

Лабораторная работа №7.

21it1.chinikaylo.a@pdu.by [Сменить аккаунт](#)



Совместный доступ отсутствует



Черновик сохранен.

***Обязательный вопрос**

Тест №7 ОТЧЕТЫ О ПРОХОЖДЕНИИ ТЕСТОВ

Запись о результате прохождения каждого тестового примера в полной форме содержит следующую информацию:

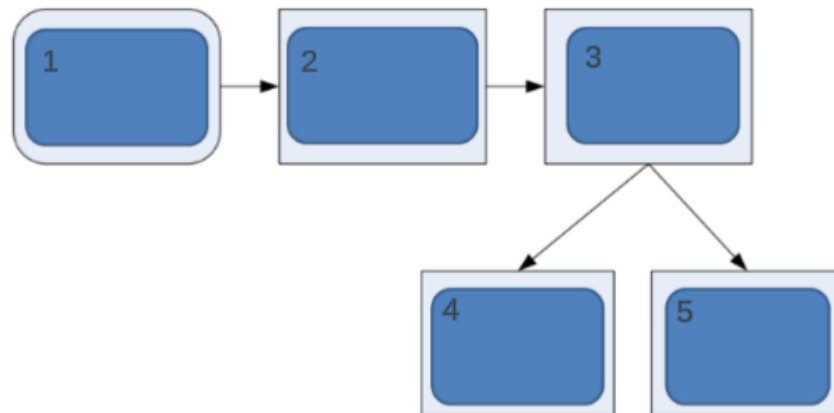
- ☒ Краткое описание тестового примера.
- ☒ Перечисление всех входных значений тестового примера.
- ☒ Для каждой пары “ожидаемое и реальное выходное значение” – информацию о совпадении или несовпадении этих значений.
- ☒ Перечисление всех ожидаемых и реальных выходных значений теста.
- ☒ Сообщение о том, пройден или не пройден тестовый пример.
- ☐ Идентификатор проекта.
- ☒ Идентификатор тестового примера.
- ☐ Перечисление всех реальных выходных значений

Интеграционное тестирование проводится методом *

- ☐ Тестирования моделей
- ☒ Белого ящика
- ☐ Чёрного ящика



Расставьте этапы в схеме отчета о генерации тестов *



- ☒ Среда тестирования -> Отчет о прохождении тестов -> Запрос на изменение -> Требования -> Исходные тесты
- ☐ Отчет о прохождении тестов -> Среда тестирования -> Запрос на изменение -> Требования -> Исходные тесты
- ☐ Запрос на изменение -> Отчет о прохождении тестов -> Среда тестирования -> Требования -> Исходные тесты

Интерфейсные дефекты выявляются при ... *

- ☒ Интеграционном тестировании
- ☐ Нагрузочном тестировании
- ☐ Системном тестировании
- ☐ Модульном тестировании



Соотнесите основные понятия и их определения *

| | входы для проверки системы и предполагаемые выходы в зависимости от входов, если система работает в соответствии с ее спецификацией требований. | входы, которые используются для проверки системы. | тест, который обнаруживает пока еще необнаруженную ошибку. | непредсказуемое поведение системы, приводящее к неожиданному результату, которое могло быть вызвано дефектами, содержащимся в ней. |
|---------------------------------|---|---|--|--|
| ОШИБКА - | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| ТЕСТОВЫЕ ДАННЫЕ- | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| УДАЧНЫЙ ТЕСТ - | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| ОТКАЗ - | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| ТЕСТОВАЯ СИТУАЦИЯ (TEST CASE) – | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |



Соотнесите строки и столбцы чтобы получились правильные утверждения *

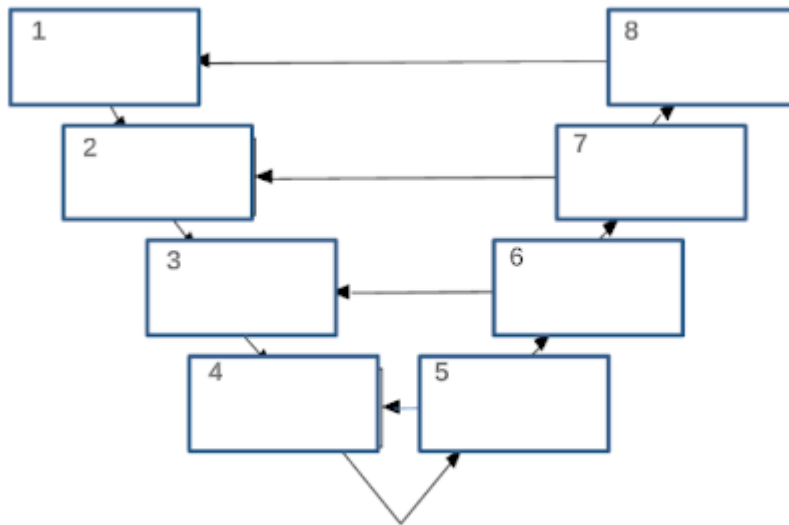
| | выполняется для каждого независимого программного модуля и является наиболее распространенным видом тестирования. | это объединение отдельных модулей в систему и тестирование системы в целом. | выполняется для полностью реализованного программного продукта. | позволяет пс прогнозиру данные производител системы |
|--------------------------------|---|--|---|---|
| Системное тестирование | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Интеграционное тестирование | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Модульное тестирование | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Нагрузочное тестирование | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |



Расставьте названия этапов V-образной модели жизненного цикла разработки ПО.

*

(Порядок этапов соответственно цифрам)



- ☒ 1 Требования -> 2 Функции -> 3 Архитектура -> 4 Код -> 5 Проверка кода -> 6 Проверка архитектуры -> 7 Проверка функций -> 8 Проверка требований
- ☐ 1 Проверка требований -> 2 Функции -> 3 Требования -> 4 Архитектура -> 5 Код -> 6 Проверка кода -> 7 Проверка архитектуры -> 8 Проверка функций
- ☐ 1 Требования -> 2 Проверка требований -> 3 Функции -> 4 Проверка функций -> 5 Архитектура -> 6 Код -> 7 Проверка кода -> 8 Проверка архитектуры
- ☐ 1 Проверка требований -> 2 Проверка кода -> 3 Проверка архитектуры -> 4 Проверка функций -> 5 Требования -> 6 Функции -> 7 Архитектура -> 8 Код

Отметьте правильные утверждения *

- ☐ Изменение программного кода, как правило НЕ ТРЕБУЕТСЯ, если поведение системы не соответствует требованиям.
- ☒ Изменение программного кода, как правило, ТРЕБУЕТСЯ, если поведение системы не соответствует требованиям.
- ☐ Изменение требований, как правило, НЕ ТРЕБУЕТСЯ, если имеет место неадекватное поведение системы в ситуациях, не предусмотренных требованиями
- ☒ Изменение требований, как правило, ТРЕБУЕТСЯ, если имеет место неадекватное поведение системы в ситуациях, не предусмотренных требованиями.

В заголовочную часть отчета о прохождении тестов включается следующая информация (отметьте пункты с правильными отчетами): *

- ☒ Идентификатор тестируемого модуля или группы модулей и номера их версий.
- ☒ Время начала выполнения теста и его продолжительность.
- ☒ Конфигурацию тестового стенда, на которой выполнялся тест.
- ☐ Название тестируемого модуля.
- ☒ Общий идентификатор группы тестовых примеров, включенных в отчет.
- ☒ Имена и фамилии автора тестов и/или лица, выполнявшего тесты.
- ☒ Ссылку на разделы и версии тест- требований или функциональных требований.
- ☒ Название проекта или тестируемой системы.



Модульное тестирование проводится методом *

- ☐ Чёрного ящика
- ☒ Белого ящика
- ☐ Тестирования моделей

Локальные дефекты выявляются при ... *

- ☐ Системном тестировании
- ☐ Интеграционном тестировании
- ☐ Нагрузочном тестировании
- ☒ Модульном тестировании

Отметьте все виды системного тестирования *

- ☒ Функциональное тестирование
- ☒ Тестирование производительности
- ☒ Нагрузочное или стрессовое тестирование
- ☒ Тестирование конфигурации
- ☒ Тестирование безопасности
- ☒ Тестирование надежности и восстановления после сбоев
- ☒ Тестирование удобства использования



