ТСПС

ЛЕКЦИЯ 1

ОРГАНИЗАЦИЯ ПРОЦЕССА КОНСТРУИРОВАНИЯ ПО ИНФОРМАЦИОННЫХ СИСТЕМ

ОРГАНИЗАЦИЯ ПРОЦЕССА КОНСТРУИРОВАНИЯ ПО

Технология создания программного обеспечения ***информационных систем*** — система инженерных принципов для создания экономичного, надежного и эффективно работающего ПО.

ЭТАПЫ КОНСТРУИРОВАНИЯ ПО

СОДЕРЖАНИЕ ЭТАПОВ.

1.Анализ требований

- ✓ Документирование требований к будущему ПО
- ✓ Выделение технических ограничений
- ✓ Задание временных и финансовых ограничений

Результат: создание подробной спецификации, отвечающей всем требованиям заказчика.

ДОПОЛНИТЕЛЬНО: ВИДЕНИЕ ПРОЕКТА

- ✓ краткое описание проекта;
- ✓ бизнес-цели;
- ✓ критерии успеха проекта;
- ✓ факторы бизнес-рисков;
- ✓ описание конечного пользователя продукта.

Готовый документ передается на утверждение заказчику для утверждения.

СОДЕРЖАНИЕ ЭТАПОВ

2.Проектирование

Состоит в создании представлений:

- ✓ архитектуры ПО;
- ✓ модульной структуры ПО;
- ✓ алгоритмической структуры ПО;
- ✓ структуры данных;
- ✓ входного и выходного интерфейса (Design Specification Document, DSD).

СОДЕРЖАНИЕ ЭТАПОВ.

2.Проектирование

Для упрощения визуализации процесса проектирования используются нотации:

- ✓ Блок-схемы;
- ✓ UML-диаграммы;
- ✓ Макеты.

ДОПОЛНИТЕЛЬНО:МАКЕТИРОВАНИЕ

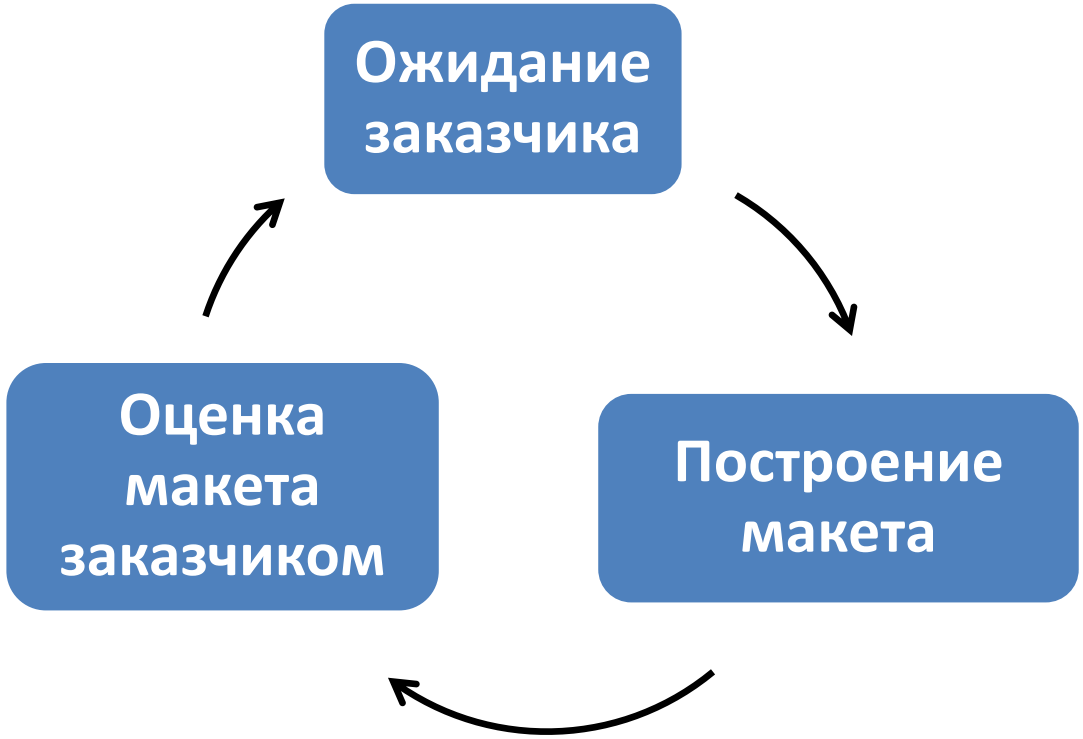
Макетирование — это процесс создания модели требуемого программного продукта.

