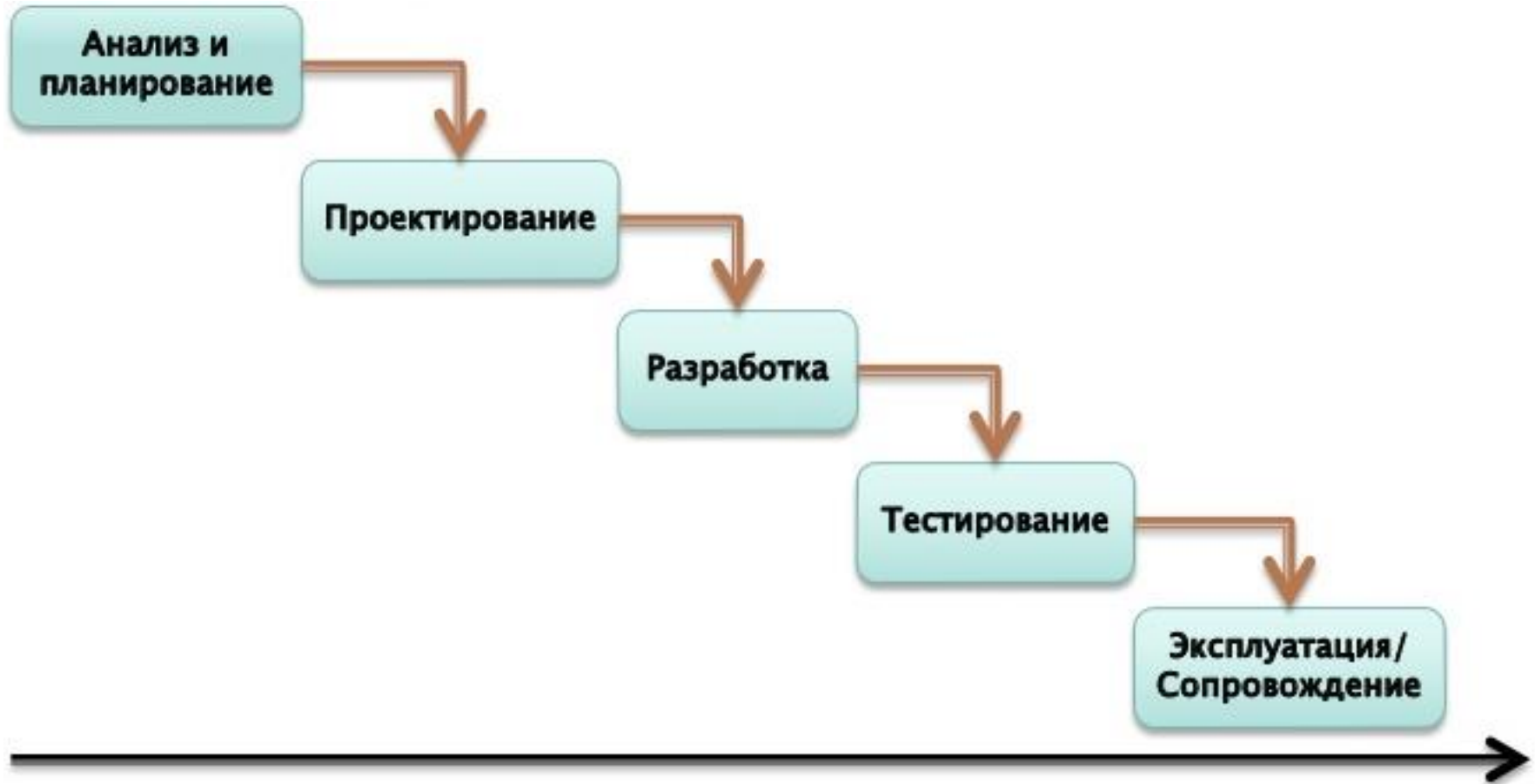


Жизненный цикл — схема упорядочивания работ, касающихся проектирования и разработки программного продукта.

Проблемы:

1. Как соотносятся между собой различные процессы разработки ПО?
2. Каким образом организовано взаимодействие с заказчиком и конечными пользователями?
3. Что считается конечным продуктом разработки?

