

Выполнил(а) Барсуков М. А., № группы P3115, оценка
Фамилия И.О. студента не заполнять

Название статьи/главы книги/видеолекции

Record-and-Replay тестирование — сочетание достоинств юнит и интеграционных тестов

ФИО автора статьи (или e-mail)

[@cepreu4habr](mailto:cepreu4habr)

Дата публикации
(не старше 2019 года)
"22" августа 2021 г.

Размер статьи
(от 400 слов)
6144

Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)

<https://habr.com/ru/post/573948/>

Теги, ключевые слова или словосочетания

TDD; Тестирование IT-систем; Java; Юнит- и интеграционное тестирование

Перечень фактов, упомянутых в статье

1. Ресурсы на тестирование ограничены, поэтому нужно искать методы тестирования, способные обеспечить как можно большую степень уверенности в качестве продукта, потратив как можно меньше ресурсов.
2. 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.
3. 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.
4. Record-and-Replay – это гибрид подходов с внешними зависимостями, мок-серверами и отчасти юнит-тестами.
5. 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.
6. Для Java есть несколько библиотек для RnR: основные это WireMock и AnyStub, VCR.py для Python, VCR для Ruby.
7. В пирамиде тестирования RnR занимает место на границе между интегр-ми и юнит-тестами.

Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)

1. RnR-тестирование как правило эффективнее интеграционного при аналогичном результате.
2. Кассеты, куда записались данные после прогона, в репозитории могут выступать дополнительным средством документирования и анализа приложения, поскольку записано все сетевое взаимодействие.
3. RnR тесты значительно упрощают рефакторинг и являются подходящим средством для тестирования изменения кодовой базы без изменений контракта.
4. В отличие от юнит-тестов, в RnR нет необходимости иметь знания о внутренностях системы, поэтому такие тесты может писать и QA-специалист.

Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)

1. Для малого количества вариаций поведения приложения RnR теряет преимущества и превращается в интеграционные тесты с дополнительными проблемами (хранение кассет и т.д.).
2. Падения RnR-тестов диагностировать существенно сложнее, чем юнит-тестов.
3. RnR-тесты являются плохим выбором для тестирования алгоритмически-содержательных частей приложения, здесь лучше использовать юнит-тесты.

Ваши замечания, пожелания преподавателю или анекдот о программистах



Заходит однажды тестировщик в бар.
 Забегает в бар.
 Пролезает в бар.
 Танцует, проникает в бар.
 Крадёт в бар.
 Врывается в бар.
 Прыгает в бар.