Практическая работа № 4 Тестирование программных систем

Акт о тестировании

Акт

о результатах тестирования программы

для электронных вычислительных машин

АКТ

от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ N \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(число, месяц, год) (номер, присваиваемый организацией

научно-технической поддержки)

о результатах тестирования программы для электронных

вычислительных машин (далее - программа для ЭВМ)

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(название) (номер версии)

Комиссия в составе:

от организации научно-технической поддержки. От

------------------------------------------- -----------------------

(должность) (наименование заказчика экспертизы)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество) (должность)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя,

отчество)

провела тестирование в \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ программы для ЭВМ "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" (название)

(место проведения тестирования)

версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

(номер версии)

Программа для ЭВМ "\_\_\_\_\_\_\_\_\_\_\_\_\_" версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_ поступила с письмом (название) (номер версии)

от \_\_\_\_\_\_\_\_\_\_\_\_\_\_ N \_\_\_\_\_\_\_\_.

Комиссией проведено тестирование программы для ЭВМ на \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (приводится краткое описание тестовых примеров)

Тестирование включало в себя процедуру ввода исходных данных,

проведение расчета и анализ результатов расчета. Сравнению подлежали

значения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(перечисляются параметры, полученные в результате расчетов)

со значениями, приведенными в \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

(приводятся реквизиты Отчета)

Тестирование показало, что программа "\_\_\_\_\_\_\_\_\_\_" версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(название) (номер версии)

обеспечивает проведение расчетов в соответствии с ее назначением, описанным

в \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

(приводятся реквизиты Отчета)

Значения хэш-функций основных файлов программы для ЭВМ:

- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (файл "Название.exe");

- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (файл "Название.lib).

От организации научно-технической поддержки. От

------------------------------------------- -----------------------------

(наименование заказчика

экспертизы)

\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (расшифровка подписи) (подпись) (расшифровка подписи)

Практическая работа № 4 Тестирование программных систем

Отчет по результатам тестирования программы

ОГЛАВЛЕНИЕ

[ОПИСАНИЕ 7](#_Toc43419827)

[СВЕДЕНИЯ О ХОДЕ ИСПЫТАНИЙ 8](#_Toc43419828)

# ОПИСАНИЕ

Цель тестирования:

Экспериментальное определение фактических (достигнутых) характеристик свойств испытываемой программной системы.

Стратегии тестирования:

ВОСХОДЯЩЕЕ ТЕСТИРОВАНИЕ – программа собирается и тестируется снизу-вверх.

НИСХОДЯЩЕЕ ТЕСТИРОВАНИЕ – программа собирается и тестируется сверху вниз. Изолировано тестируется только головной модуль.

Методы тестирования:

МЕТОД БОЛЬШОГО СКАЧКА – каждый модуль тестируется автономно. По окончании тестирования модулей они интегрируются в систему все сразу.

МЕТОД САНДВИЧА – представляет собой компромисс между восходящим и нисходящим подходами. При использовании этого метода одновременно начинают восходящее и нисходящее тестирование, собирая программу как снизу, так и сверху и встречаясь, в конце концов, где-то в середине. Точка встречи зависит от конкретной тестируемой программы и должна быть заранее определена при изучении ее структуры.

# СВЕДЕНИЯ О ХОДЕ ИСПЫТАНИЙ

|  |  |
| --- | --- |
| Методы | Результат |
| НИСХОДЯЩЕЕ ТЕСТИРОВАНИЕ | Положительный |
| МЕТОД БОЛЬШОГО СКАЧКА | Положительный |
| МЕТОД САНДВИЧА | Положительный |
| ВОСХОДЯЩЕЕ ТЕСТИРОВАНИЕ | Положительный |

По окончанию тестирование, ошибок выявлено не было.