Выберите фазы процесса тестирования: *

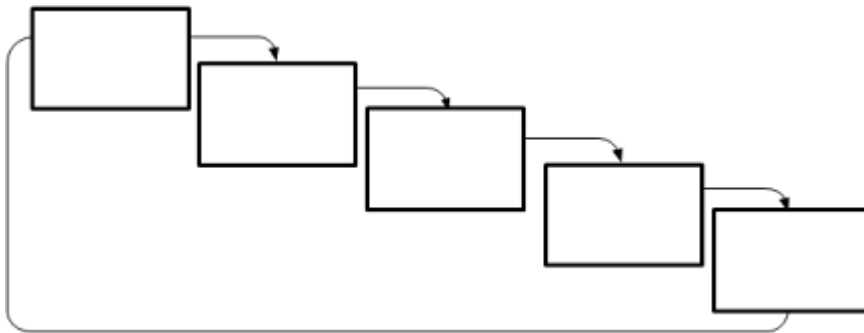
- ☒ ПЛАНИРОВАНИЕ
- ☐ ТЕСТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОЙ ДОКУМЕНТАЦИИ
- ☐ ТЕСТИРОВАНИЕ ИНСТАЛЛЯЦИИ
- ☒ РАЗРАБОТКА ТЕСТОВ
- ☒ АНАЛИЗ РЕЗУЛЬТАТОВ
- ☒ ОПРЕДЕЛЕНИЕ ЦЕЛЕЙ
- ☒ ВЫПОЛНЕНИЕ ТЕСТОВ

После исправления дефект переводится в состояние "Resolved". При этом разработчик должен указать следующую информацию о дефекте: *

- ☐ Описание дефекта.
- ☒ Причину возникновения дефекта.
- ☒ Место исправления, как минимум, с точностью до исправленного файла.
- ☒ Время, затраченное на исправление.
- ☐ Наименование подсистемы в которой был обнаружен дефект.
- ☒ Краткое описание того, что было исправлено.

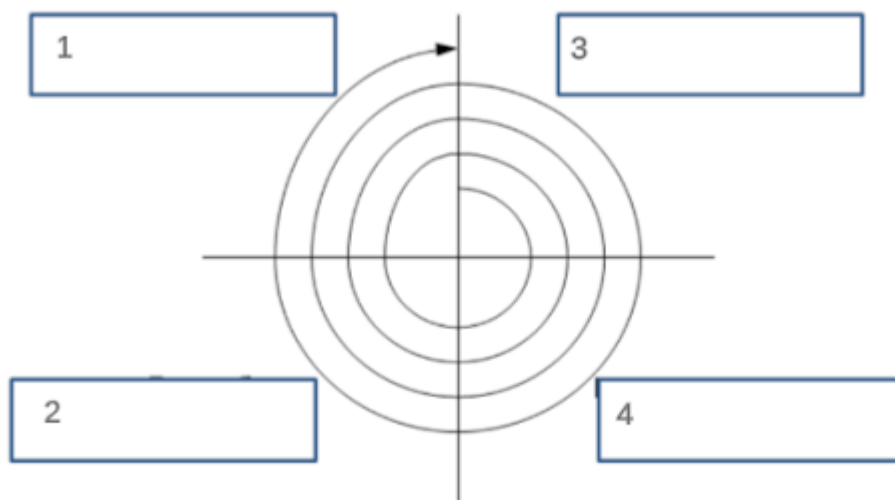


Выберите правильную последовательность этапов каскадной модели жизненного цикла разработки ПО. *



- ☐ Требования к ПО -> Системные требования -> Код -> Архитектура -> Тесты
- ☐ Системные требования -> Код -> Требования к ПО -> Архитектура -> Тесты
- ☒ Системные требования -> Требования к ПО -> Архитектура -> Код -> Тесты
- ☐ Архитектура -> Требования к ПО -> Системные требования -> Код -> Тесты

Расставьте названия этапов спиральной модели жизненного цикла разработки ПО. *



- ☒ 1 Верификация, 2 Разработка, 3 Анализ, 4 Проектирование
- ☐ 1 Анализ, 2 Верификация, 3 Разработка, 4 Проектирование
- ☐ 1 Анализ, 2 Проектирование, 3 Верификация, 4 Разработка
- ☐ 1 Верификация, 2 Проектирование, 3 Разработка, 4 Анализ

[Назад](#)

[Отправить](#)

[Очистить форму](#)

Никогда не используйте формы Google для передачи паролей.

Форма создана в домене Полоцкий государственный университет имени Евфросинии Полоцкой.
[Сообщение о нарушении](#)

Google Формы