Виды моделей:

- ✓ бумажный макет или макет на основе ПК;
- ✓ работающий макет (выполняет некоторую часть требуемых функций);
- ✓ существующая программа (характеристики которой затем должны быть улучшены).

МАКЕТИРОВАНИЕ

Цель макетирования — снять неопределенности в требованиях заказчика.



Цикл макетирования

МАКЕТИРОВАНИЕ

Преимущество: обеспечивает определение полных требований к ПО.

Недостатки:

- ✓ заказчик может принять макет за продукт;
- ✓ разработчик может принять макет за продукт.

Программы для создания макетов:

http://invisionapp.com/

https://webflow.com/

https://moqups.com/

СОДЕРЖАНИЕ ЭТАПОВ.

3.Разработка

Четыре основных стадии:

1) Разработка алгоритмов – создание логики работы программы;

2) Написание исходного кода;

3) Компиляция – преобразование в машинный код;

4) Тестирование и отладка компонентов системы.

СОДЕРЖАНИЕ ЭТАПОВ

4.Документирование.

Уровни документации:

– *Архитектурная* (проектная) – документы, описывающие модели, методологии, инструменты и средства разработки, выбранные для данного проекта.

4.Документирование.

Уровни документации:

– *Техническая* – вся сопровождающая разработку документация. Сюда входят различные документы, поясняющие работу системы на уровне отдельных модулей. Как правило, пишется в виде комментариев к исходному коду, которые впоследствии структурируются в виде HTML-документов.

4.Документирование.

Уровни документации:

– *Пользовательская* – включает справочные и поясняющие материалы, необходимые конечному пользователю для работы с системой.

– *Маркетинговая* – включает рекламные материалы, сопровождающие выпуск продукта. Ее цель – в красочной форме представить функциональность и конкурентные преимущества продукта.

СОДЕРЖАНИЕ ЭТАПОВ.

5.Сопровождение

Виды работ:

- ✓ исправление ошибок;
- ✓ адаптация к изменениям внешней для ПО среды;
- ✓ усовершенствование ПО по требованиям заказчика.

СОДЕРЖАНИЕ ЭТАПОВ.

5.Сопровождение

- ✓ Исполнители: сотрудники отдела технической поддержки
- ✓ Решаемые задачи:
  - – обратная связь с пользователями, их консультирование.
  - –составление и передача отчетов об ошибках команде разработки.
    - В зависимости от серьезности проблемы, либо немедленно выпускает исправление (т.н. hot-fix), либо откладывает его до следующей версии программы.