Классическая водопадная модель (waterfall model)



- ▶ Анализ и планирование
 - Сбор требований
 - Анализ требований
 - Планирование проекта
- ▶ Проектирование
 - Разработка архитектуры
 - Разработка моделей данных
 - Разработка алгоритмов
- ▶ Реализация
 - Кодирование
 - Отладка
- ▶ Тестирование/верификация
- ▶ Сопровождение
 - Внедрение
 - Эксплуатация
 - Внесение изменений

Классическая модель проектирования ПО

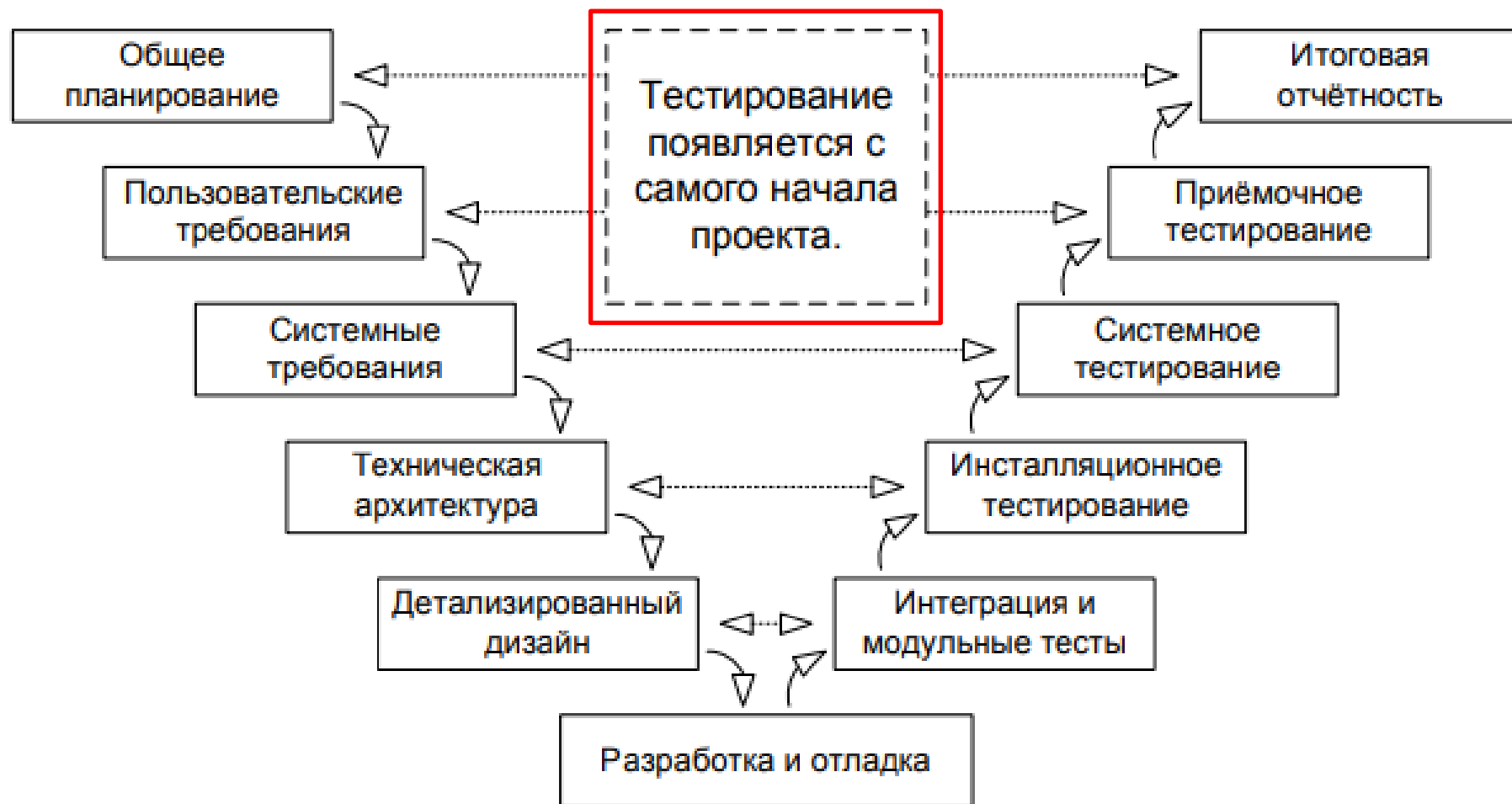
► Достоинства:

- Имеется план и график по всем этапам конструирования
- Ход конструирования – упорядочен
- Имеется богатый опыт использования

► Недостатки:

- Не всегда соответствует реальным проектам (отсутствует гибкость)
- Часто всех требований на начальном этапе нет
- Результат доступен только в конце

