Программная инженерия
Этапы жизненного цикла ПО

Компонентно-ориентированный подход

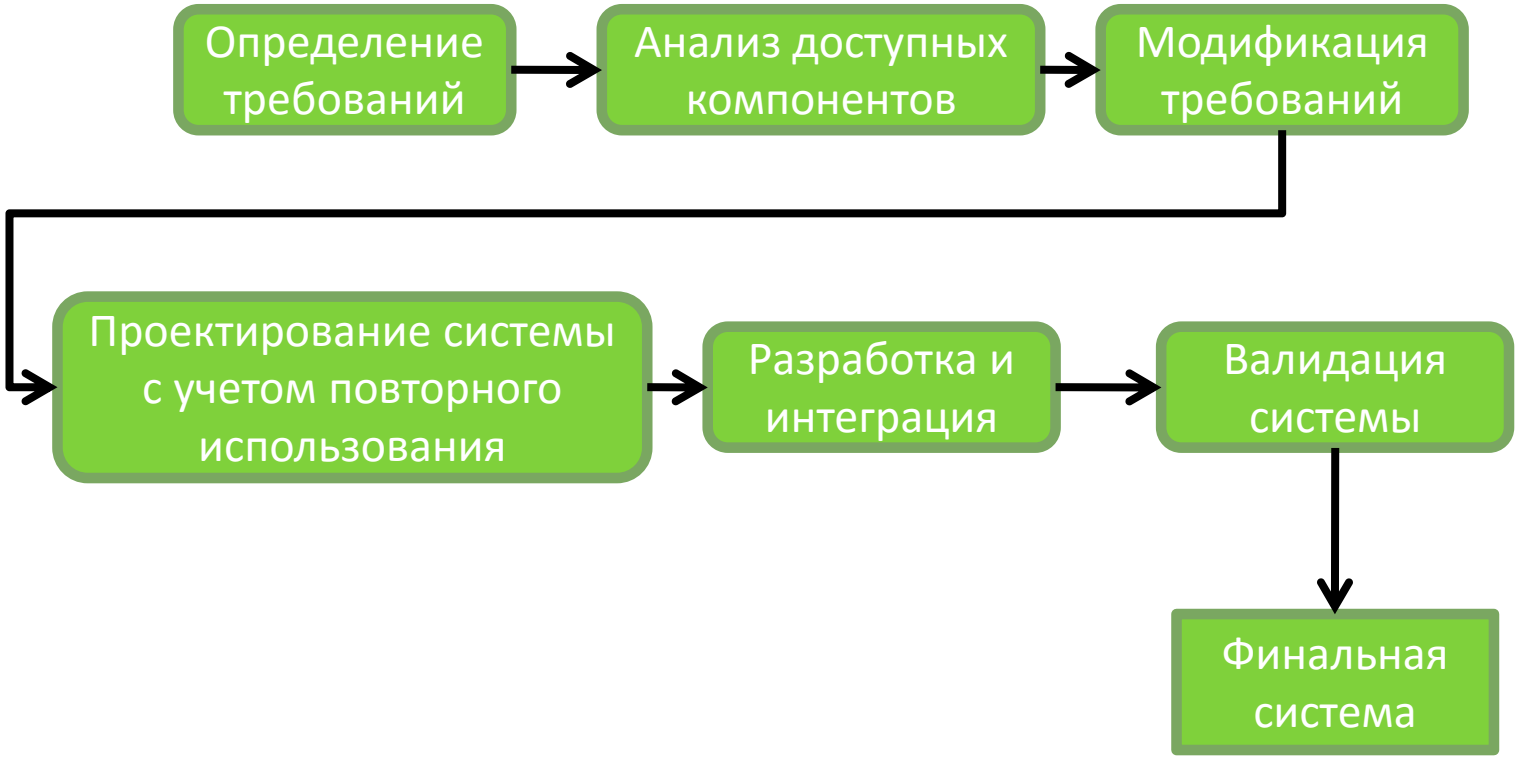
Компонентно-ориентированная разработка

- Компонентно-ориентированная разработка основывается на формальном закреплении повторного использования кода.
- **Программный компонент** – это автономный элемент программного обеспечения, предназначенный для *многократного использования*, который может распространяться для использования в других программах в виде скомпилированного кода.

