Университет ИТМО, факультет программной инженерии и компьютерной техники

Двухнедельная отчётная работа по «Информатике»: аннотация к статье

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата прошедшей лекции | Номер прошедшей лекции | Название статьи/главы книги/видеолекции | Дата публикации (не старше 2021 года) | Размер статьи (от 400 слов) | Дата сдачи |
| 25.09.2024 | 2 | МОДЕЛИРОВАНИЕ CИCТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ В CЕТИ | 05.12.2021 | ~960 слов | 25.09.2024 |
| 09.10.2024 | 3 | Применение алгоритмов сжатия информации для задачи когнитивного отображения проектного пространства многослойных экранов защиты от радиации | 21.11.2023 | ~1000 слов | 09.10.2024 |
| 23.10.2024 | 4 | 1. Что такое Protocol Buffers (Protobuf) | 09.08.2023 | ~600 слов | 23.10.2024 |
| 06.11.2024 | 5 | 1. Начало работы с VBA в Office | 07.04.2023 | ~4100 слов | 06.11.2024 |
| 20.11.2024 | 6 | XeLaTeX для оформления текстов: Текст, рисунки, таблицы, автоматизация | 30.09.2024 | ~3000 слов | 20.11.2024 |
| 04.12.2024 | 7 | Record-and-Replay тестирование — сочетание достоинств юнит и интеграционных тестов | 22.08.2021 | ~3500 слов | 18.12.2024 |

Выполнил(а) Суворов.С.Д , № группы *P3115* , оценка

Фамилия И.О. студента не заполнять

|  |
| --- |
| **Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)**  https://habr.com/ru/articles/573948/ |
| **Теги, ключевые слова или словосочетания (минимум три слова)**  **Тестирование IT-систем; TDD; Юнит- и интеграционное тестирование; Java.** |
| **Перечень фактов, упомянутых в статье (минимум четыре пункта)**   1. Record-and-Replay – это гибрид подходов с внешними зависимостями, мок-серверами и отчасти юнит-тестами. 2. Очень часто интеграционные тесты бывают случайно успешными (flaky), т.е. то проходят, то не проходят, а также дороги в машинных ресурсах в зависимости от инфраструктуры. 3. Для Java есть несколько библиотек для RnR: основные это WireMock и AnyStub, VCR.py для Python, VCR для Ruby. 4. Юнит-тесты сильно связаны с исходным кодом и часто тестируют не контракт, а имплементацию и мало говорят о работе системы в целом, писать и поддерживать их дорого. 5. В пирамиде тестирования RnR занимает место на границе между интегр-ми и юнит-тестами. 6. Ресурсы на тестирование ограничены, поэтому нужно искать методы тестирования, способные обеспечить как можно большую степень уверенности в качестве продукта, потратив как можно меньше ресурсов. 7. В RnR при первом проходе тестов записываются ответы сторонних систем, а при последующих воспроизводим записанные ранее ответы, что решает несколько проблем: по сравнению с чистыми интеграционными скорость выполнения возрастает, исчезает проблема мокирования. |
| **Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)**   1. **RnR тесты значительно упрощают рефакторинг и являются подходящим средством для тестирования изменения кодовой базы без изменений контракта.** 2. **Кассеты, куда записались данные после прогона, в репозитории могут выступать дополнительным средством документирования и анализа приложения, поскольку записано все сетевое взаимодействие.** 3. **В отличие от юнит-тестов, в RnR нет необходимости иметь знания о внутренностях системы, поэтому такие тесты может писать и QA-специалист.** 4. **RnR-тестирование как правило эффективнее интеграционного при аналогичном результате.** |
| **Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)**   1. Падения RnR-тестов диагностировать существенно сложнее, чем юнит-тестов. 2. Для малого количества вариаций поведения приложения RnR теряет преимущества и превращается в интеграционные тесты с дополнительными проблемами (хранение кассет и т.д.). 3. RnR-тесты являются плохим выбором для тестирования алгоритмически-содержательных частей приложения, здесь лучше использовать юнит-тесты. |
| **ЖЕСТКАЯ ЦИТАТА ДНЯ:**  **Никогда не сдавайтесь, идите к своей цели! А если будет сложно – сдавайтесь.** |