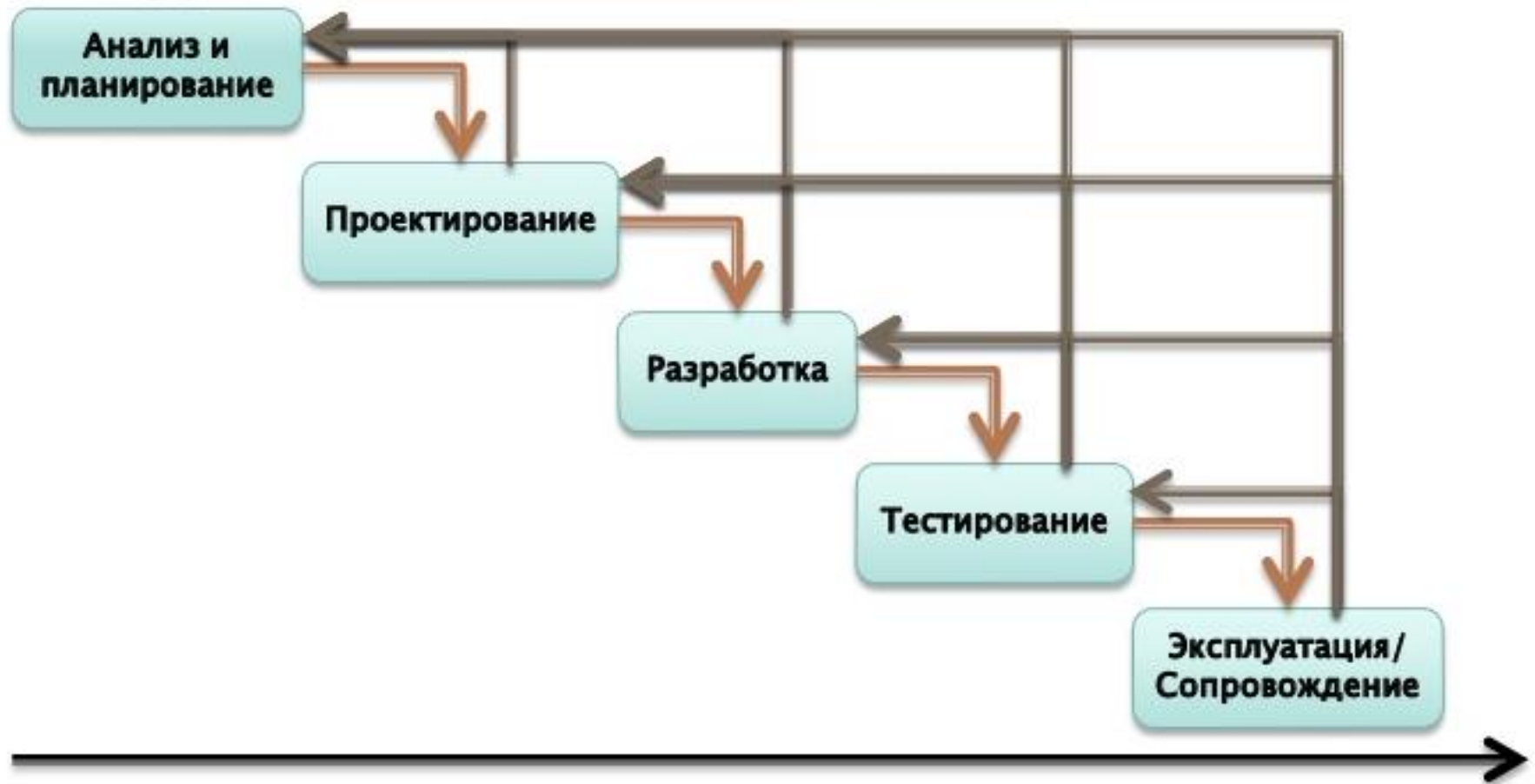
V-образная модель (V-model)



