

Классификация тестирования по уровням

ЗИСП11-1, ЗИСР11-2

Классификация тестирования по уровням

Цель занятия: рассмотреть четыре основных уровня тестирования



Классификация тестирования по уровням

Принято разделять тестирование на виды по следующим категориям:

1. По объектам
2. По глубине тестирования – то есть разделение тестовых испытаний на типы проводится в зависимости от количества времени и объема работы
3. В соответствии с традиционными показателями качества.

Классификация тестирования по уровням

Принято классифицировать тестирование по следующим уровням:

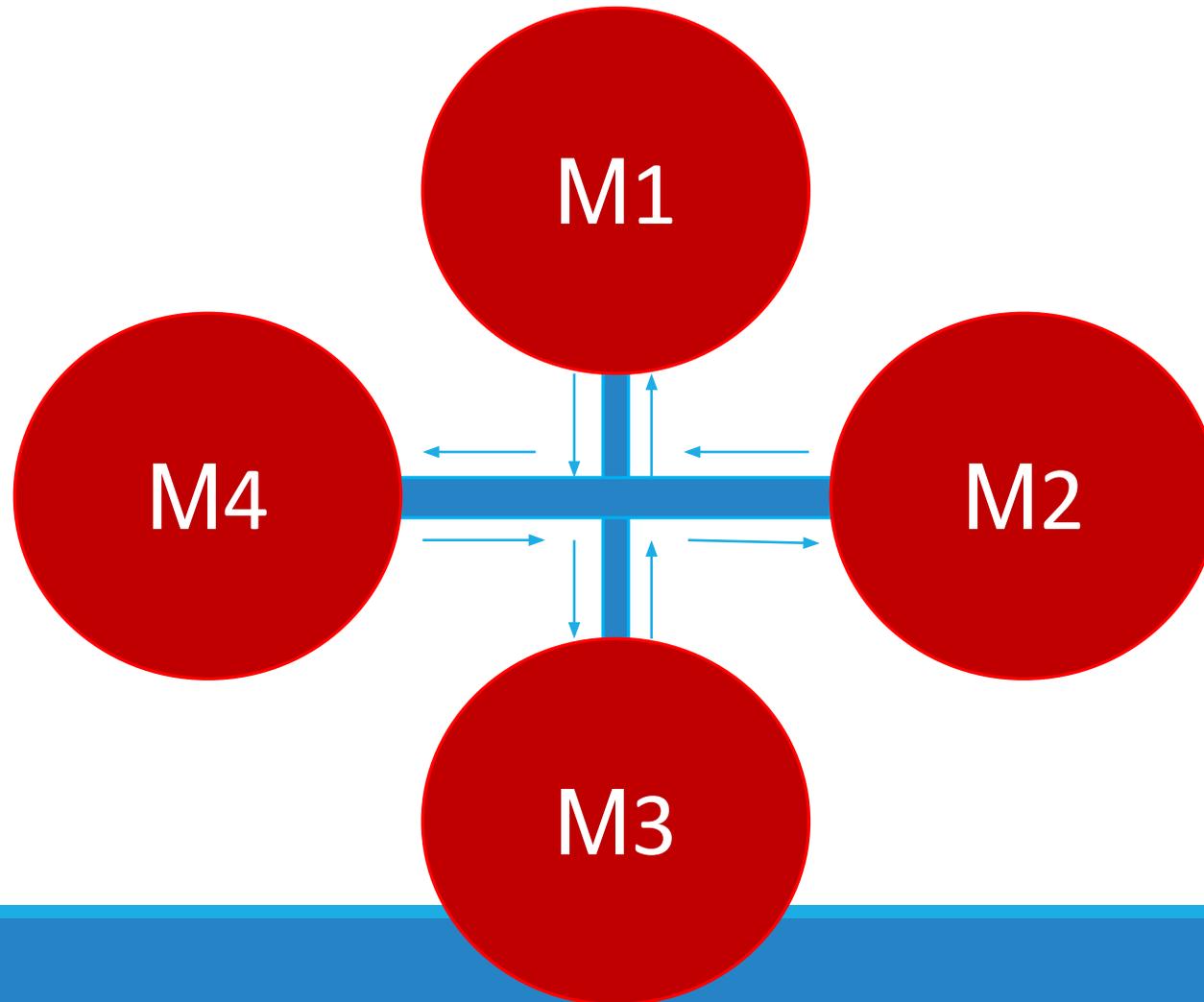
1. Модульное тестирование
2. Интеграционное тестирование
3. Системное тестирование
4. Приемочное тестирование

Первый уровень. Модульное тестирование

Модульное тестирование (Unit testing) – тестирование каждой функциональности приложения отдельно, в искусственно созданной среде.

