

Подходы к организации тестирования при развёртывании системы

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной работе представлена роль тестирования в разработке системы на основе микросервисной архитектуры. Приводятся примеры типов тестов и пирамиды тестирования. В результате исследования определена роль тестирования при развёртывании систем. Полученные данные позволяют распределить тесты по квадрантам, определив их важность и применимость на каждом этапе разработки.

Ключевые слова: квадрант тестирования, пирамида тестирования, тестирование, развёртывание системы.

Approaches to organizing testing during system deployment

Vikhlyayev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This paper presents the role of testing in the development of a system based on microservice architecture. It provides examples of test types and a testing pyramid. The study identifies the role of testing in system deployment. The findings allow for the distribution of tests across quadrants, highlighting their importance and applicability at each stage of development.

Keywords: testing quadrant, testing pyramid, testing, system deployment.

1 Введение

1.1 Актуальность

Современная разработка программного обеспечения требует внятной стратегии проверки качества, иначе усилия распределяются неравномерно, одни виды дефектов находятся слишком поздно, а другие проверки дублируются или выполняются впустую. На практике многие команды сталкиваются с перекосом, либо чрезмерным упором на дорогие интерфейсные тесты, либо слабым покрытием на уровне отдельных модулей. В этой работе взяты две известных модели, пирамида и квадрант тестирования, и рассмотрены в связке. Пирамида даёт чёткое представление о том, сколько тестов и на каком уровне должно быть, чтобы процесс оставался быстрым и предсказуемым. Квадрант добавляет к этому классификацию по целям и аудитории, помогает разграничить проверки, направленные на поиск

ошибок в коде, и те, что оценивают соответствие бизнес требованиям или нефункциональные характеристики. В условиях, когда архитектуры становятся распределёнными, а циклы поставки более частыми, обе модели по отдельности уже не дают полной картины. Их совместное использование позволяет выстроить сбалансированный подход, где каждый тип проверок занимает своё место, не перегружая процесс и не оставляя слепых зон. Поэтому тема оказывается полезной не только с теоретической стороны, но и для повседневной практики как основа для принятия решений при построении или рефакторинге процесса тестирования в конкретном проекте.

1.2 Обзор исследований

В своей работе Б.Гылызов, Н.И.Томилова ссылаясь на пирамиду тестирования, определили самые низкие категории, методом модульного и интеграционного тестирования [1]. А.В.Мельников, А.В.Мищенко выделили основные уровни в тестировании промышленных систем, сформулировав расположение данных уровней в пирамиду тестирования, в которой показали наиболее эффективную стратегию тестирования [2]. А.Шепель в своём исследовании особое внимание уделил применению контрактного и интеграционного тестирования, построению пирамиды тестов и внедрению CI/CD. На основании анализа предлагаются рекомендации по внедрению эффективной стратегии тестирования в микросервисных системах. [3]. В работе Б.С.Кириякова представлена статистическая модель многократного тестирования учащихся. [4]. А.В.,Ткачев, Д.В.Иртегов описали методику автоматического тестирования развивающегося веб-приложения [5]. В работе И.Р.Низамутдинова автоматизированное E2E-тестирование представлено как инструмент обеспечения стабильности релизов при постоянных А/В-экспериментах [6].

1.3 Цель исследования

Цель исследования – описать подходы к организации тестирования при развёртывании системы

2 Материалы и методы

В материалах приводятся подходы на основе квадранта тестирования и пирамиды тестирования. Представлено уравнивание проверок на примере EndToEnd и Unit тестов.

3 Результаты и обсуждения

Квадрант тестирования позволяет категоризировать тесты и понять их важность и применимость в разных этапах разработки.

В нижней части квадранта находятся технологически направленные тесты (рис 1). Они помогают разработчикам создавать систему. Небольшие и быстрые юнит-тесты и тесты производительности обычно автоматизированы.

В верхней части квадранта находятся бизнес направленные тесты. Это тесты, которые необходимы неискушённым пользователям. Например,

заказчиком или аналитиком. Понять, что система реализует те функциональности, которые они заказывали. Это могут быть дорогие и сложные ручные EndToEnd-тесты. А также исследовательские тесты, направленные на поиск ошибок в системе. Каждый из типов тестов необходим и служит определённой цели. И какое количество каких тестов необходимо, зависит от конкретной задачи, которую решает система.

Если важна надёжность, то реализуется большое количество различных тестов.

Если важнее скорость разработки, то её увеличивают за счёт того, что мирится с ошибками в боевой среде.



Рис. 1. Квадрант тестирования

Пирамида тестирования позволяет понять, какое количество, каких тестов нужно использовать в системе (рис.2).

EndToEnd-тесты дают больше уверенности в работоспособности всей системы. Обычно EndToEnd-тест осуществляется либо через внешние API системы, либо через пользовательский интерфейс. При этом полностью тестируется какая-то определённая функциональность. Такие тесты длительные и дорогие в поддержке.