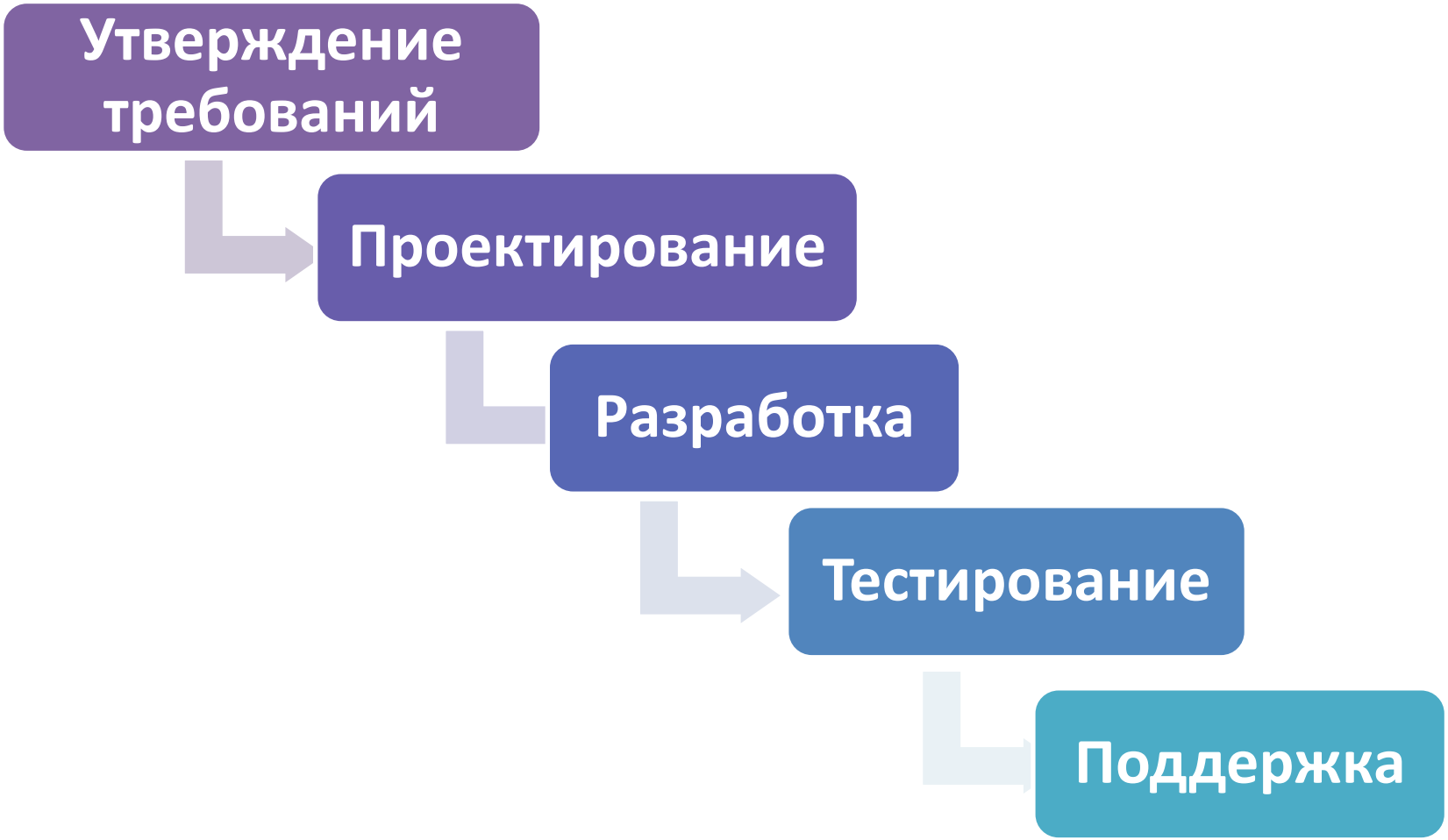
5.Сопровождение

- ✓ Решаемые задачи:
  - Сбор и систематизация различных метрик – показателей работы программы в реальных условиях.

СТРАТЕГИИ КОНСТРУИРОВАНИЯ ПО

- ✓ однократный проход (водопадная стратегия) — линейная последовательность этапов конструирования;
- ✓ инкрементная стратегия (итеративная модель);
- ✓ эволюционная стратегия (спиральная модель).

КАСКАДНАЯ МОДЕЛЬ (Waterfall model)



КАСКАДНАЯ МОДЕЛЬ

Представляет собой последовательность этапов, причем переход на следующий этап происходит только после полного завершения работ на текущем этапе.

WATERFALL

Достоинства

- ✓ Полное документирование каждого этапа;
- ✓ Четкое планирование сроков и затрат;
- ✓ Прозрачность процессов для заказчика
  - –

WATERFALL

Недостатки

- ✓ реальные проекты часто требуют отклонения от стандартной последовательности шагов;
- ✓ модель основана на точной формулировке исходных требований к ПО (реально в начале проекта требования заказчика определены лишь частично);
- ✓ результаты проекта доступны заказчику только в конце работы.

