Директорию с ее вложением:

**lib/core/repositories/**

необходимо положить в:

**lib/data/,** а папку «**repositories**» и переименовать на «**mock\_tests\_service**»

так как генератор данных по смыслу больше подходит к слою **data**

Директорию с ее вложением:

**lib/core/logger/**

необходимо положить в:

**lib/utils/**

так как оптимизация логирования вспомогательная задача и лучше подходит по смыслу для **utils (утилиты)**

Папку «**bloc**» что находится по адресу:

lib/core/bloc/

желательно переименовать на «**bloc\_observer**» так как ее содержимое слишком обобщенное и не дает ясного понимания о её содержимом.

Файл «**test\_repository.dart**» необходимо переименовать на «**base\_test\_repository.dart**» или лучше на «**base\_repository.dart**»

Директория со своим вложением:

**lib/domain/analytics\_item**

нигде в коде не учавствует. Если этот проект завершенный – такие избыточные/мертвые структуры нужно удалять.

Папку «**domain**» лучше переименовать в «**models**» - так как там хранятся модели данных и их интерфейсы.

В коде очень много встречается не реализованных методов возвращающих «**throw UnimplementedError()**;» если эти методы не нужны в проекте их нужно удалить (почистить интерфейсы или переструктурировать их) или реализовать их.

В директории **lib/presentation** надо-бы навести структурный порядок. Директории (**analytics\_screen**, **tests\_screen**, **settings\_screen**) сгрупировать в папку **tabs**. Папку «**drawing\_screen**» с вложением можно поместить в папку **tests\_screen**. Папку **base\_app** необходимо вынести на уровень выше и переименовать на **pressets\_app**, так как там происходят основные предварительные, глобальные настройки приложения.

В классе **\_MyAppState**, в методе «**initState()**» нарушен порядок вызова функций. Сначала надо вызвать **super.initState();**, а потом уже **\_initDependencies();.** согласно соглашению по оформлению кода

Так-же необходимо функцию **List<BlocProvider> \_initBlocs(BuildContext context)** разместить перед методом **Widget build(BuildContext context)**, а не после, чтобы не нарушать порядок ЖЦ **StatefulWidget**.

В файле «**lib/presentation/tests\_screen/test\_card\_list\_item.dart**» необходимо заменить виджет **TestCardTrailing** на метод **\_buildTestCardTrailing**, так как он не имеет собственного состояния. А также в методе **\_buildTestCardTrailing** необходимо выделить 2 подметода, так как **\_buildTestCardTrailing** слишком громоздкий для восприятия.

В файле «**lib/presentation/tests\_screen/test\_list\_item.dart**» в методе **ClipRRect \_buildTrailingIcon()** необходимо заменить возвращаемый тип **ClipRRect** на **Widget**. А также на строке 24 виджет **Row** преобразовать в метод **Widget \_buildItemCard(BuildContext context)**

В файле «**lib/presentation/tests\_screen/test\_list.dart**» в классе \_TestListState поле widgets переименовать на \_**listTestItems**, так как в приватном классе нет нужды держать публичные поля, а также поля должны иметь какой-то декларативный смысл. Метод **\_onTestPressed(TestSingleItem test)** необходимо реализовать перед методом **Widget build(BuildContext context)** согласно соглашению по оформлению кода

В файле «**lib/presentation/tests\_screen/tests\_screen.dart**» в классе **\_TestsScreenState** в методе **initState()** необходимо вызвать функцию **bloc.add(TestsLoad())**; после функции **super.initState()**;, чтобы не нарушать порядок ЖЦ **StatefulWidget**.