ЗАДАНИЕ №4

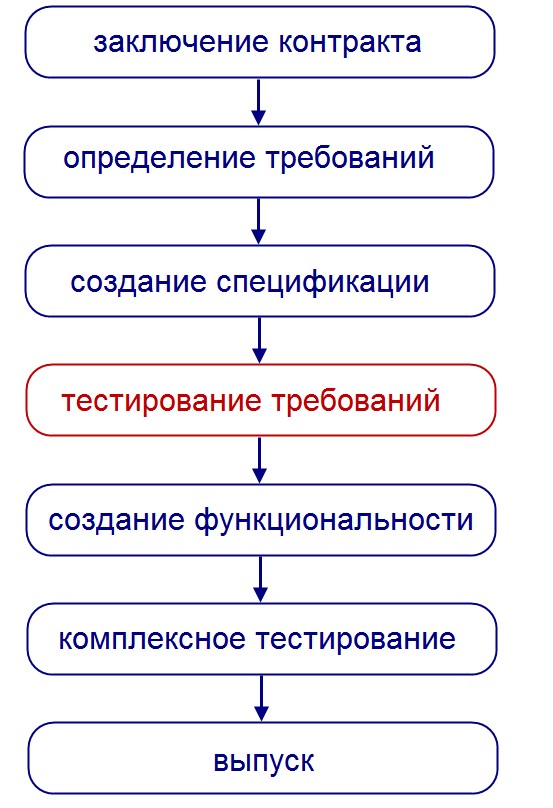
Тестирование требований

*Цель работы:* изучить критерии качества требований, выполнить тестирование спецификации требований.

# *Теоретические сведения*

Требования обеспечивают основу для последующего тестирования – процесса анализа программного средства на предмет соответствия зафиксированным требованиям и/или ожиданиям и нуждам пользователя или заказчика. От качества сформированных требований зависит качество программного обеспечения, т.к. требования к программному продукту являются базой для генерации тестов и обнаружения дефектов, представляющих собой любое отклонение от спецификации.

В связи с вышеизложенным при разработке программного обеспечения тестирование необходимо выполнять уже на стадии разработки спецификации (рисунок 1).



# Рисунок 1 – Жизненный цикл проекта

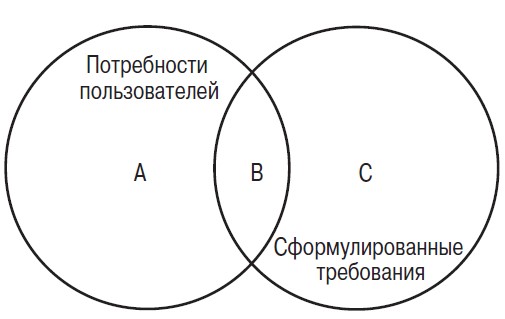
Тестирование требований является важным этапом разработки продукта, так как это приводит к:

1. снижению риска получить продукт, не отвечающий ожиданиям заказчика или нуждам конечных пользователей;
2. снижению затрат на разработку и тестирование продукта;
3. сокращению сроков сдачи готового продукта;
4. налаживанию взаимопонимания при создании продукта между всеми вовлеченными исполнителями.

Выделяют 9 критериев качества требований:

1. корректность;
2. атомарность;
3. недвусмысленность;
4. полнота;
5. непротиворечивость;
6. упорядоченность по важности и стабильности;
7. проверяемость;
8. модифицируемость;
9. трассируемость (прослеживаемость);

***Корректные требования.*** Набор требований к программному обеспечению является корректным тогда и только тогда, когда каждое требование, сформулированное в нем, представляет нечто, требуемое от создаваемой системы. Данная формулировка отражена на рисунке 2.



# Рисунок 2 – Множества потребностей и требований

Если левый круг (*область А*) представляет множество потребностей пользователя, а правый (*область С*) — требования, то корректные требования будут находиться в области пересечения кругов, *область В*.

**Атомарность требований.** Требование является атомарным, если его нельзя разбить на отдельные требования без потери завершённости, и оно описывает одну и только одну ситуацию.

***Недвусмысленные требования.*** Требование является недвусмысленным тогда и только тогда, когда его можно однозначно интерпретировать. Хотя главным свойством любого требования по праву считается корректность, неоднозначность зачастую представляет собой более сложную проблему. Если формулировка требований может по-разному интерпретироваться разработчиками, пользователями и другими участниками проекта, вполне может оказаться, что построенная система будет полностью отличаться от того, что представлял себе пользователь.

