# Министерство образования Республики Беларусь

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Могилевский государственный политехнический колледж»

# Домашняя контрольная работа №1

По дисциплине: «**Стандартизация и сертификация программного обеспечения**»

Группа ПО-455

## Выполнил О. И. Любаль

Шифр 14

# 2022

**35 Охарактеризуйте передовые технологии в тестировании (автоматизация тестирования)**

## 1. Цель

Каждый программный продукт должен выполнять одну или несколько ключевых задач. От приложения с геокартами мы ожидаем точной ориентации в пространстве, от сайта интернет-магазина ― корректного поиска товаров по заданным параметрам и т. д. Но те же программные продукты мы можем протестировать и с точки зрения дизайна.

Таким образом, анализ ПО с позиции его ключевых или вспомогательных функций определяет тип тестирования:

* **Функциональное**
* **Нефункциональное**

Функциональное тестирование направлено на проверку того, какие функции ПО реализованы, и того, насколько верно они реализованы. Краткая диаграмма видов тестирования представлена на рисунке 1.

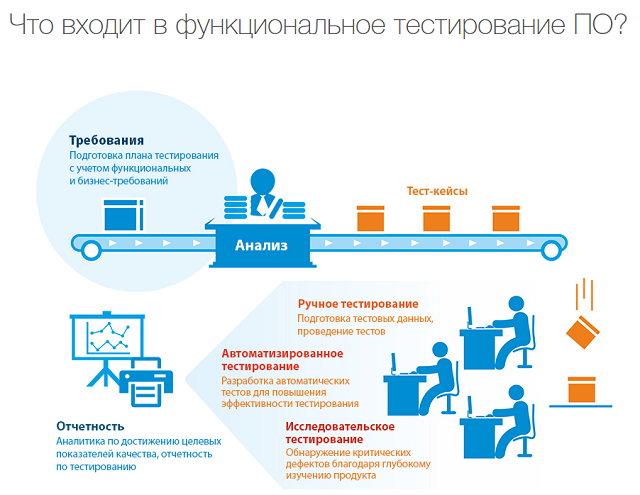


Рисунок 1. Краткая диаграмма видов тестирования.

Нефункциональное – проверка корректности работы нефункциональных требований. Оценивается, КАК программный продукт работает. Эта проверка включает в себя следующие виды:

* **Тестирование производительности** – работа ПО под определённой нагрузкой.
* **Тестирование пользовательского интерфейса** – удобство пользователя при взаимодействии с разными параметрами интерфейса (кнопки, цвета, выравнивание и т. д.).
* **Тестирование UX** – правильность логики использования программного продукта.
* **Тестирование защищенности** – определение безопасности ПО: защищено ли оно от атак хакеров, несанкционированного доступа к данным и т. д.
* **Инсталляционное тестирование** – оценка вероятности возникновения проблем при установке, удалении, а также обновлении ПО.
* **Тестирование совместимости** – тестирование работы программного продукта в определённом окружении.
* **Тестирование надежности** – работа программы при длительной средней ожидаемой нагрузке.
* **Тестирование локализации** –оценка правильности версии программного продукта (языковой и культурный аспекты).

## 2. Степень автоматизации

В зависимости от того, используют ли тестировщики дополнительные программные средства для тестирования приложений или программ, тестирование бывает:

* **Мануальное (ручное)** – без использования дополнительных программных средств, т. е. «вручную».
* **Автоматизированное** – с использованием программных средств (более детально в описании курса по автоматизации тестирования ПО).

Каждый из подходов имеет свои преимущества и недостатки. Ручное тестирование проще освоить, оно широко применяется на проектах всех типов, но мануальные проверки отличаются монотонностью. А вот написание тестов даёт больше возможностей для творческой реализации, но автоматизация требует базовых навыков программирования.

## 3. Позитивность сценария

Этот подход определяет поведение системы в привычных и экстремальных условиях.