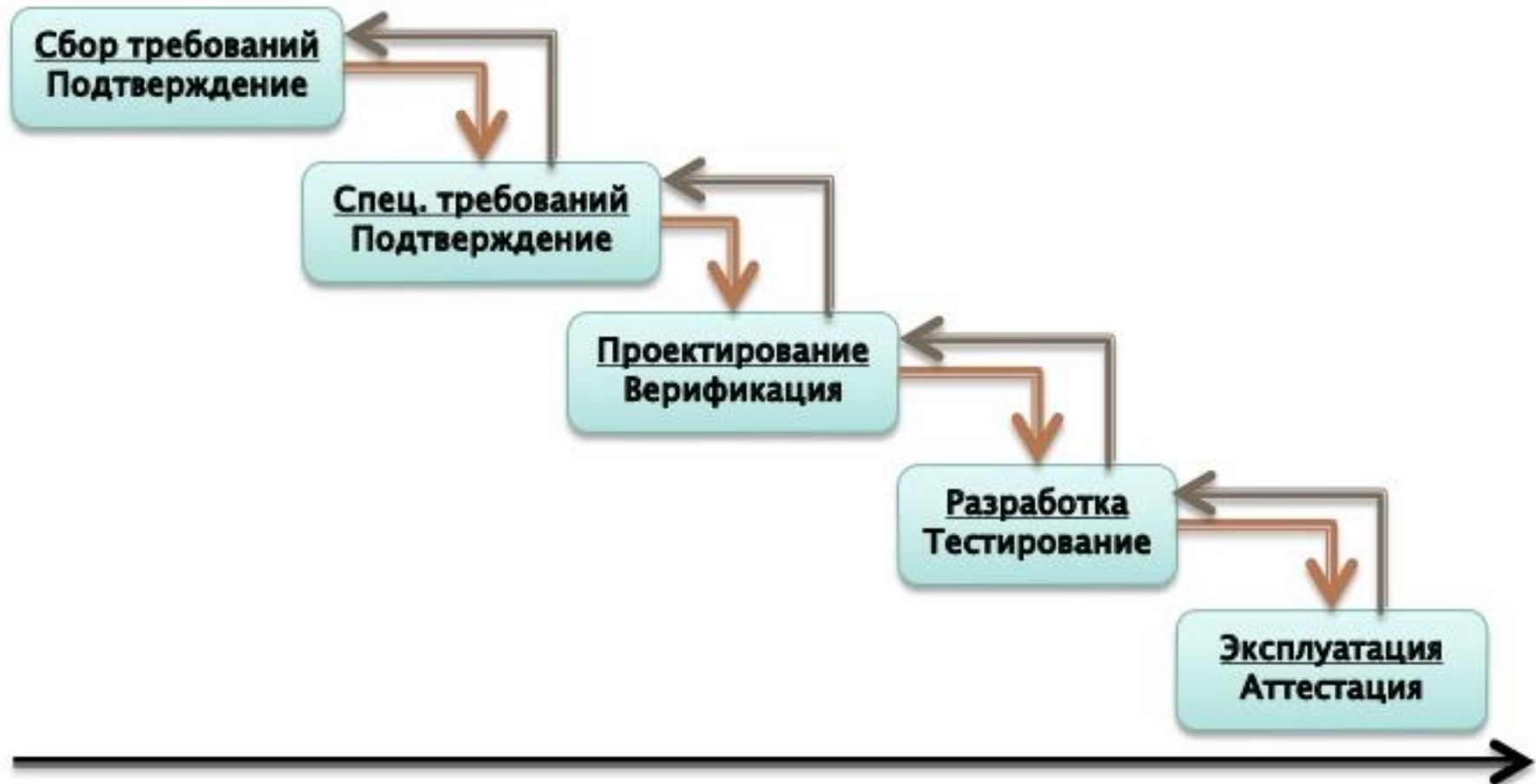
Классическая модель проектирования ПО

- ▶ Имеется несколько модификаций
 - Общепринятая линейная модель
 - Классическая итерационная
 - Предложена В. Ройсом, 1970 г.
 - Обратная связь после каждого этапа
 - Каскадная модель
 - Завершение каждого этапа проверкой
 - Строгая каскадная модель
 - Минимизация возвратов к пройденным этапам

Классическая итерационная модель ППО



Строгая каскадная модель



Итерационная инкрементная модель (iterative, incremental model)

- ▶ Весь проект делится на инкременты – версии продукта с определенной функциональностью
- ▶ Для каждого инкремента выполняется:
 - Анализ
 - Проектирование
 - Разработка
 - Тестирование
- ▶ Результат каждого инкремента – работающий продукт

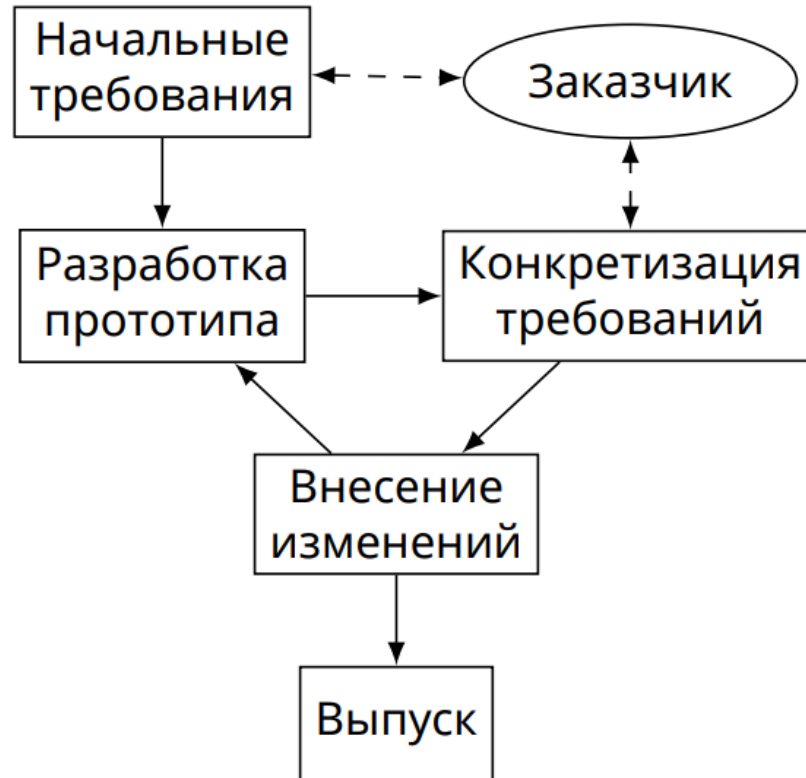


Прототипирование (макетирование)