WATERFALL

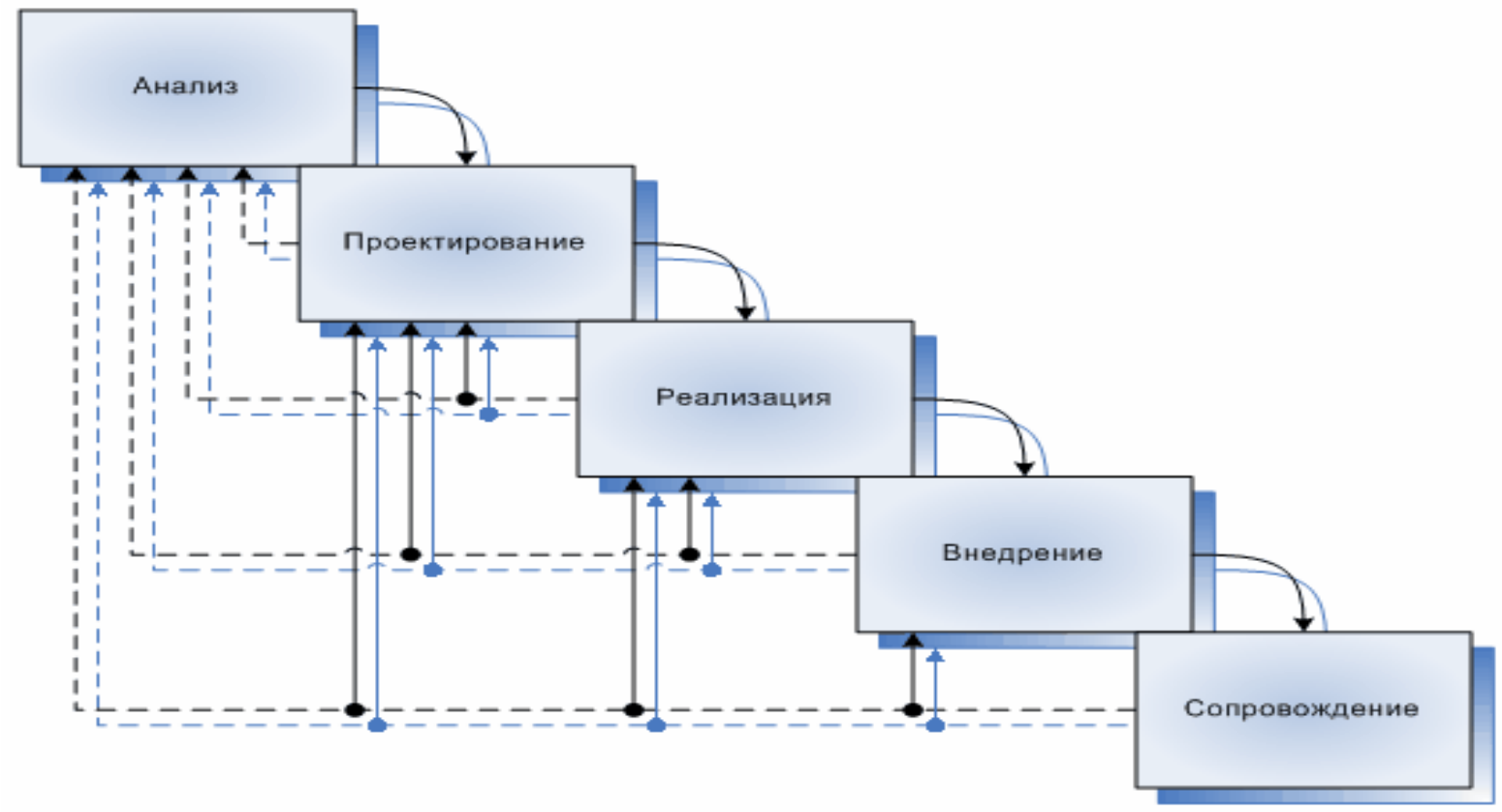
Применение:

При создании систем, критичных для жизни человека

- ✓ в военном деле;
- ✓ в космических разработках;
- ✓ в медицине;
- ✓ в авиации (контроль полетов и т.п.)

Особенность разработки ПО по Waterfall

- ✓ При устранении ошибок текущего этапа – переход к предшествующим этапам, начиная с анализа



ИТЕРАТИВНАЯ МОДЕЛЬ

- ✓ Жизненный цикл продукта разбит на ряд отдельных мини-циклов, которые называются **итерациями**.

- ✓ В каждой из итераций происходит разработка отдельного компонента системы, после чего этот компонент добавляется к уже ранее разработанному функционалу.

ИТЕРАТИВНАЯ МОДЕЛЬ

- ✓ Требования к ПО полностью определены в начале процесса разработки;
- ✓ Требования разбиты по версиям продукта;
- ✓ Релиз промежуточных версий может отсутствовать.