* **Позитивная проверка** – оценка ожидаемого поведения. Это тестирование проводится в первую очередь, ведь позволяет определить корректность работы программы.
* **Негативная** – определение устойчивости системы в нестандартной ситуации. Например, неожиданный сценарий взаимодействия пользователя с интерфейсом.

Эти типы тестирования нередко проводятся параллельно. Ведь работая над некоторой функциональностью, тестировщику проще оценить её поведение и в стандартных, и в нестандартных условиях.

## 4. Доступ к коду программного продукта

В процессе тестирования инженер может работать с ПО, не обращаясь к его коду, а может определить правильность работы, взглянув на код. По доступу к коду программного продукта тестирование делится на:

* **Тестирование «белого ящика»** – с доступом к коду.
* **Тестирование «черного ящика»** – без доступа к коду продукта.
* **Тестирование «серого ящика»** – на основе ограниченного знания внутренней структуры ПО. Часто говорят, что это смесь тестирования «белого ящика» и «чёрного ящика», но это в корне неверно. В данном случае тестировщик не работает с кодом программного продукта, но он знаком с внутренней структурой программы и взаимодействием между компонентами.

Проверка программного продукта по каждому из сценариев требует достаточно глубоких знаний. К примеру, об особенностях тестирования «чёрного ящика» в своей книге подробно рассказал Борис Бейзер. Это фундаментальная работа, с которой полезно ознакомиться каждому на старте работы в QA.

## 5. Уровень

Этот пункт определяет объект тестирования.

* **Модульное / юнит-тестирование** – проверка корректной работы отдельных единиц ПО, модулей. Этот вид тестирования могут выполнять сами разработчики.
* **Интеграционное тестирование** – проверка взаимодействия между несколькими единицами ПО.
* **Системное** – проверка работы приложения целиком.
* **Приёмочное** – оценка соответствия заявленным требованиям к программному продукту.

Переход на каждую новую ступень – движение от микроуровня к макро. Это важный этап тестирования, ведь безошибочно написанные модули могут просто не работать вместе.

## 6. Исполнитель

От объекта тестирования движемся к его субъекту. Вы могли слышать об альфа- и бета-тестировании. А поучаствовать в одном из них можно, даже не будучи тестировщиком. Итак, по исполнителю тестирование делится на:

* **Альфа-тестирование** – проверка программного продукта на поздней стадии разработки. Проводится разработчиками или тестировщиками.
* **Бета-тестирование** – оценка ПО перед выходом на рынок в фокус-группе или добровольцами. Отзывы собираются, анализируются и учитываются при внесении правок.

## 7. Формальность

Этот пункт определяет подготовленность тестировщика перед началом проверки.

* **Тестирование по тестам** – использование написанных заранее тест-кейсов.
* **Исследовательское тестирование** – одновременная разработка тестов и их использование.
* **Свободное тестирование** – проверка качества без разработки тестов и написания документации. Основывается на интуиции и опыте тестировщика.

Начинающие тестировщики редко работают на свободном уровне. А вот опытные QA-специалисты могут позволить себе проверку без дополнительной подготовки. Мастерство растёт со временем, как и оплата труда тестировщика. О том, сколько получают инженеры, [читайте](https://qa-academy.by/qaacademy/news/skolko-zarabatyvayut-testirovshhiki/) в нашем блоге.

## 8. Важность

* **Дымовое тестирование** – проверка самой важной функциональности программного продукта.
* **Тестирование критического пути** – проверка функциональности, используемой типичными пользователями в повседневной деятельности.
* **Расширенное тестирование** – проверка всей заявленной функциональности.

QA-область очень многообразна. Помимо отличий в технологии оценки качества, тип тестирования может отличаться индустрией или видом программного продукта. К примеру, начинающий тестировщик может выбрать для себя специализацию:

* тестирование мобильных или десктопных приложений;
* банкинг;
* социальные сети;
* [игры](https://qa-academy.by/programmy-i-kursy/trening-testirovanie-bezopasnosti-po/);
* и другое.