- ▶ Применяется, когда имеются не все требования
- ▶ Позволяет быстро увидеть некоторые свойства продукта
 - Удобство
 - Внешний вид
 - Применимость
- ▶ Часто применяется при проектировании
 - Информационных систем
 - Программных продуктов с ГПИ
- ▶ Используются средства быстрой разработки приложений



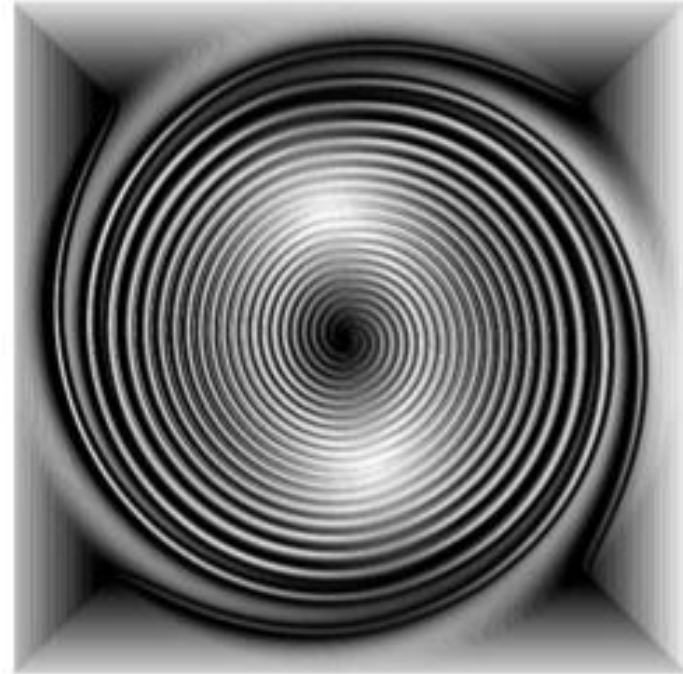
Эволюционная модель



Эволюционная модель — разработка ПО с использованием функциональных прототипов, которые *эволюционируют* в элементы конечного продукта.

Спиральная модель

- ▶ Предложена Б. Боемом, 1988г
- ▶ Базируется:
 - На классическом ЖЦ
 - На макетировании
- ▶ Дополнена анализом рисков
- ▶ Основные компоненты
 - Планирование
 - Анализ
 - Конструирование
 - Оценивание



Спиральная модель



Каждый виток спирали – фаза процесса разработки ПО (постановка задачи, определение требований, дизайн архитектуры и т.д.)

Спиральная модель ППО

► Достоинства:

- Адекватно отражает эволюционный характер проектирования
- Позволяет явно учитывать риски на каждом витке эволюции
- Использует моделирование

► Недостатки:

- Высокие требования к заказчику
- Трудность контроля времени разработки и управления им

Что влияет на успешность программного проекта?

- ▶ Решаемая задача
- ▶ Заказчик
- ▶ Со стороны разработчика
 - Команда разработки
 - Инфраструктура
 - **Выбранная методология проектирования ПО**

Методология разработки ПО

- это регламент, основанный на той или иной модели или принципе, и содержащий конкретное описание действий в процессе разработки ПО
- включает в себя набор методов по управлению разработкой:
правила, техники и принципы, которые делают её более эффективной

Модели и методологии разработки ПО

МОДЕЛИ



Методологии



Что обычно включает методология

- ▶ Методологии определяются:
 - Составом и последовательностью работ
 - Ролью участников проекта
 - Составом и шаблонами документов
 - Организацией и управлением требованиями
 - Порядком контроля и проверки качества
 - Способы взаимодействия участников
 - ...

Какие бывают методологии

Часто организации стараются создать и анонсировать свою собственную корпоративную методологию, которая порой будет отличаться только некоторыми специфичными особенностями.

На данный момент создано огромное количество разнообразных методологий

Известные методологии проектирования ПО

- › Agile software development
- › Agile Unified Process (AUP)
- › Behavior Driven Development (BDD)
- › Big Design Up Front (BDUF)
- › Constructionist design methodology (CDM)
- › Design-driven development (D3)
- › Design Driven Testing (DDT)
- › Domain-Driven Design (DDD)
- › Dynamic Systems Development Method (DSDM)
- › Evolutionary Model
- › Extreme Programming (XP)
- › Feature Driven Development
- › Iterative and incremental development
- › Kaizen
- › Kanban
- › Lean software development
- › Microsoft Solutions Framework (MSF)
- › Model-driven architecture (MDA)
- › Open Unified Process
- › Rapid application development (RAD)
- › Rational Unified Process (RUP)
- › Scrum
- › Software Craftsmanship
- › Spiral model
- › Structured Systems Analysis and Design Method (SSADM)
- › Team Software Process (TSP)
- › Test-driven development (TDD)
- › Unified Process (UP)
- › V-Model
- › Waterfall model
- › Wheel and spoke model

