**ЛЕКЦИЯ 11**

**Тема:** Признаки критического модуля. Отладка ПО. Методы и принципы отладки. Критерии завершения тестирования ГОСТ Р ИСО/МЭК 12119-2000

**Цель:** изучить общие положения о тестировании и отладки программы. Изучить основные признаки критического модуля

**МЕТОДИКА ТЕСТИРОВАНИЯ ПС.**

**Методики тестирования**:

1. *Тестирование элементов* (модулей), проверяющее результаты кодирования ПС.

Проверяют результаты модулей, соответствуют действительности или нет

1. *Тестирование интеграции*, ориентированное на выявление ошибок этапов проектирования ПС.

Проверяют на правильность проектирования ПО, допущены ли были при проектировании системы ошибки.

1. *Тестирование правильности*, проверяющее корректность этапа анализа требований к ПС.

Вводим требования , которые мы получили на этапе анализа и вводим в систему и сравниваем соответствие результаты.

Хотели, чтобы система распознавала лица, сканируем изображение и сморим распознает или нет и какие ошибки выдает.

1. *Системное тестирование*, выявляющее дефекты этапа системного анализа ПС.

Проверяем на несоответствия Программного оборудования. К примеру будет ли работать система при малом объёме оперативной памяти.

**Тестирование модулей подвергаются:**

1. Интерфейс.
2. Внутренние структуры данных.
3. Независимые пути.
4. Пути обработки ошибок.
5. Граничные условия.

**При тестировании путей обнаруживаются следующие категории ошибок**:

1. Ошибочные вычисления.
2. Некорректные сравнения.
3. Неправильный поток управления.

***Для тестирования модуля необходимы дополнительные средства***:

• ***Драйвер*** – *управляющая программа, которая принимает исходные данные и ожидаемые результаты тестовых вариантов*, запускает в работу тестируемый модуль, получает из модуля реальные результаты и формирует донесение о тестировании.

• ***Заглушка*** – замещает модули, которые вызываются тестируемым модулем.

**ТЕСТИРОВАНИЕ КОМПЛЕКСА ПРОГРАММ**

**Цель сборки и тестирования комплекса программ** (интеграций) – взять модули, протестированные как элементы, и построить программную структуру, требуемую проектом.

Проверить правильную ли систему мы спроектировали и реализовали зависимости от поставленных задач в начале проекта .

соответствует ли система требования заказчика, все ф-и выполняет или нет.

**Два способа комбинирования модулей в систему:**

• ***Монолитное тестирование*** – *каждый модуль тестируется независимо друг от друга*, последовательно или параллельно, затем модули собираются в программу за «один раз». **Преимущество**: можно распараллелить процесс, тем самым убыстрить его выполнение.

• ***Пошаговое тестирование*** *– постепенно тестируем* либо нисходящее (можно показать программу сразу), либо восходящее тестирование (система не существует до тех пор, пока не будет добавлен последний модуль)

Постепенно тестируем либо готовое элемент либо тестируем его в процессе разработки.

**Признаки критического модуля**:

это то когда нужно созданный модуль разделять или исправлять ошибки

1. ***Реализует несколько требований к системе.***

Требование и высокая скорость считывания и обработка файлов, но при этом без использования SSD накопитель и с маленькой ОП.

1. ***Имеет высокий уровень управления.***

Сильно связный с другими модулями, может вывести всю систему сиз работоспособного состояния

1. **Имеет высокую сложность или склонность к ошибкам.**

Не понятно как работает его внутренние ф-и на чем они завязаны или есть вероятность, что при большом количестве вводимых данных наш модуль выйдет из строя.

1. **Имеет определенные требования к производительности обработки. Тестирование функций. Тестирование системы**

Предъявлено много требований одному модулю

1. **Тестирование удобства использования.**

Не удобно его подключать к системе или занимает длительный процесс времени(написание доп.скрипта)

1. **Тестирование на предельных объемах.**

Есть задачи, есть предметная область, должны прогнать модуль на определенном объёме входных данных и отследить, как себя ведет модуль.