**13 Опишите виды программ по ГОСТ 19781‑90**

 Таблица 1

|  |  |
| --- | --- |
| Виды программ | |
| Термин | Определение |
| 4. **Системная программа**  System program | Программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования в процессе выполнения прикладных программ |
| 5. **Управляющая программа**  Control program | Системная программа, реализующая набор функций управления, в который включают управление ресурсами и взаимодействие с внешней средой системы обработки информации, восстановление работы системы после проявления неисправностей в технических средствах |
| 6. **Супервизор**  Supervisor | Часть управляющей программы, координирующая распределение ресурсов системы обработки информации |
| 7. **Прикладная программа**  Application program | Программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации |
| 8. **Программа обслуживания**  Utility program | Программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу системы обработки информации |
| 9. **Абсолютная программа**  Non-relocatable program | Программа на машинном языке, выполнение которой зависит от ее местоположения в оперативной памяти |
| 10. **Переместимая программа**  Relocatable program | Программа на машинном языке, выполнение которой не зависит от ее местоположения в оперативной памяти |
| 11. **Реентерабельная программа**  Reenterable program | Программа, один и тот же экземпляр которой в оперативной памяти способен выполняться многократно, причем так, что каждое выполнение может начинаться в любой момент по отношению к другому выполнению |
| 12. **Мобильная программа**  Portable program | Программа, которая написана для ЭВМ одной архитектуры, но может исполняться в системах обработки информации с другими архитектурами без доработки или при условии ее доработки, трудоемкость которой незначительна по сравнению с разработкой новой программы |
| 13. **Драйвер**  Driver | Программа, предназначенная для управления работой периферийных устройств, обычно в мини- и микроЭВМ |
| 14. **Подпрограмма**  Subprogram | Программа, являющаяся частью другой программы и удовлетворяющая требованиям языка программирования к структуре программы |
| 15. **Программный модуль**  Program module | Программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память |
| 16. **Исходный модуль**  Source module | Программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции |
| 17. **Объектный модуль**  Object module | Программный модуль, получаемый в результате компиляции исходного модуля. |
|  | Примечание. Объектный модуль обычно полностью готов к редактированию связей |
| 18. **Загрузочный модуль**  Load module | Программный модуль, представленный в форме, пригодной для загрузки в основную память для выполнения |
| 19. **Макроопределение**  Macrodefinition | Программа, под управлением которой макрогенератор порождает макрорасширения макрокоманд |
| 20. **Рекурсивная подпрограмма**  Recursive subroutine | Подпрограмма, которая может обращаться к себе самой |

**51 Опишите отдел стандартизации (отдел управления качеством продукции) на предприятии, где Вы работаете. Назовите его структуру и задачи. Укажите перечень ТНПА (номер и название), используемых при выпуске продукции на предприятии, где Вы работаете**

**Основные задачи:**  
• разработка, внедрение, сертификация системы менеджмента качества проектирования сельскохозяйственной техники в соответствии с требованиями СТБ ИСО 9001–2015;  
• управление нормативной документацией организации;  
• осуществление нормоконтроля конструкторской документации и нормативных документов;  
• осуществление организационного и методического руководства патентно-лицензионной, изобретательской и рационализаторской работой.

**52 Разработайте техническое задание в соответствии с требованиями ЕСПД на программный модуль, разработанный Вами в курсовом проекте по учебной дисциплине «Основы алгоритмизации и программирования»**

**Список используемых источников**

1 Осипенко, Н. Б. Стандартизация и сертификация программного обеспечения: тексты лекций для студентов математических специальностей / Н.Б. Осипенко - Гомель: ГТУ им. Ф. Скорины, 2012. -155 с.

2 Шандриков, Л. С. Стандартизации и сертификация программного обеспечения: учебное пособие / А.С. Шандриков. - Минск: РИПО. 2014. – 304 с.: ил.

3 https://kiosker.by/publication/303-standartizatsija-i-sertifikatsija-programmnogo-obespechenija/issue/1003138/