

**НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ УКРАИНЫ «КПИ»**



Модели жизненного цикла программного обеспечения

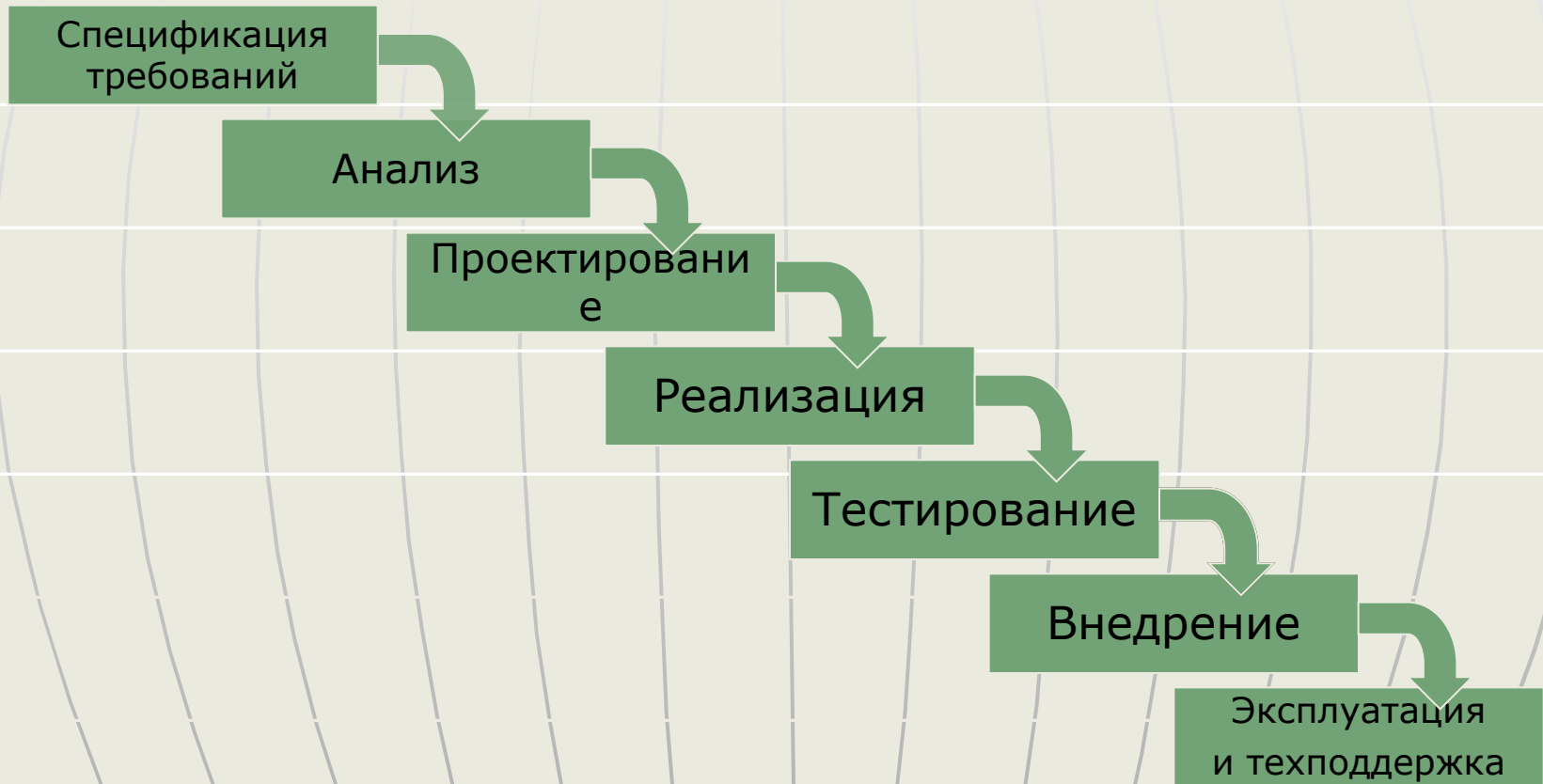
План

- Классические модели жизненного цикла ПО
- Современные модели жизненного цикла ПО
- Описание основных этапов жизненного цикла