Статистика использования методологий

Методология Waterfall	14%
Методология Agile	12%
Гибридный формат Waterfall + Agile	39%
Не используют никакую формальную методологию	17%
Используют несколько методологий в рамках одного проекта	18%

Исследование ProjectManager: 60% из опрошенных РМ-ов заявили, что в 2022 году для управления проектами используют гибридные методологии

<https://l-a-b-a.com/blog/3059-kak-menyaetsya-prodzhekt-menedzhment-v-2022-5-trendov>

Характеристики методологий проектирования

- ▶ Стратегия конструирования
- ▶ Адаптивность процесса
- ▶ Этапы и связи между ними

Стратегии конструирования ПО

- ▶ **Однократные**
 - Определены все требования
 - Один цикл конструирования
 - Промежуточных версий нет
- ▶ **Инкрементные**
 - Иногда - инкрементно-итеративные
 - Определены все требования
 - Множество циклов конструирования
 - Промежуточные версии могут распространяться
- ▶ **Эволюционные**
 - Иногда - эволюционно-итеративные
 - Определены не все требования
 - Множество циклов конструирования
 - Промежуточные версии могут распространяться



Адаптивность процесса к окружению

- ▶ Тяжеловесные (прогнозирующие)
 - Фиксированные требования
 - Большая команда
 - Разная квалификация разработчиков
- ▶ Адаптивные (облегченные)
 - Постоянно меняющиеся требования
 - Маленькая команда
 - Высококвалифицированные разработчики



Классическая методология разработки ПО

Waterfall



- ▶ Предложена в 1960-х годах, впервые описана 1970 г., В. Ройсом
- ▶ Водопадный (однократный) подход
- ▶ Относится к прогнозирующим методологиям
- ▶ Предполагает полное наличие всех требований на момент старта проекта
- ▶ Требования не могут меняться в процессе проектирования
- ▶ Программный продукт появляется по окончании проектирования
- ▶ Промежуточные версии не предусмотрены



Классический Waterfall?

В июне 1956 года
в США на симпозиуме по передовым
методам программирования для
цифровых вычислительных машин
был представлен доклад
Герберта Бенингтона (Herbert D.
Benington)
из MIT Lincoln Laboratory: «Production of
Large Computer Programs»

Доклад, иллюстрирующий производство
программного обеспечения системы
ПВО SAGE, отображал переходы от
одной фазы производства ПО системы к
другой.

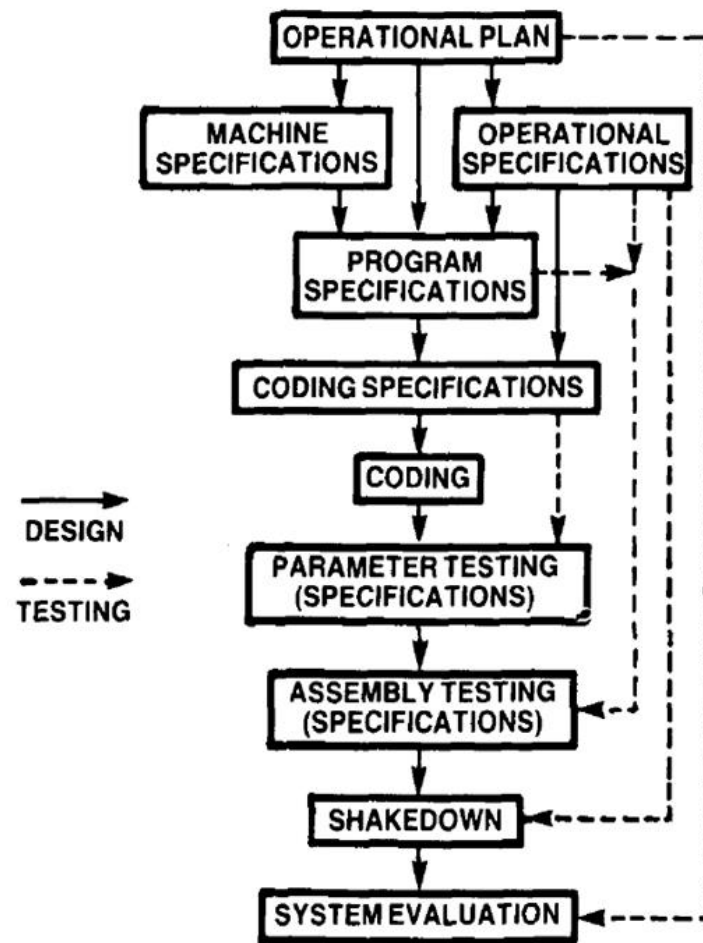


Figure 4. Program production. Production of a large-program system proceeds from a general operational plan through system evaluation; for example, assembly testing verifies operational and program specifications.

Классический Waterfall?