***Полнота набора требований.*** Набор требований является полным тогда и только тогда, когда он описывает все важные требования, интересующие пользователя, в том числе требования, связанные с функциональными возможностями, производительностью, ограничениями проектирования, атрибутами или внешними интерфейсами. Полный набор требований должен также задавать требуемый ответ программы на всевозможные классы ввода — как правильные, так и неправильные — во всевозможных ситуациях. Помимо этого, он должен содержать полные ссылки и подписи всех рисунков, таблиц и диаграмм набора требований, а также определения всех терминов и единиц измерения.

***Непротиворечивость набора требований.*** Множество требований является внутренне непротиворечивым, когда ни одно его подмножество, состоящее из отдельных требований, не противоречит другим подмножествам. Конфликты могут иметь различную форму и проявляться на различных уровнях детализации; если набор требований был написан достаточно формально и поддерживается соответствующими автоматическими средствами, конфликт иногда удается обнаружить посредством механического анализа. Но, скорее всего, разработчикам вместе с другими участниками проекта придется провести проверку множества требований вручную, чтобы удалить все потенциальные конфликты.

***Упорядочение требований по их важности и стабильности.*** В высококачественном наборе требований разработчики, клиенты и другие заинтересованные лица упорядочивают отдельные требования по их важности для клиента и стабильности. Этот процесс упорядочения особенно важен для управления масштабом. Если ресурсы недостаточны, чтобы в пределах выделенного времени и бюджета реализовать все требования, очень полезно знать, какие требования являются не столь уж обязательными, а какие пользователь считает критическими.

***Проверяемые требования.*** Требование в целом является проверяемым, когда каждое из составляющих его элементарных требований является проверяемы, т.е. когда можно протестировать каждое из них и выяснить, действительно ли они выполняются.

***Модифицируемый набор требований.*** Множество требований является модифицируемым, когда его структура и стиль таковы, что любое изменение требований можно произвести просто, полно и согласованно, не нарушая существующей структуры и стиля всего множества. Для этого требуется, чтобы пакет требований имел минимальную избыточность и был хорошо организован, с соответствующим содержанием, индексом и возможностью перекрестных ссылок.

***Трассируемые требования.*** Требование в целом является трассируемым, когда ясно происхождение каждого из составляющих его элементарных требований и существует механизм, который делает возможным обращение к этому требованию при дальнейших действиях по разработке. На практике это обычно означает, что каждое требование имеет уникальный номер или идентификатор. Возможность трассировки имеет огромное значение. Разработчики могут использовать ее как для достижения лучшего понимания проекта, так и для обеспечения более высокой степени уверенности, что все требования выполняются данной реализацией.

**Существуют различные методы тестирования требований:**

1. Метод просмотра (универсальный метод, выполняется бизнес-аналитиком или тестировщиком):
   * Ознакомление с требованиями.
   * Проверка требований по критериям качества.
   * Оформление дефектов. - Оформление отчета.
2. Метод экспертизы (выполняется при участии команды из бизнес-аналитиков, представителей заказчика, разработчиков, лояльных пользователей, тестировщиков):
   * Планирование.
   * Обзорная встреча.
   * Подготовка.
   * Совещание.
   * Переработка.
   * Завершающий этап.
3. Метод составления вариантов тестирования (выполняется тестировщиком). Варианты тестирования занимают промежуточную позицию между User Case и Test Case, помимо использования для тестирования требований в дальнейшем легко расширяются до Test Cases и составляют основу тестовой документации.

## Порядок выполнения работы

1. Получить задание у преподавателя.
2. Протестировать спецификацию методом просмотра.
3. Оформить отчет и защитить работу. В отчете перечислить найденные дефекты (в виде цитаты и необходимых пояснений), указать какому критерию качества не удовлетворяет, предложить корректный вариант.

## Содержание отчета

1. Отчет по тестированию спецификации. В отчете перечислить найденные дефекты

## Контрольные вопросы

1. Как выглядит жизненный цикл проекта?
2. Какие выделяют критерии качества?
3. Какие требования считаются проверяемыми?
4. Какие требования считаются модифицируемыми?
5. Какие требования считаются корректными?
6. Какие требования считаются недвусмысленными?
7. Какие требования считаются полными?
8. Какие требования считаются непротиворечивыми?
9. Какие требования считаются упорядоченными по важности и стабильности?
10. Какие требования считаются трассируемыми?
11. Какие существуют методы тестирования требований?
12. Какие этапы включает в себя метод просмотра при тестировании требований?