Классические модели жизненного цикла ПО

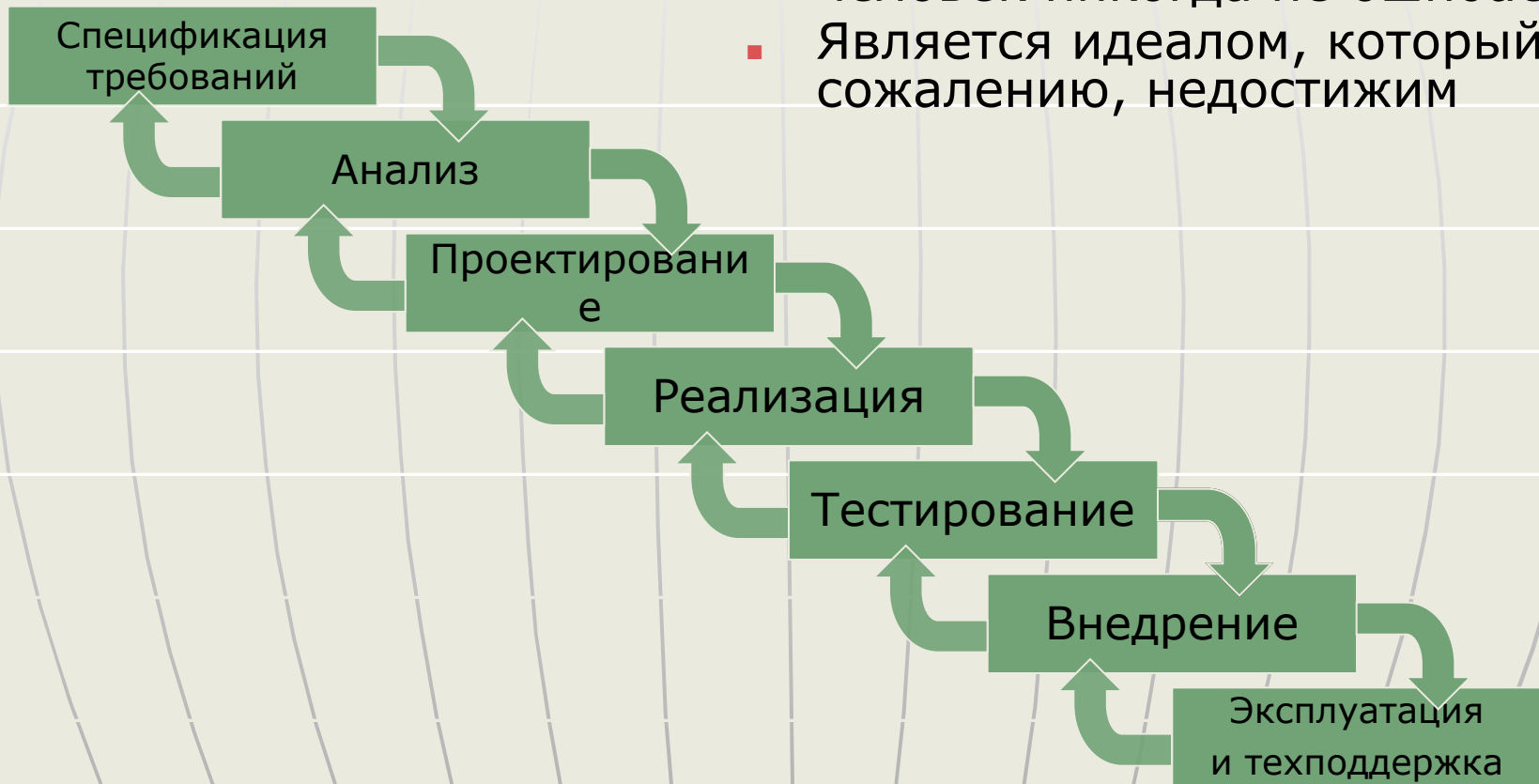
- Каскадная модель
- Каскадная модель с промежуточным контролем
- Спиральная модель

Каскадная модель



Каскадная модель с обратной СВЯЗЬЮ

- Модель базируется на том, что человек никогда не ошибается
- Является идеалом, который, к сожалению, недостижим