Именно потребность в создании искусственной рабочей среды для определенного модуля, требует от тестировщика знаний в автоматизации тестирования и некоторых навыков программирования.

Первый уровень. Модульное тестирование



Первый уровень. Модульное тестирование

Драйвер – определенный модуль теста, который выполняют тестируемый нами элемент.

Заглушка – часть программы, которая симулирует обмен данными с тестируемым компонентом, выполняет имитацию рабочей системы.

Первый уровень. Модульное тестирование

Заглушки нужны для:

1. Имитирования недостающих компонентов для работы данного элемента.
2. Поддачи или возвращения модулю определенного значения, возможность предоставить тестеру самому ввести нужное значение.
3. Воссоздания определенных ситуаций (исключения или другие нестандартные условия работы элемента).

Первый уровень. Модульное тестирование

Во-первых, архитектура проекта должна быть спроектирована в соответствии с идеями ООП (четкое деление на классы, каждый из которых выполняет свою определенную функцию), что обеспечит систему грамотным делением на модули.

Также, модульное тестирование должно быть менее затратным при поиске дефектов, чем другие виды тестов и должно снижать время отладки кода.

Первый уровень. Модульное тестирование

Модульное тестирование мотивирует программистов писать код максимально оптимизированным, проводить рефакторинг (упрощение кода программы, не затрагивая ее функциональность), так как с помощью Юнит-тестирования можно легко проверить работоспособность рассматриваемого компонента.

Первый уровень. Модульное тестирование

Необходимость отделения реализации от интерфейса (ввиду особенностей модульного тестирования), что позволяет минимизировать зависимости в системе.

Документация Юнит-тестов может служить примером «живого документа» для каждого класса, тестируемого данным способом.

Модульное тестирование помогает лучше понять роль каждого класса на фоне всей программной системы.

