| Название | Наличие | Почему есть/пояснения |
| --- | --- | --- |
| В коде |  |  |
| Спагетти код | Нет | Правда тестирование кода есть не везде, а в остальном мой код вполне задокументирован и понятен на вид. |
| Лазанья код | Есть | Минимально, но это присутствует. Допустим если я захочу расширить модель добавив ещё название для столбца. То вьюхи как минимум многие используют обращение по именам. Их тоже придётся расширять и их функционал. Импорты тоже всегда будут присутствовать в коде. Как решить: поменьше импортов., Делать вьюхи для конкретной задачи (чтоб её не нужно было изменять), а лучше создавать новую с новым функционалом. Писать кода так чтоб не нужно было расширять у блока кода возможности, а добавлять новый блок кода для новых возможностей. |
| Слепая вера | Есть | Там где у меня парсится строка я слепо уверен в том что никто не введёт & в имени. Строка режется на список строк по символу &. Вроде логично что таких имён почти не будет. Но так как у меня сайт посвящён настолкам название D&D как раз и приводил к багу обработки строки. Решение написать алгоритм понимающий где имя, а где разделение. Я думаю что между знаками равно надо считать & и если их больше 1, то что обрезать строку в нужном месте. |
| Шифрокод | Нет | Стараюсь делать имена переменных, функций и тд с понятными названиями. |
| Жёсткое кодирование | Есть | Но как правило это находится в урлах и я уверен что эти урлы менять я не буду. Поэтому считаю что тут нет проблемы. Но при желании можно создать файл конфиг и туда их, но необходимости в этом я не вижу. |
| Мягкое кодирование | Нет | Так как я не запрашиваю все данные, то считаю тут нет такой проблемы. Стараюсь то что изменяется запросить, а то что не изменяется хардкодить. |
| Поток лавы | Нет | Самой ошибки визуально у меня нет, но проф тестов я не проводил так как на это времени просто нет и я больше тестирую руками работу. В настоящем проекте нужно много автоматических тестов делать чтоб выявлять много ситуаций с проблемами. |
| Магические числа | Нет | Если передаю число в параметр функции либо в виде переменной либо использую присвоение к имени параметра чтоб было ясно что это за число передал. Исключение делаю для функций типа range (так как там и так всё понятно). |
| В ООП |  |  |
| Боязнь размещать логику... | Нет | Не боюсь размещать методы в моделях или вообще там где они должны быть. |
| Божественный объект | Есть | Полностью нельзя от них избавиться в крупных приложениях. Тот же класс Engine много выполняет и RobotFramework тоже. Такие классы не стоит плодить. Так же им не стоит давать больше функций чем им нужно. |
| Полтергейст | Нет | По крайней мере я такой ошибки, которая возникает часто в потоках не нашёл визуально. При общении с бд нужен один поток. У нас один поэтому думаю нет повода для этой ошибки. |
| Сплошное одиночество | Нет | У меня в проекте он только один синглтон. Их не стоит плодить я и не пложу их. |
| Приватизация | Нет | У меня есть один приватный метод \_\_instance в классе синглтоне. Но там он не мешает так как его потомки не должны наследовать. |
| Питоновские |  |  |
| Функции с возвратом None | Нет | Не стоит допускать в код чтоб функция могла вернуть None буквально. Вместо этого лучше использовать raise. |
| Сложное создание dict | Нет | Нужно использовать генерацию словаря внутри {} вместо dict(). |
| Проверка операции | Есть | Пример где у меня прилетает название для json файла. Там нужна нормальная валидация данных, а вместо этого только if стоит. Если имя из формы не прилетит, то просто прилетит None и вобьется в название. Решение сделать валидацию с try и не допускать создание файлов с именами None. Сделал так потому что это временное решение и с переводом проекта на базы данных эти json формы не понадобится. Если я это не выпилю то сделаю нормальную валидацию данных там. |
| Отдельное присвоение... | Нет | Использую распаковку там где это нужно |
| Методологические |  |  |
| Программирование ctrl + v | Нет | Даже если я использую чужой код в качестве шаблона я стараюсь набирать код руками чтобы не ошибиться с адаптацией кода. |
| Золотой молоток | Нет | Мой девиз нет вещи лучше чем другая вещь и для каждой задачи свой инструмент. Я стараюсь не идти только одним инструментом для всех задач. |
| Фактор невероятности | Есть | Так же как пример выше со слепой верой я там описал ситуацию, а это в принципе то же самое. |
| Ранняя оптимизация | Нет | Нужно хорошо протестировать код чтоб ей занятся. Исправление багов критичных думаю это не касается (я бы их исправлял сразу как увидел). |
| Изобретение велосипеда | Есть | Пример свой фреймворк. Есть готовые фреймворки, но в данном случае его написание делалось с целью обучения. На реальных проектах думаю в большинстве своём возьмут готовый уже. |
| Изобретение квадратного колеса | Есть | Мы написали фреймворк безусловно хуже тех что пишут крупные компании. Но так как это делалось с целью обучения это не критично. А в продакшн такое точно не выпустят никогда. |
| Архитектурные |  |  |
| Большой комок грязи | Нет | Так как задействованы паттерны для проектирования и код читаем и понятен. |
| Затычка на ввод | Есть | Так как у меня в коде вместо валидации try есть проверки if. В некоторых случаях прилетают None например если в форме нет name. Нужно написать везде валидацию на try и не допускать чтоб пролетели лишние или неправильные данные. |
| Волшебная кнопка | Нет | Нужно работать по MVC шаблону |
| Членовредительство | Да | Довольно много таких классов, заточенных на определенный функционал в нашем проекте. Я считаю что это плохо когда класс нужно расширить функционально, а он не может. Это хорошо когда класс не нужно расширять в функциональности. Не всегда понятно нужно ли будет класс расширять в будущем. |
| Дымоход предприятия | Нет | Наше приложение не связано с другими приложениями (особенно приложениями других компаний). |
| Дымоход системы | Нет | Связи с какими-то другими подразделениями компаний у нас точно нет в приложении. |
| Путаница | Нет | Стараюсь не подмешивать в функции того что в них не требуется по функционалу и ожидаемости от них. |
| Инверисия абстракции | Нет | У меня есть один приватный метод \_\_instance в классе синглтоне. Но там он не мешает так как его потомки не должны наследовать. Так как потомкам он не нужен, то можно использовать. |