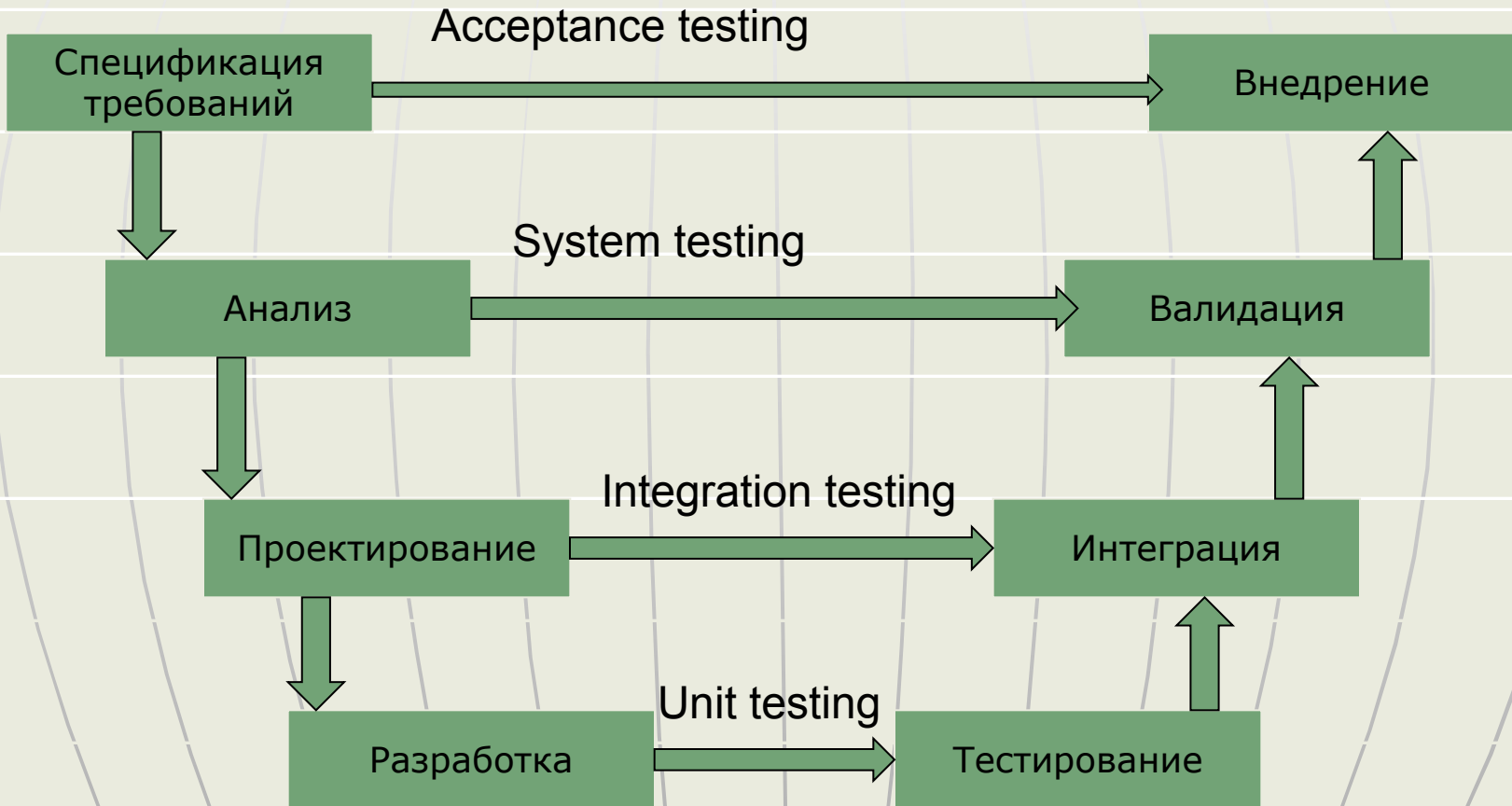
Спираль



Современные модели жизненного цикла ПО

- V-model
- Прототипирование
- Инкрементная модель
- Итерационная модель

V-model



Прототипирование

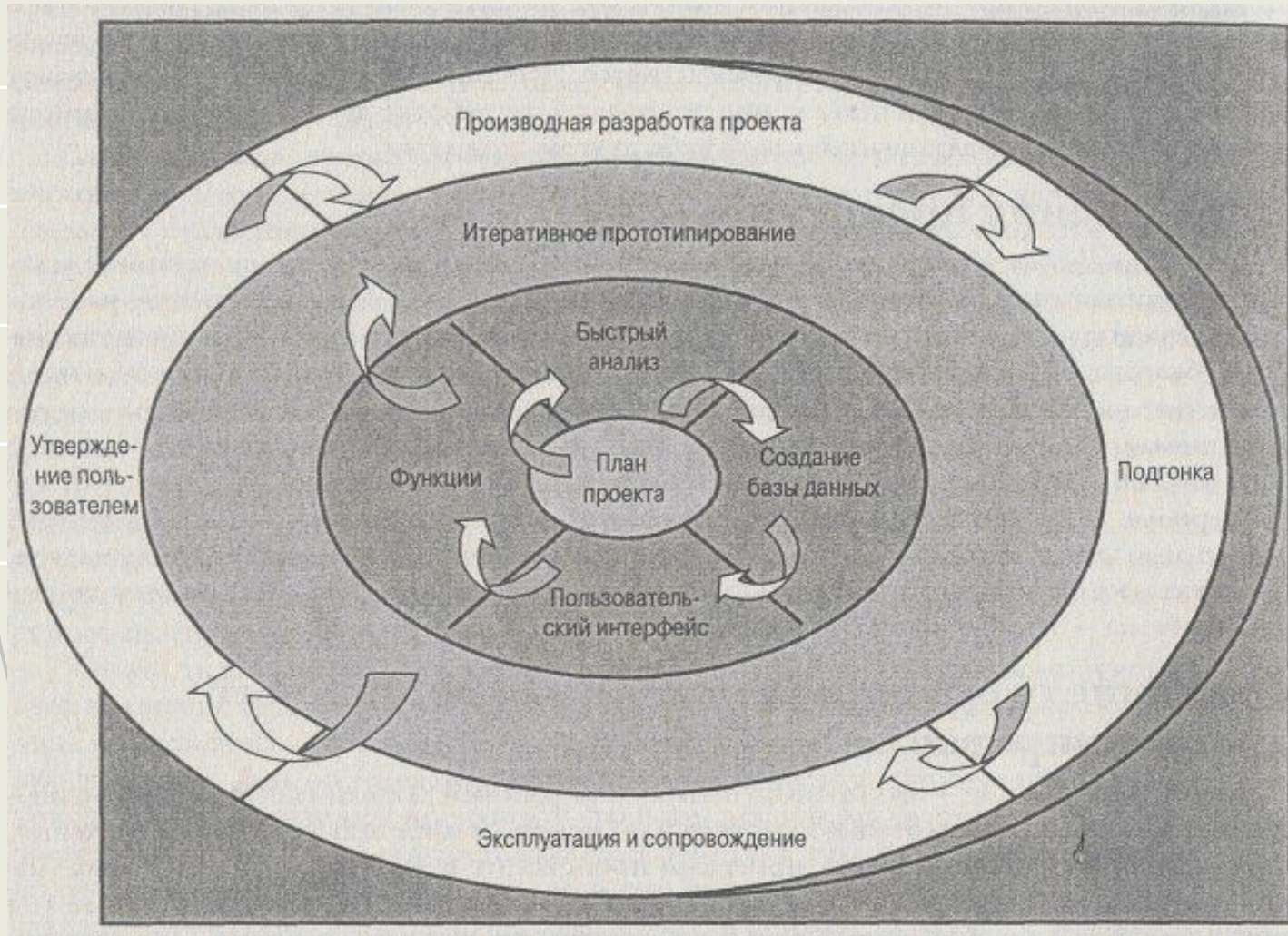
Авторы: Джон Коннэлл, Линда Шафер

Описание: метод, предназначенный для определения требований, при котором потребности пользователя извлекаются, представляются и разрабатываются посредством построения рабочей модели конечной системы — быстро и в требуемом контексте

Показания к применению:

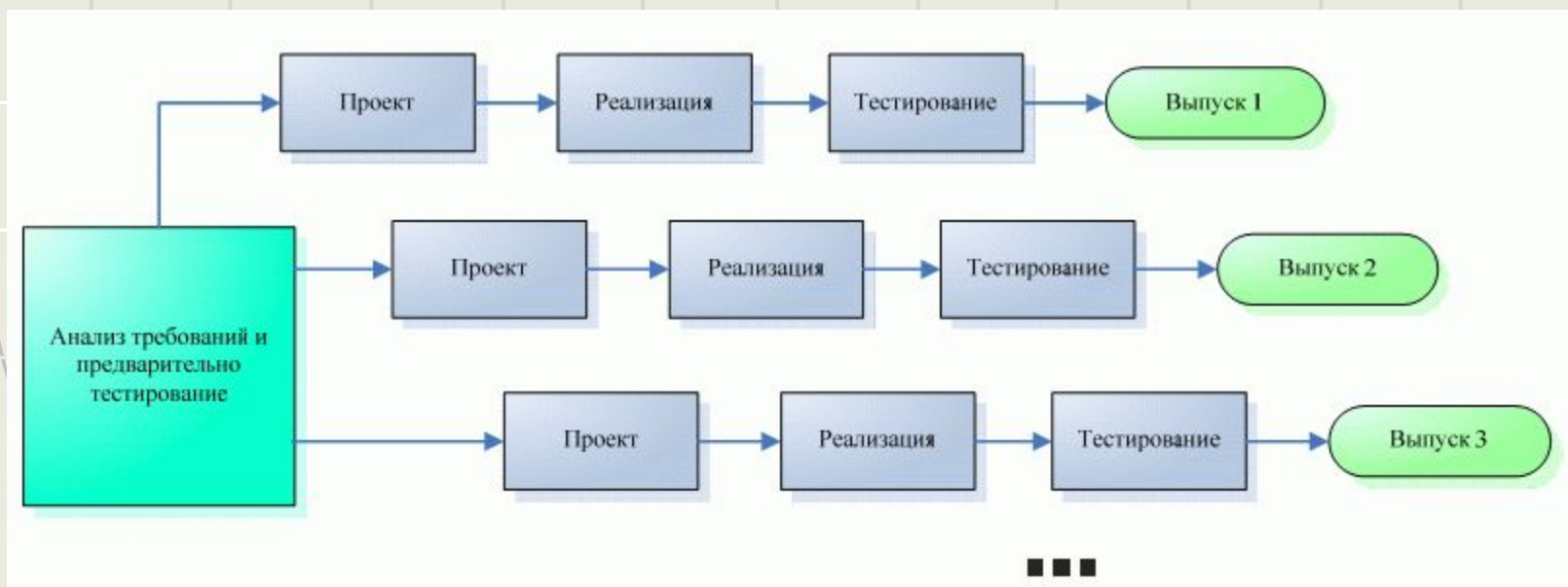
- Требования расплывчатые/неопределенные
- Требуется проверка концепции реализации
- Новая, не имеющая аналогов разработка

Прототипирование



Инкрементная модель

Инкрементная разработка представляет собой процесс частичной реализации всей системы и медленного наращивания функциональных возможностей.



Итеративная модель

Особенности итеративной модели:

- короткий цикл разработки на итерацию
- Фиксированные интервалы



Различие инкрементной и итеративной моделей



Инкрементная модель



Итеративная модель



Основные этапы жизненного цикла

- **Спецификация требований**
(Оценка реального объема проекта, его целей и задач)
- **Анализ**
(исследование бизнес-процессов)
- **Проектирование**
(формирование модели данных)
- **Реализация**
- **Тестирование**
- **Внедрение**
- **Эксплуатация и техподдержка**

Спецификация требований

Цели:

- Оценка реального объема проекта
- Определение сущностей и функций на высоком уровне

Результат фазы – документ описывающий:

- График выполнения работ
 - Что когда и в каком объеме
- Бюджет проекта
 - Планируемые затраты
 - Планируемые выгоды
 - Время окупаемости
- Риски
- Целевая аудитория

Основные вопросы:

- Будем ли делать этот проект за такие деньги или нет
- Будем ли вообще делать ЭТОТ проект с ЭТИМ разработчиком?

Анализ

Результат – информационная модель системы

Сбор и оценка информации от заказчиков, проверка её на:

- Полноту
- Непротиворечивость
- Полезность
- Дублируемость

На данном этапе проводят анализ (тестирование) требований

Наиболее частый инструментарий: Rational Rose, Microsoft Visio

На выходе получают диаграммы потоков данных и диаграммы жизненных циклов сущностей (описание системы в динамике)

Проектирование

Результат – модель данных

Задачи проектирования:

- Оценка полноты результатов анализа
- Согласование с заказчиком
- Определение архитектуры системы
- Используемые интерфейсы и стороннее ПО
- Архитектура баз данных
- Определение средств разработки
- Определение требований безопасности

Реализация

Тонкие моменты:

- Проектировщик определяет что делать, программист – как
- Проектировщик на этапе реализации выполняет функции ходячего справочника
- Желательно тесное взаимодействие с группой тестирования
- Координация пользовательских интерфейсов с заказчиком

Тестирование

- Модульное тестирование
- Интеграционное тестирование
- Системное тестирование
- Приемосдаточные испытания

Обязательно проверить реакцию системы на отказы

Полезно использовать генераторы тестовых данных

Интеграция

Основные этапы

- Портирование информации
- Накопление информации
- Выход на проектную мощность

Эксплуатация и техническая поддержка

Финальный этап: работа формально завершена, в идеале – стороны довольны друг другом

Документ технической приемки: определяются необходимые ресурсы для работы системы (железо и персонал), а так же условия нарушения эксплуатации продукта и ответственности сторон

Задачи службы поддержки – помощь заказчику в работе с системой и исправление обнаруженных ошибок.