Компонентно-ориентированная разработка

- Компонентно-ориентированный подход – развитие объектно-ориентированного. Создан для проектирования и реализации *крупных и распределенных программных систем (корпоративных приложений)*
- ***С точки зрения КОП*** программная система – это набор компонентов с четко определенным интерфейсом.
  - Изменения в систему вносятся путем создания новых компонентов или изменения старых.
  - **Наследование реализации запрещено. Наследуется только интерфейс.**

Компонентно-ориентированная разработка



Достоинства и недостатки компонентно-ориентированной разработки

- Достоинства
  - Уменьшается объем ПО, которое необходимо разработать => уменьшается цена и риски.
  - Уменьшается время разработки
- Недостатки
  - Компромиссы при выработке требований могут привести к тому, что реальные требования пользователей не будут учтены.
  - Контроль над системой может быть потерян при появлении новых версий используемых компонентов

Формализм в процессе разработки ПО

Сколько формализма необходимо?

- С 1970 по 1995 считалось, что чем тщательней оформлена документация, тем лучше
- Далее перешли на «компактность» документации: диаграммы и схемы
- Но может быть лучше пусть будет специальный человек, который знает всю документацию и может объяснить?

Факторы, влияющие на оптимальный уровень формализма

- **Масштаб проекта:** чем крупнее проект, тем более формальный подход необходим.
- **Критичность проекта:** если цена ошибки высока, необходимо применять более формальные модели разработки.
- **Распределение участников:** чем компактнее расположены участники разработки, тем менее формально можно подходить к разработке системы.
- **Новизна проекта:** чем более новые технологии и задачи стоят перед командой, тем больше нужно формализма.
- **Требования заказчика:** если заказчик – гос. учреждение, то он, скорее всего, будет требовать более формальный подход.
- **Ожидаемая долговечность проекта:** чем дольше предполагается срок жизни системы, тем более формальный подход к разработке необходимо применить.

Резюме

- ***Программная инженерия*** – это дисциплина, отражающая все грани разработки программного продукта рамках существующих *организационных*, *финансовых* и *временных* ***ограничений.***
- Качественный программный продукт требует в 10 раз больше трудозатрат чем программа с тем же функционалом

Резюме

- ***Процесс разработки ПО (жизненный цикл ПО)*** – это набор действий и связанных с ними результатов, направленных на разработку и/или развитие программного продукта**.**

- Идеального процесса разработки не существует!

Резюме

- Можно выделить 3 класса моделей разработки ПО:
  - Водопадная модель:
    - самая первая;
    - самая формальная;
  - Модель поэтапной разработки:
    - самая гибкая;
    - результат – после первой итерации;
  - Компонентно-ориентированная модель:
    - повторное использование кода;
    - ускоренная разработка.

Этапы жизненного цикла ПО

Этапы разработки ПО

Нету «Православного» деления на этапы разработки ПО:

Sommerville:

- Постановка задачи
- Разработка
- Валидация
- Развитие и поддержка

Unified Process:

- Начало
- Уточнение
- Построение
- Внедрение

Спольски:

- Требования
- Архитектура
- Конструирование
- Тестирование
- Внедрение

Относительная стоимость исправления ошибки

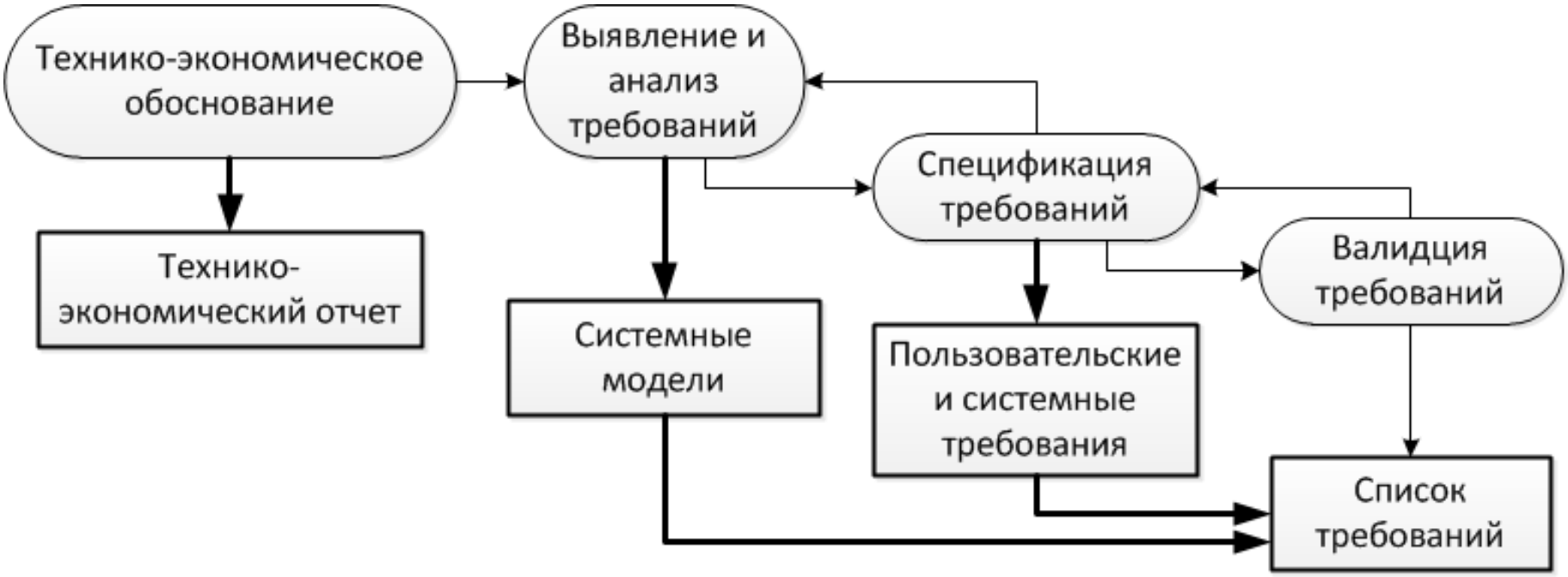
1 Постановка задачи

- *Постановка задачи* – это процесс определения **набора сервисов**, которые должна предоставлять система, а также определения **ограничений**, в рамках которых система будет разрабатываться и исполняться.

1 Фазы постановки задачи

1. Технико-экономическое обоснование
2. Выявление и анализ требований
3. Спецификация требований
4. Валидация требований

1 Процесс постановки задачи



2 Процесс разработки ПО

- Процесс разработки ПО состоит из двух действий:
  - **Проектирование** – описание структуры ПО, моделей данных, интерфейсов компонентов, алгоритмов;
  - **Реализация** – преобразование спецификации разрабатываемой системы в готовый программный продукт.

2 Модель процесса проектирования

2 Реализация

- Процесс программирования – это индивидуальная деятельность, которая не может быть описана определенным процессом.
- Не смотря на это, в компании могут быть выработаны стандарты кодирования:
  - именование переменных и методов;
  - форматирование исходного кода; методы комментирования и документирования; процесс управления версиями и т.п.)

3 Процесс валидации

- Процесс валидации (верификации и валидации) должен показать, что разработанная система соответствует спецификации и желаниям пользователей системы.
- Основной объем затрат в процессе валидации приходится на тестирование системы

Но, прежде всего, приведем два главных закона теории тестирования программных продуктов:

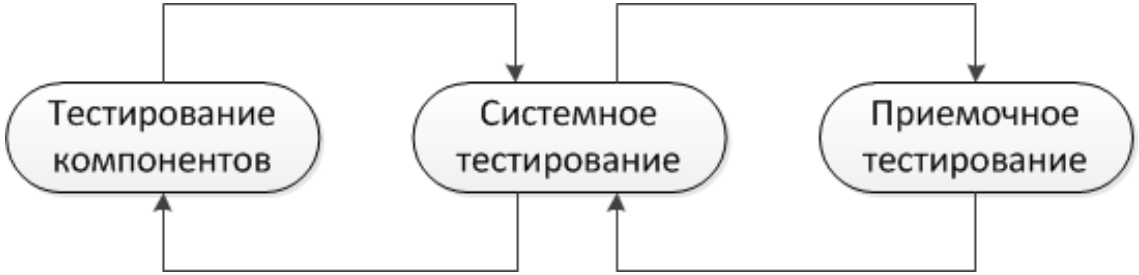
- Невозможно отыскать абсолютно все ошибки в программном продукте. Ошибки остаются всегда.
- Построение исчерпывающего входного теста невозможно.