В 1968 г. в немецком городе Гармиш под эгидой научного комитета НАТО была проведена первая, а затем в 1969 году в Риме вторая конференция по программной инженерии.

Присутствовало более 50 ведущих учёных-практиков из 11 стран мира, среди которых были такие звезды мировой науки, как Edsger W. Dijkstra (Эдсгер Вибе Дейкстра) и Charles Antony Richard Hoare (Чарльз Энтони Ричард Хоар).

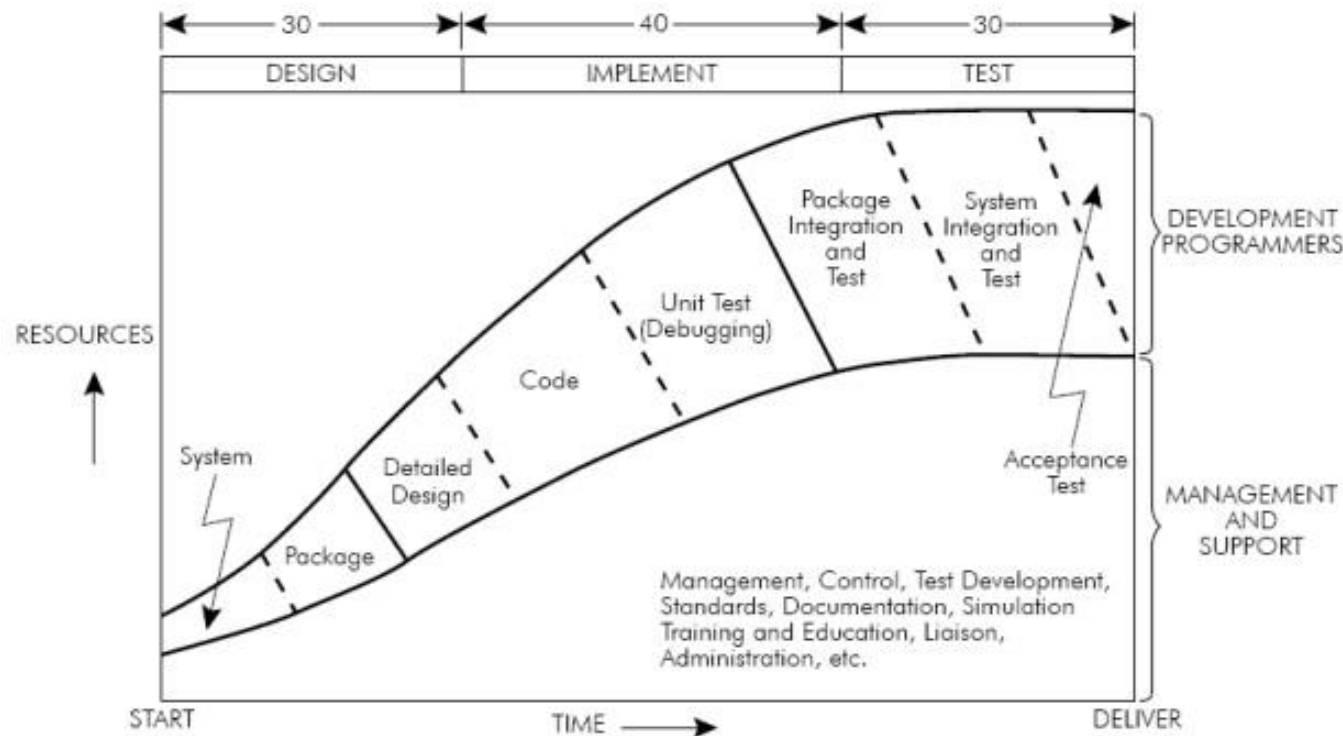


Figure 2. System Life Cycle

Классический Waterfall? Статья У. Ройса, 1970

В августе 1970 года в США в журнале IEEE директор центра по технологии программного обеспечения компании Lockheed доктор Уинстон Ройс (Dr. Winston W. Royce) опубликовал статью «Управление разработкой крупных программных систем»

- **Описаны варианты методологии Waterfall**
- **Показаны проблемы реализации и слабые места методологии**
- **Приведены практические рекомендации по применению**