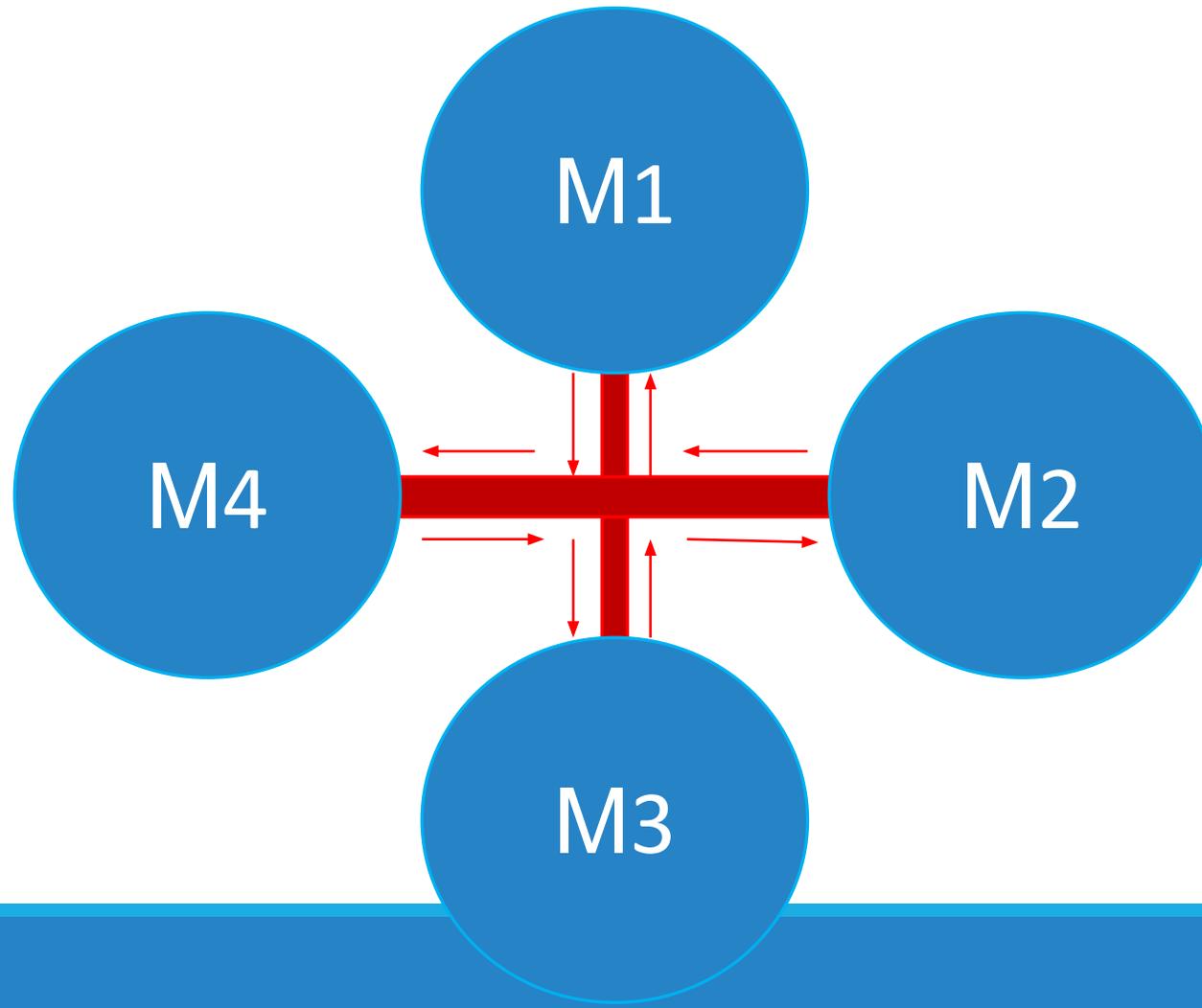
Первый уровень. Модульное тестирование

Также, при «разработке через тестирование», которая активно используется в экстремальном программировании, модульное тестирования является одним из основных инструментов, позволяющий разрабатывать модули в соответствии с требованиями к данному модулю.

Второй уровень. Интеграционное тестирование

Исходя из различий между модульным тестированием и системным тестированием, интеграционное тестирование является переходным этапом между представлением программы в виде отдельных модулей в виде полностью функциональной системы.

Второй уровень. Интеграционное тестирование



Второй уровень. Интеграционное тестирование

Интеграционное тестирование – вид тестирования, при котором на соответствие требований проверяется интеграция модулей, их взаимодействие между собой, а также интеграция подсистем в одну общую систему. Для интеграционного тестирования используются компоненты, уже проверенные с помощью модульного тестирования, которые группируются в множества.

Второй уровень. Интеграционное тестирование

Данные множества проверяются в соответствии с планом тестирования, составленным для них, а объединяются они через свои интерфейсы.

Второй уровень. Интеграционное тестирование

Существует несколько подходов к интеграционному тестированию:

- Снизу вверх. Сначала собираются и тестируются модули самих нижних уровней, а затем по возрастанию к вершине иерархии. Данный подход требует готовности всех собираемых модулей на всех уровнях системы.

Второй уровень. Интеграционное тестирование

Существует несколько подходов к интеграционному тестированию:

- Сверху вниз. Данный подход предусматривает движение с высокоуровневых модулей, а затем направляется вниз. При этом используются заглушки для тех модулей, которые находятся ниже по уровню, но включение которых в тест еще не произошло.

Второй уровень. Интеграционное тестирование

Существует несколько подходов к интеграционному тестированию:

- Большой взрыв. Все модули всех уровней собираются воедино, а затем тестируется. Данный метод экономит время, но требует тщательной проработки тест кейсов.

Второй уровень. Интеграционное тестирование

При автоматизации тестирования используется Система непрерывной интеграции. Принцип ее действия заключается в следующем:

1) Система непрерывной интеграции производит мониторинг системы контроля версий.

2) При изменении исходных кодов в репозитории производится обновление локального хранилища.