3 Тестирование

- **Тестирование** - это процесс выполнения системы или компонента, при определенных тестировщиком условиях, для наблюдения и для оценки некоторых аспектов работы системы либо компонента.
- В процессе тестирования производится анализ ПО для выявления *несоответствия* между существующими особенностями выполнения ПО и требованиями к ПО (выявления *дефектов*).
- **Тестовый случай –** это набор входных данных, условий выполнения системы и ожидаемых результатов, разработанных для определенной цели, как то для оценки определенного пути выполнения приложения или для проверки соответствия определенному требованию.

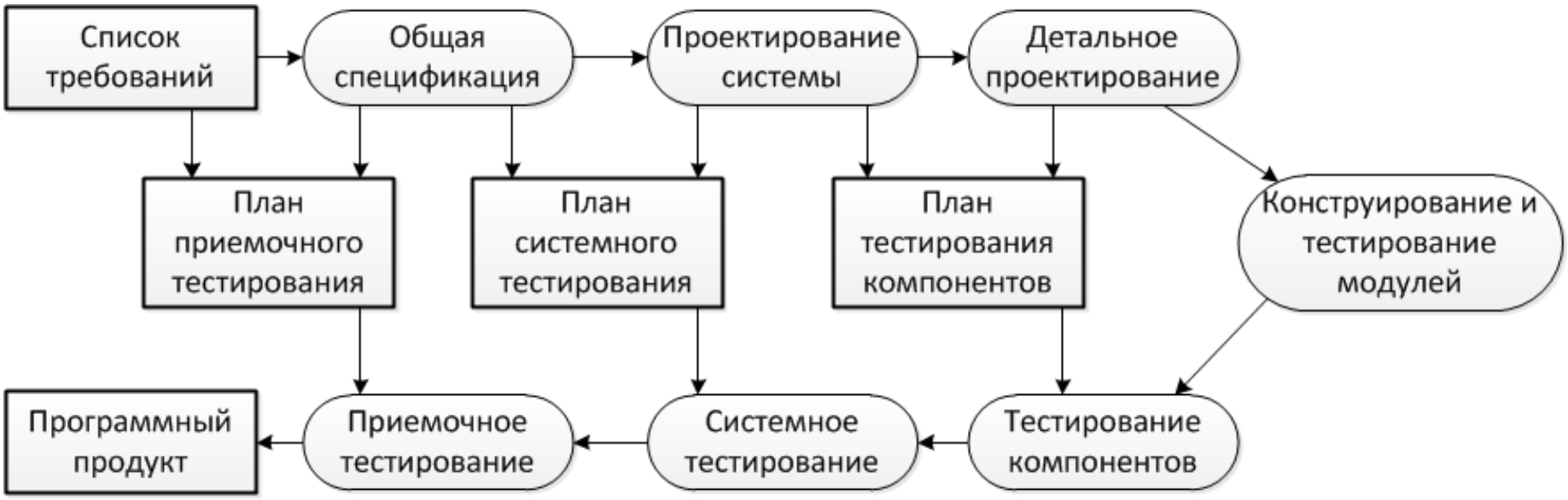
IEEE Standard 610-90 (Standard Glossary of Software Engineering Terminology)

3 Процесс тестирования



- **Компонентное тестирование (юнит-тестирование):** тестируются отдельные компоненты системы для определения корректности их работы. Каждый компонент тестируется независимо от других. Компонентами могут быть отдельные функции, классы, библиотеки.
- **Системное тестирование (интеграционное тестирование)**: компоненты объединяются в общую систему и она тестируется как единое целое. В результате выявляются ошибки взаимодействия компонентов и проблемы определения интерфейсов.
- **Приемочное тестирование**: финальная стадия процесса тестирования. Система тестируется на данных, предоставленных пользователем (покупателем). Могут выявиться ошибки интерпретации данных и требований к системе, т.к. система может вести себя по-другому на реальных данных.

3 Внедрение тестирования в процесс разработки ПО



4 Развитие и поддержка ПО

- Важное отличие ПО от аппаратных платформ и других объектов реального мира – относительная простота изменения и развития
- В связи с этим, процент «абсолютно новых» программных систем очень невысок. Большинство – развитие предыдущих, уже вышедших программных решений.