Шаг 1. Проектирование программ важнее всего

Шаг 2. Документируйте разработку

Шаг 3. Раннее прототипирование и моделирование

Шаг 4. Планировать, управлять и контролировать испытания

Шаг 5. Вовлеките заказчика

Классический подход к разработке ПО – система стандартов ГОСТ 34

Стадии и этапы создания автоматизированной системы по ГОСТ Р 59793—2021

Стадии	Этапы работ
1. Формирование требований к АС	1.1. Обследование объекта и обоснование необходимости создания АС 1.2. Формирование требований пользователя к АС 1.3. Оформление отчёта о выполненной работе и заявки на разработку АС (тактико-технического задания)
2. Разработка концепции АС	2.1. Изучение объекта 2.2. Проведение необходимых научно-исследовательских работ 2.3. Разработка вариантов концепции АС, удовлетворяющего требованиям пользователя 2.4. Оформление отчёта о выполненной работе
3. Техническое задание	3.1. Разработка и утверждение технического задания на создание АС
4. Эскизный проект	4.1. Разработка предварительных проектных решений по системе и её частям 4.2. Разработка документации на АС и ее части
5. Технический проект	5.1. Разработка проектных решений по системе и её частям 5.2. Разработка документации на АС и ее части 5.3. Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации
6. Рабочая документация	6.1. Разработка рабочей документации на систему и её части. 6.2. Разработка или адаптация программ.
7. Ввод в действие	7.1. Подготовка объекта автоматизации к вводу АС в действие 7.2. Подготовка персонала 7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями) 7.4. Строительно-монтажные работы 7.5. Пусконаладочные работы 7.6. Проведение предварительных испытаний 7.7. Проведение опытной эксплуатации 7.8. Проведение приёмочных испытаний

**Стадии и этапы создания автоматизирован-ной системы
по ГОСТ Р 59793—2021 (ГОСТ 34.601-90)**

Стадия по ГОСТ Р 59793– 2021	Содержание работ
1. Формирование требований к АС	Заказчик самостоятельно определяет требования пользователя к системе и размещает заказ на закупку
2. Разработка концепции АС	Привлеченный исполнитель обследует инфраструктуру Заказчика и нормативную документацию, определяет угрозы безопасности информации, подготавливает Отчет об обследовании, Модель угроз, согласование которой может продолжаться в течение последующих стадий проекта
3. Техническое задание	Выполняется разработка и утверждение Технического задания на создание системы
4. Эскизный проект	Выполняется разработка и утверждение комплекта документации технического проекта, рабочего проекта на систему и комплекта организационно-распорядительной документации
5. Технический проект	
6. Рабочая документация	
7. Ввод в действие	Выполняются пуско-наладочные работы, подготовка персонала, проводятся испытания системы и, при необходимости, аттестация системы по требованиям безопасности информации
8. Сопровождение АС	Выполняются работы в соответствии с гарантийными обязательствами и послегарантийное обслуживание. Стадия работ продолжается до вывода системы из эксплуатации

**Стадии и этапы создания автоматизирован-ной системы
по ГОСТ Р 59793—2021 (ГОСТ 34.601-90)**

Стадии	Этапы работ
1 Формирование требований к АС	1.1 Обследование объекта и обоснование необходимости создания АС 1.2 Формирование требований пользователя к АС 1.3 Оформлениe отчета о выполненной работе
2 Разработка концепции АС	2.1 Изучение объекта 2.2 Проведение необходимых научно-исследовательских работ 2.3 Разработка вариантов концепции АС и выбор варианта концепции АС. удовлетворяющего требованиям пользователя 2.4 Оценка рисков проекта 2.5 Оформлениe отчета о выполненной работе
3 Техническое задание	3.1 Разработка и утверждение технического задания на создание АС
4 Эскизный проект	4.1 Разработка предварительных проектных решений по АС и ее частям 4.2 Разработка документации на АС и ее части

Стадии	Этапы работ
5 Технический проект	<ul style="list-style-type: none"> •5.1 Разработка проектных решений по АС и ее частям •5.2 Разработка документации на АС и ее части •5.3 Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку •5.4 Разработка заданий на проектирование в смежных частях проекта объекта автоматизации
6 Рабочая документация	<ul style="list-style-type: none"> •6.1 Разработка рабочей документации на АС и ее части •6.2 Разработка или адаптация отдельных видов обеспечения АС
7 Ввод а действие	<ul style="list-style-type: none"> •7.1 Подготовка объекта автоматизации к вводу АС в действие •7.2 Подготовка персонала •7.3 Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями) •7.4 Строительно-монтажные работы •7.5 Пусконаладочные работы •7.6 Проведение предварительных испытаний •7.7 Проведение опытной эксплуатации •7.8 Проведение приемочных испытаний
8 Сопровождение АС	<ul style="list-style-type: none"> •8.1 Выполнение работ в соответствии с гарантийными обязательствами •8.2 Послегарантийное обслуживание

Стадии и этапы работ, выполняемые организациями, участвующими в работах по созданию АС устанавливаются в организационно-распорядительных документах этих организаций, договорах и техническом задании на создание АС.

Допускается исключать отдельные этапы работ на всех стадиях.

Допускается исключать стадию «Эскизный проект». В случае исключения стадии «Эскизный проект» работы, выполняемые на этой стадии, могут включаться в стадию «Технический проект».

Допускается объединять стадии «Технический проект» и «Рабочая документация» в одну стадию «Технорабочий проект».

В зависимости от специфики создаваемых АС и условий их создания допускается выполнение отдельных этапов работ до завершения предшествующих стадий, параллельное во времени выполнение этапов работ, включение новых этапов работ.