ИТЕРАТИВНАЯ МОДЕЛЬ

- ✓ Поинкементная разработка

ИТЕРАТИВНАЯ МОДЕЛЬ

- На каждой линейной последовательности этапов создается инкремент ПО.
- Первый инкремент приводит к получению базового продукта, реализующего базовые требования.
- Каждый последующий инкремент предусматривает модификацию базового продукта, обеспечивающую дополнительные характеристики и функциональность.

ИТЕРАТИВНАЯ МОДЕЛЬ

Преимущества

- ✓ Быстрое создание работающей версии ПО;
- ✓ каждая итерация – этап, для которого тестирование и анализ рисков обеспечить проще, чем для всего жизненного цикла продукта.

СПИРАЛЬНАЯ МОДЕЛЬ

- ✓ требования меняются в процессе разработки
- ✓ разработка версий с наращиванием (изменением) функциональности в каждой версии.

СРАВНЕНИЕ МОДЕЛЕЙ

- ✓ водопадная модель
  - требования полностью известны до начала разработки,
  - на выходе ТОЛЬКО ГОТОВЫЙ ПРОДУКТ – без промежуточных этапов
- ✓ инкрементная модель
  - требования полностью известны до начала разработки,
  - разработка версий с поэтапным наращиванием функциональности
- ✓ спиральная (эволюционная) модель
  - требования полностью НЕ ИЗВЕСТНЫ до начала разработки,
  - разработка версий с поэтапным наращиванием или изменение функциональности при изменении требований
  - Оценка результатов по циклам разработки

СПИРАЛЬНАЯ МОДЕЛЬ

- ✓ жизненный путь разрабатываемого продукта изображается в виде спирали, которая, начавшись на этапе планирования, раскручивается с прохождением каждого следующего шага.
- ✓ На выходе из очередного витка мы должны получить готовый протестированный прототип пригодный для релиза.

СПИРАЛЬНАЯ МОДЕЛЬ

СПИРАЛЬНАЯ МОДЕЛЬ

- ✓ *1* — начальный сбор требований и планирование проекта;
- ✓ *2* — та же работа, но на основе рекомендаций заказчика;
- ✓ 3 — анализ риска на основе начальных требований;
- ✓ *4* — анализ риска на основе реакции заказчика;

СПИРАЛЬНАЯ МОДЕЛЬ

- ✓ 5 — выход;
- ✓ 6 — начальный макет системы;
- ✓ 7 — следующий уровень макета;
- ✓ *8* — сконструированная система;
- ✓ 9 — оценивание заказчиком.

СПИРАЛЬНАЯ МОДЕЛЬ

- ✓ Базовые действия в цикле разработки:
  - –**Планирование** — определение целей, вариантов и ограничений.
  - –**Анализ риска** — анализ вариантов реализации и распознавание/выбор риска.
  - –**Конструирование** — разработка продукта следующего уровня.
  - –**Оценивание** — оценка заказчиком текущих результатов конструирования.
- ✓ С каждой итерацией по спирали строятся все более полные версии ПО.

СПИРАЛЬНАЯ МОДЕЛЬ

Риски

- ✓ Нереалистичный бюджет и сроки;
- ✓ Дефицит специалистов;
- ✓ Частые изменения требований;
- ✓ Чрезмерная оптимизация;
- ✓ Низкая производительность системы;
- ✓ Несоответствие уровня квалификации специалистов разных отделов.

СПИРАЛЬНАЯ МОДЕЛЬ

Достоинства

- ✓ улучшенный анализ рисков;
- ✓ гибкость – возможность внесения изменений и добавления новой функциональности даже на относительно поздних этапах;
- ✓ раннее создание рабочих прототипов.

–

СПИРАЛЬНАЯ МОДЕЛЬ

Недостатки

- ✓ может быть достаточно дорогой в использовании;
- ✓ управление рисками требует привлечения высококлассных специалистов;
- ✓ успех процесса в большой степени зависит от стадии анализа рисков;
- ✓ не подходит для небольших проектов.

СПИРАЛЬНАЯ МОДЕЛЬ

Условия использования

- ✓ – когда важен анализ рисков и затрат;
- ✓ – крупные долгосрочные проекты с отсутствием четких требований или вероятностью их динамического изменения;
- ✓ – при разработке новой линейки продуктов.
- ✓ ключевой элемент (Agile) подходов к разработке программного обеспечения.