Уровни тестирования

[**Уровни тестирования**](https://dist.belstu.by/mod/assign/view.php?id=57433) – это то, насколько глубоко мы тестируем систему/веб или мобильное приложение. Любое приложение состоит из фич, которые объединяются в компененты, которые в свою очередь объединяются в приложение.

 Например, социальная сеть. Это приложение состоит из разных компонентов (профиль человека; стена, куда он делает посты; альбом с фотографиями, хранилище с музыкой; список фрэндов или фолловеров; и т.д.). Эти компоненты можно разбить на фичи. Например, профиль человека состоит из фичи «добавление/редактирование/удаление аватарки», информация о пользователе, ссылки на сообщества, на которые он подписан, и т.д.

Когда компонент попадает к нам на тестирование, то сначала мы делаем компонентное тестирование. Т.е. проверяем работоспособность и корректность UI самого компонента. Например, если мы говорим про аватарку, то проверками будет:

·   Загрузить аватарку

·   Удалить аватарку

·   Изменить аватарку

·   Загрузить разные типы файлов (jpg, png, bmp, exe)

·   Загрузить большой файл

·   Загрузить маленький файл

·   И т.д.

Когда мы полностью проверили весь компонент, мы переходим на следующий уровень тестирования: интеграционное тестирование. Мы проверяем, как этот компонент взаимодействует с другими компонентами. Например, как аватарка отображается на стене, когда человек запостил что-то, или проверяем, как в комментариях отображается его аватарка. Или идем в альбом, выбираем фото и делаем его аватаркой. Потом идем в профиль и смотрим, что аватарка корректно поменялась.

Надеюсь, что смысл и разница этих двух уровней понятна. Теперь про само задание:

В социальное сети ВКонтакте есть компонент «Фотографии». Необходимо написать проверки в виде чеклиста для компонентного и интеграционного тестирования.

**Оформление:**

Название файла должно быть: levels\_of\_tests\_<имя>\_<фамилия>.xlsx

В файле должно быть 2 закладки:

1.       Компонентные тесты

2.      Интеграционные тесты

Тесты должны быть пронумерованы. Их нужно написать как можно больше и стремиться покрыть весь функционал проверками.