С другой стороны, юнит-тесты небольшие и автоматизированные. С их помощью тестируют изолированные части кода, как правило, полностью автоматизированы, пишутся разработчиками и выполняются быстро. Таким образом, подход пирамиды тестирования, говорит о том, что должно быть большое количество автоматизированных Unit-тестов. Так как они могут проверить автоматически код изолированный друг от друга.

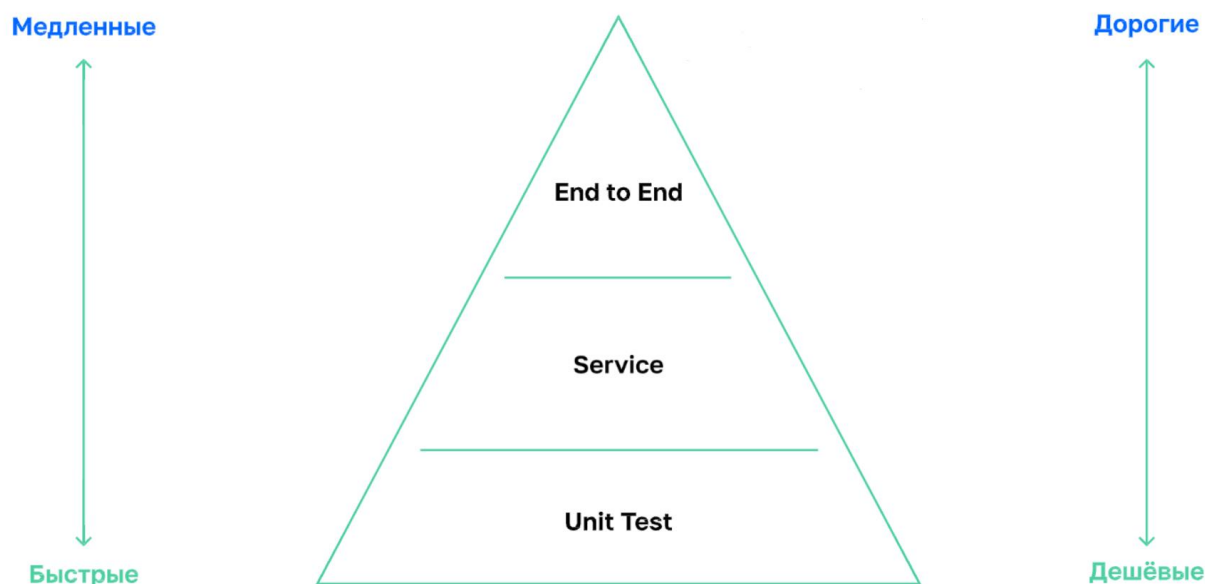


Рис. 2. Пирамида тестирования

Должно быть, достаточное количество сервисных интеграционных тестов, которые позволяют проверить конкретный сервис в изолированной среде. И небольшое количество полноценных EndToEnd-тестов, которые проверяют определённый функционал от начала до конца. При этом, чем сложнее тест, тем дольше он выполняется и тем дороже его разработка и поддержка.

Все тесты должны быть частью системы развёртывания. Если при сборке в рамках конвейера непрерывной интеграции, какой-то из юнит-тестов падает, он должен остановить создание артефакта и остановить дальнейшие шаги конвейера. То же самое касается интеграционного теста или EndToEnd-теста. Каждый упавший тест должен останавливать артефакт от продвижения дальше и возвращать его обратно разработчику на доработку.

4 Вывод

Таким образом, чем больше имеется автоматизированных тестов и чем больше они интегрированы в конвейер развёртывания, тем больше вероятность надёжности системы. Потому что все проверки, осуществляемые автоматически, направлены на то, чтобы не допустить регресса. Вследствие чего, разные типы тестов требуют разного подхода и разных инструментов и тестирование должно быть частью процесса развёртывания сервисов.

Библиографический список

1. Гылызов Б., Томилова Н. И. Шаблоны тестирования программного обеспечения //Иновации. Наука. Образование. 2021. №. 26. С. 1688-1699.
2. Мельников А.В., Мищенко А. В. Тестирование промышленных программных систем с использованием пирамиды тестирования //Актуальные направления научных исследований XXI века: теория и

- практика. 2019. Т. 7. №. 1. С. 257-261.
3. Шепель А. Особенности тестирования в проектах с микросервисной архитектурой //Актуальные исследования. 2023. №. 47 (177).
 4. Кирьяков Б.С. Статистическая модель многократного тестирования учащихся //Вестник Рязанского государственного университета им. СА Есенина. 2008. №. 18. С. 3-25.
 5. Ткачев А.В., Иртегов Д.В. Методика автоматического тестирования развивающегося веб-приложения //Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2019. Т. 17. №. 3. С. 93-110.
 6. Низамутдинов И.Р. Автоматизированное E2E-тестирование как инструмент обеспечения стабильности релизов при постоянных A/B-экспериментах //Universum: технические науки. 2026. Т. 1. №. 2 (143). С. 20-25.