Второй уровень. Интеграционное тестирование

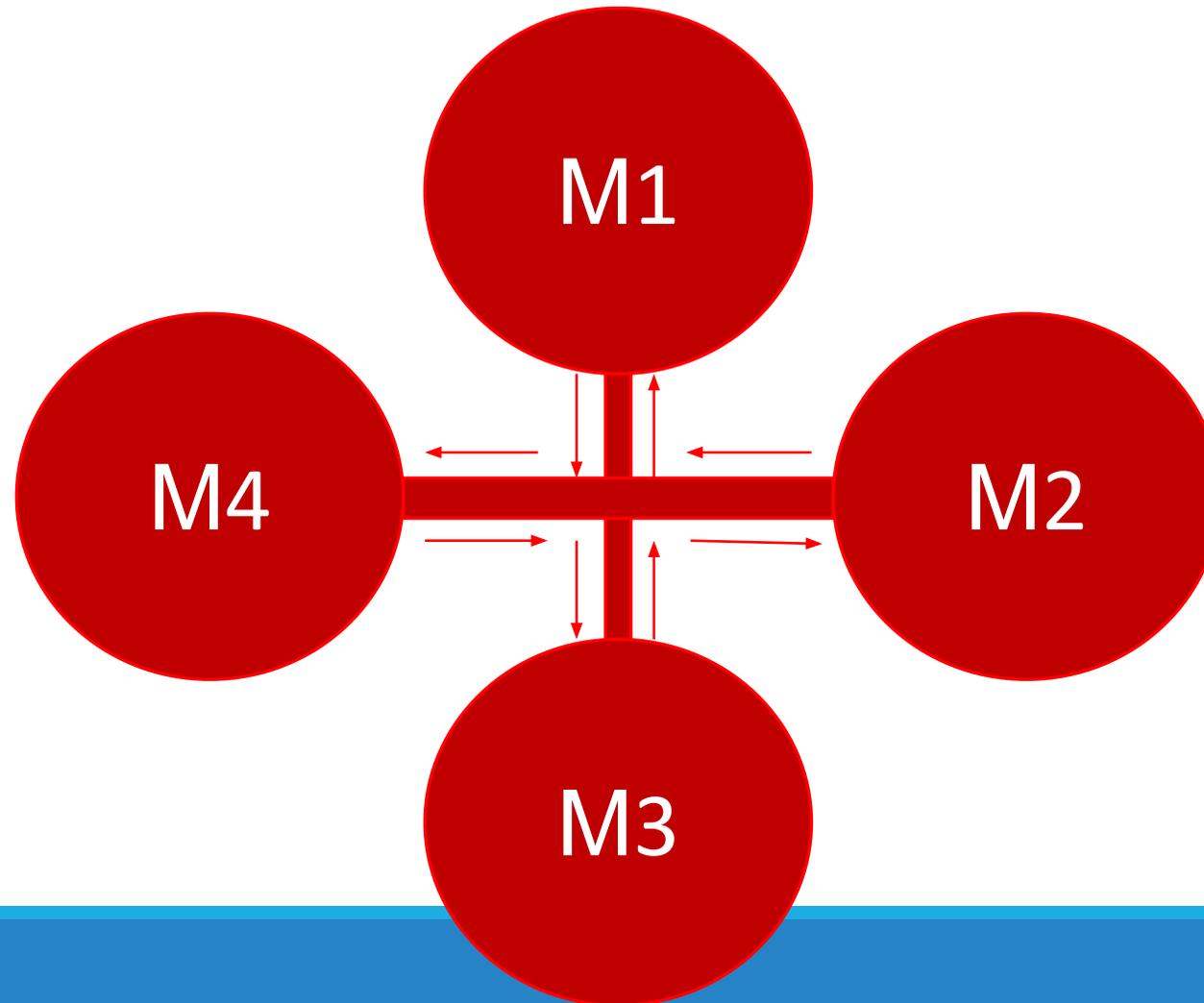
- 4) Исходные коды компилируются в готовые выполняемые модули.
- 5) Выполняются тесты интеграционного уровня.
- 6) Генерируется отчет о тестировании.

Это позволяет тестировать систему сразу после внесения изменений, что существенно сокращает время обнаружения и исправления ошибок.

Третий уровень. Системное тестирование

Системное тестирование – это тестирование программного обеспечения выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям, как функциональным, так и не функциональным.

Второй уровень. Интеграционное тестирование



Третий уровень. Системное тестирование

Позволяет обнаружить следующие типы дефектов:

- Неправильное использование системных ресурсов.
- Непредусмотренные комбинации пользовательских данных.
- Проблемы с совместимостью окружения.
- Непредусмотренные сценарии использования.
- Несоответствие с функциональными

Третий уровень. Системное тестирование

Системное тестирование выполняется методом «Черного ящика», т.к. проверяемое множество является «внешними» сущностями, которые не требуют взаимодействия с внутренним устройством программы. Также выполнять его рекомендуется в окружении, максимально приближенном к окружению конечного пользователя.

Третий уровень. Системное тестирование

Можно выделить 2 подхода к системному тестированию:

- На базе требований. Тестирование проводится в соответствии с функциональными или нефункциональными требованиями, для каждого из которых пишется testcase (тестовые прецеденты)..

Третий уровень. Системное тестирование

- На базе случаев использования. Тестирование происходит в соответствии с вариантами использования продукта, на основе которых создаются usecases (пользовательские прецеденты). Для каждого из данных пользовательских прецедентов создаются свои тестовые прецеденты.

Четвертый уровень. Приемочное тестирование

Приемочное (acceptance) – вид тестирования, проводимый на этапе сдачи готового продукта (или готовой части продукта) заказчику.

Целью приемочного тестирования является определение готовности продукта, что достигается путем прохода тестовых сценариев и случаев, которые построены на основе спецификации требований к разрабатываемому ПО.

Четвертый уровень. Приемочное тестирование

Результатом приемочного тестирования может стать:

- Отправка проекта на доработку.
- Принятие его заказчиком, в качестве выполненной задачи.

Четвертый уровень. Приемочное тестирование

Это финальный этап тестирования продукта перед его релизом. При этом, он не является сверх тщательным, всеохватывающим и полным – тестируется, в основном, только основной функционал.

Четвертый уровень. Приемочное тестирование

Приемочное тестирование проводится либо самим заказчиком, либо группой тестировщиков, представляющих интересы заказчика, либо тестировщиками компании-разработчика. Зависит от предпочтений компании-заказчика.