

## Общая информация

В рамках курсовой работы нужно реализовать три вида тестов – модульные, интеграционные и системные. Время на выполнение каждого вида тестирования – 2 занятия (месяц). В случае, если разработанный в осеннем семестре программный продукт не подпадает под минимальные требования по количеству тестов или данный вид тестов неприменим/нецелесообразен, то предлагается работать над любым программным продуктом с открытым исходным кодом. Примеры: kafka, zookeeper, grafana, elasticsearch, consul, jenkins, atom, Eclipse Che, notepad++ или любой другой популярный продукт на выбранном вами языке.

## Модульное тестирование

Необходимо выполнить модульное тестирование разработанного программного продукта. Кодовая база всего продукта должна быть покрыта тестами на 80% и более, но не менее 20 тестов на каждого члена команды.

Обязательные требования:

1. Тестирование должно производиться автоматически при сборке проекта тем сборщиком, который используется для формирования исполняемого файла (cmake, gradle, maven, ant, etc.) Фреймворки для написания юнит-тестов: junit, testNG, XUnit.Net, Nunit – любые, подходящие для языка вашего проекта и формирующие необходимую отчётность.
2. Применение нескольких техник тест-дизайна: классы эквивалентности, граничные условия, попарное тестирование, etc.

Отчёт по модульному тестированию должен содержать:

1. Описание выполненной работы, использованных инструментов, применённых техник тест-дизайна.
2. Отчёт о прохождении тестов с результатами и оценкой покрытия кода тестами.
3. Описание процедуры расширения тестового набора на примере добавления нового блока кода, алгоритма, метода.

## Интеграционное тестирование

Интеграционное тестирование предполагает наличие нескольких модулей или, если приложение построено в соответствии с микросервисной архитектурой, микросервисов разработанного приложения. В качестве модулей могут быть функциональные части (например: модуль авторизации\аутентификации, модуль взаимодействия с пользователем, модуль интеграции со сторонним сервисом).

Необходимо выполнить интеграционное тестирование нескольких модулей в соответствии с предварительно описанными сценариями тестирования. Каждый сценарий тестирования должен представлять собой некоторый законченный вариант использования системы (кейс) пользователя.

Интеграционное тестирование предполагает командную работу над программным продуктом, поэтому необходимо проводить его с использованием одного из инструментов CI, доступных на рынке: Gitlab, Jenkins, Bamboo, TeamCity, etc.

Минимальное количество сценариев - 10.

Обязательные требования:

1. Предварительное формирование документа, описывающего тестовые сценарии (утверждение сценариев у преподавателя).
2. Применение заглушек (mock-сервисов или mock-объектов) для изоляции от окружения и внешних сервисов или для ускорения прохождения тестов.
3. Рассмотрение негативных сценариев тестирования.
4. Поднятие сервера непрерывной интеграции и запуск задачи по интеграционному тестированию по временному триггеру или по событию изменения кода приложения\интеграционных тестов.

Отчёт по интеграционному тестированию должен содержать:

1. Отчёт о выполненной работе, использованных инструментах, применённых техниках тест-дизайна.
2. Тест-план со словесным описанием тестовых сценариев, которые планируется реализовать в интеграционном тестировании (необходимо предварительно утвердить у преподавателя)
3. Отчёт о прохождении тестов с результатами на сервере непрерывной интеграции.
4. Описание процедуры расширения тестового набора на примере добавления новой функциональной части (или модуля)

## Системное/End-to-End тестирование

На данном уровне необходимо протестировать готовый продукт по бизнес-требованиям, сформированным перед началом непосредственного проектирования программного продукта. В качестве тестовых сценариев выбираются основные сценарии использования ПО в полностью рабочем окружении. Целесообразно использование инструментов веб-тестирования или UI-тестирования, таких как selenium, puppeteer и т. д.

Обязательные требования:

1. Предварительное формирование документа, описывающего тестовые сценарии (утверждение сценариев у преподавателя)
2. Поднятие сервера непрерывной интеграции и запуск задачи системного тестирования по временному триггеру или в ручном режиме (кроме тестирования производительности)

Отчёт по системному тестированию должен содержать:

1. Отчёт о выполненной работе, использованных инструментах.
2. Тест-план со словесным описанием тестовых сценариев
3. Отчёт о прохождении тестов с результатами на сервере непрерывной интеграции.

## Общие критерии оценки

60–70 баллов – удовлетворительно

71–80 баллов – хорошо

81+ баллов – отлично

За сорванные сроки, итоговый балл уменьшается на 10 баллов/месяц, за каждый вид тестов. За не сделанный вид тестов (учитываются только обязательные) количество баллов уменьшается на 20.

## Модульное тестирование

Минимальное количество тестов – 20 на каждого члена команды.

	% покрытия кода тестами	количество баллов
1	60 - 70	10
2	71 - 80	15
3	81+	25

## Интеграционное тестирование

Минимальное количество сценариев - 10

Каждый сценарий оценивается в 2 балла, максимум баллов – 40

## Системное/End-to-End тестирование

Минимальное количество сценариев – 10.

Каждый сценарий оценивается в 2 балла, максимум баллов - 40

В рамках системного тестирования, в случае применимости, возможно произвести тестирование производительности:

1. Нагрузочное тестирование [load testing] (+5 баллов)
2. Тестирование стабильности [stability testing] (+5 баллов)
3. Тестирование восстановления [recovery testing] (+5 баллов)
4. Объёмное тестирование [volume testing] (+5 баллов)