1. **Тестирование на предельных нагрузках.**
2. **Удобства эксплуатации.**
3. **Защита**.
4. **Производительность**.
5. **Требования к памяти**.
6. **Конфигурация оборудования**.

Должен иметь определенный набор аппаратного и программного обеспечения и прошивок и сопроводительной документации по их исользованию

1. **Совместимость**.
2. **Удобства установки**.
3. **Надежность**.
4. **Восстановление**.
5. **Удобства обслуживания**.
6. **Документация.**

***Приемо-сдаточные испытания***

***Критерии завершения тестирования***: ГОСТ Р ИСО/МЭК 12119-2000

1. Выбирается какое-то *количество заранее установленных ошибок или время проверки*.
2. При проведении тестирования *тесты должны стать неудачными*.
3. *По графику*.

**Цель Тестирование правильности**– *подтвердить, что функции, описанные в спецификации требований, соответствуют ожиданиям заказчика*.

**элементов информации**, **вырабатываемых в процессе разработки ПС:**

проверяем каждый пункт заданным требования и его описанию, есть он в наличии или нет, если нет, то добавляем его.

1. Спецификация требований к ПС.
2. Руководство пользователя.
3. Системная спецификация.

Представление требований системе

1. План программного проекта.
2. Спецификация проектирования.
3. Листинги исходных текстов программ.
4. План и методика тестирования.
5. Руководство по работе и инсталляции.
6. .exe код выполняемой программы.
7. Описание БД.
8. Руководство пользователя по настройке.
9. Документ сопровождения.
10. Стандарты и методики разработки программного средства.

**ОТЛАДКА**

**Отладка** -тап разработки компьютерной программы, на котором обнаруживают, локализуют и устраняют ошибки.

Чтобы понять, где возникла ошибка, приходится: узнавать текущие значения переменных; выяснять, по какому пути выполнялась программа

**Отладка** – совокупность действий:

1. Разработка тестовых данных.
2. Динамическое и статическое тестирование.
3. Диагностика, локализация причин отклонения, корректировки, изменения.

**Два исхода отладки**:

• Причина ошибки найдена, уничтожена.

• Не найдена.

**Методы отладки**

1. ***Аналитические*** – используют методы дедукции и индукции.
2. ***Экспериментальные*** – методы «грубой силы»: трассировка, отладочная печать, трассировка переменных, потоков управления.

**Принципы отладки**:

1. Использовать средства отладки как вспомогательные.
2. Избегать экспериментирования.
3. Думать.
4. Остерегаться ошибок при процессе корректировки.
5. Все исправления следует документировать.

**Контрольные вопросы:**

1. Что такое отладка?
2. Какую совокупность действий включает в себя отладка?
3. Методы отладки?
4. Какие исходы отладки вы знаете?
5. Какие элементы информации вырабатываются в процессе разработки ПС?
6. Какие критерии завершения тестирования вы знаете?
7. Назовите признаки критического модуля?
8. Назовите методики тестирования?

Что подвергается тестированию модулей?

**Список использованных источников:**

1. Орлов, С.А. Технологии разработки программного обеспечения: учебник / С.А. Орлов. – СПб: Питер, 2002. – 464 с.
2. Липаев, В.В. Управление разработкой программных средств: Методы, стандарты, технология / В.В. Липаев. – М.: Финансы и статистика, 1993.
3. Липаев, В.В. Тестирование программ / В.В. Липаев. – М.: Радио и связь, 1986.
4. Липаев, В.В., Технология сборочного программирования / В.В. Липаев, Б.А. Позин, А.А. Штрик. – М.: Радио и связь, 1992.
5. Сертификация продукции. Международные стандарты и руководства ИСО/МЭК в области сертификации и управления качеством.
6. Лифиц, И.М. Стандартизация, метрология и сертификация /И.М. Лифиц. –М.: Юрайт-издат, 2004. – 335 с.
7. Сертификация сложных технических систем /Л.Н. Александровская [и др.]. – М.: Логос, 2001. – 312 с.
8. Якушев, А.И., Взаимозаменяемость, стандартизация и технические измерения / А.И Якушев, Л.Н. Воронцов, Н.М. Федотов. – М.: Машиностроение,