Системы интеграционного тестирования ПО

Слайд 1

Предмет нашего исследования - системы интеграционного тестирования программного обеспечения. Но перед переходом к непосредственной теме, следует упомянуть, что такое тестирование.

**Тестирование программного обеспечения** — это проверка соответствия между реальным и ожидаемым поведением программы.

В​ мире разработки программного ‍обеспечения ‍тестирование ​является ключевым элементом​ обеспечения качества продукта.

Слайд 2

Первым этапом тестирования является **модульное тестирование**. Оно фокусируется на проверке *отдельных компонентов программы*, ‍чтобы убедиться, что каждый из них работает корректно *в изоляции*.

**Интеграционное тестирование** в свою очередь направлено на проверку *взаимодействия между* различными *модулями* или‍ компонентами системы.   
Его **целью** является *выявление дефектов, возникающих при интеграции компонентов*, и обеспечение их корректной работы в совокупности.

Слайд 3

Существует несколько подходов к интеграционному тестированию, наиболее распространёнными из них являются **метод большого взрыва** и **инкрементальное тестирование**:

Смысл **метода большого взрыва** можно определить так: ***все модули*** или компоненты *объединяются* сразу и ***тестируются*** ***как единое целое***.

К преимуществам данного метода можно отнести то, что он позволяет *сэкономить время и деньги*, поскольку все тесты можно проводить одновременно.

Оно дает хорошее *представление о функциональности и производительности* системы, что помогает *снизить риск упустить* какую-либо *проблему* во время тестирования.

Тестирование методом "большого взрыва" может быть проведено быстро, а значит, можно быстрее вывести свой продукт на рынок.

К недостаткам можно отнести *сложности с* выделением и *устранением отдельных проблем*. При возникновении таких ошибок их поиск и решение может быть *дорогостоящим и отнимать много времени*.

Также могут возникнуть сложности с координацией работы с несколькими командами и отделами.

Определить, готова ли система к запуску, убедиться в том, что все тесты проведены правильно, может быть непросто.

Слайд 4

Инкрементальное тестирование в свою очередь можно разделить на 3 вида: **нисходящая, восходящая и гибридная**.

Инкрементальный подход осуществляется с помощью фиктивных программ, называемых **заглушками** и **драйверами**. Заглушки и драйверы не реализуют всю логику модуля, а только *моделируют его работу.*

\*click\*

**Нисходящий** метод *начинается* *с верхнего уровня* и постепенно спускается вниз к более мелким компонентам. Используются заглушки для имитации работы нижних уровней.

\*click\*

**Восходящий** метод начинается с нижнего уровня и постепенно продвигается вверх, к более крупным компонентам. Используются драйверы для имитации работы верхних уровней.

\*click\*

**Гибридный** метод – это комбинация нисходящего и восходящего подходов, при котором используются и драйверы, и заглушки.

Слайд 5

Существуют различные инструменты и фреймворки для проведения интеграционного тестирования, среди которых:

- JUnit (для Java): Поддерживает создание и выполнение тестов, включая интеграционные.

- TestNG (для Java): Более мощный и гибкий инструмент по сравнению с JUnit, часто используемый для интеграционного тестирования.

- PyTest (для Python): Фреймворк, поддерживающий как модульное, так и интеграционное тестирование.

- Selenium: Инструмент для автоматизированного тестирования веб-приложений, часто используемый для интеграционного тестирования веб-интерфейсов.

- Postman: Используется для тестирования API(программных интерфейсов), помогает в интеграционном тестировании взаимодействия различных микросервисов.

Слайд 6

Для нашего проекта мы выбрали JUnit. Он является распространённым фреймворком для тестирования программ на языке Java, который упрощает создание надежных и эффективных тестов.

Наш интеграционный тест проверяет совместную работу четырёх вычислительных модулей. Нами был использован метод большого взрыва, потому как сложность вычислений достаточно низка, чтобы мы могли вычислить задачу самостоятельно и сравнить данные вычислений модулей с рассчитанными. Правильность модулей, входящих в состав нашего приложения, была предварительно проверена на корректность, потому в использовании инкрементального подхода в данном случае нет необходимости. А в случае появления некорректного результата будет достаточно провести модульное тестирование.

Слайд 7

Таким образом, интеграционное тестирование — это неотъемлемая часть процесса разработки программного обеспечения, направленная на проверку взаимодействия между различными модулями системы. Оно играет ключевую роль в обеспечении качества и надежности программных продуктов.

Правильное и своевременное проведение интеграционного тестирования позволяет выявить дефекты на ранних стадиях разработки, что существенно снижает затраты на их исправление и повышает общую эффективность разработки.