Практическая работа № 4 Тестирование программных систем

План тестирования

[ОПИСАНИЕ 7](#_Toc43420041)

[СВЕДЕНИЯ О ХОДЕ ИСПЫТАНИЙ 8](#_Toc43420042)

[ВВЕДЕНИЕ 11](#_Toc43420043)

[1.1 Цель работы 11](#_Toc43420044)

[1.2 Стратегии тестирования: 11](#_Toc43420045)

[1.3 Методы тестирования: 11](#_Toc43420046)

[1.4 Документирование 12](#_Toc43420047)

[ЦИКЛ ТЕСТИРОВАНИЯ 13](#_Toc43420048)

[2.1. Срочная активность 13](#_Toc43420049)

[2.2. Тестирование релиза 13](#_Toc43420050)

[2.3. Ежедневная активность 14](#_Toc43420051)

[2.5. Полугодичная активность 15](#_Toc43420052)

[ТЕСТОВЫЙ СТЕНД 16](#_Toc43420053)

[3.1. Планировщик задач автоматизированного тестирования 16](#_Toc43420054)

[3.2. Конфигурации оборудования 17](#_Toc43420055)

# ВВЕДЕНИЕ

# 1.1 Цель работы

Экспериментальное определение фактических (достигнутых) характеристик свойств испытываемой программной системы.

Тестирование является завершающим этапом разработки программного продукта. Ему предшествует этап статической и динамической отладки программ. В узком смысле цель тестирования состоит в обнаружении ошибок, цель же отладки – не только в обнаружении, но ив устранении ошибок. Цели у отладки и испытания разные. Полностью отлаженная программа может не обладать определенными потребительскими свойствами и тем самым быть непригодной к использованию по своему назначению. Не может служить альтернативой испытанию и проверка работоспособности программы на контрольном примере, так как программа, работоспособная в условиях контрольного примера, может оказаться неработоспособной в других условиях применения. Попытки охватить контрольным примером все предполагаемые условия функционирования сводятся в конечном счете к тем же испытаниям.

# 1.2 Стратегии тестирования:

ВОСХОДЯЩЕЕ ТЕСТИРОВАНИЕ – программа собирается и тестируется снизу-вверх.

НИСХОДЯЩЕЕ ТЕСТИРОВАНИЕ – программа собирается и тестируется сверху вниз. Изолировано тестируется только головной модуль.

# 1.3 Методы тестирования:

МЕТОД БОЛЬШОГО СКАЧКА – каждый модуль тестируется автономно. По окончании тестирования модулей они интегрируются в систему все сразу.

МЕТОД САНДВИЧА – представляет собой компромисс между восходящим и нисходящим подходами. При использовании этого метода одновременно начинают восходящее и нисходящее тестирование, собирая программу как снизу, так и сверху и встречаясь, в конце концов, где-то в середине. Точка встречи зависит от конкретной тестируемой программы и должна быть заранее определена при изучении ее структуры.

# 1.4 Документирование

Результатом выполнения данной работы является разработанный план тестирования ПC, отчет о тестировании и Акт о тестировании ПC.

# ЦИКЛ ТЕСТИРОВАНИЯ

В данной главе описан процесс тестирования, который состоит из следующих видов активности в порядке приоритета: срочная внеплановая активность, тестирование релиза, ежедневная плановая активность, разработка новых тестов, полугодичная активность.

# 2.1. Срочная активность

Заключается в выполнении тестировщиком срочных поручений менеджера проекта, большая часть из которых является критическими и блокирует ход разработки.

В данный вид активности также включается проведение верификации критических ошибок и доработок.

Тестировщику следует внимательно подходить к верификации, внесение изменений оказывает влияние на другие участки кода и может вносить дополнительные ошибки. Необходимо тщательно проверять предполагаемые зависимые участки кода таким образом, чтобы верификация приближалась к регрессивному тестированию по каждому инциденту. Менеджерам следует понимать, что большое количество срочности срывает выполнение плановых работ и ведет к уменьшению тестирования. Для этого необходимо планировать загрузку и увеличивать ритмичность работы тестирования.

# 2.2. Тестирование релиза

Данная процедура носит временный характер, стартуя в момент начала финального тестирования и заканчиваясь после принятия релиза. Обычно целью тестирования релиза является сдача продукта с определенными характеристиками к определенному сроку. Обязательным атрибутом данного вида тестирования является проверка обеспечения функциональных характеристик продукта в соответствии со спецификацией требований. Для этого проводится тщательный анализ спецификаций требований к системе в целом и к каждому компоненту. Важным является определение отличий текущего релиза от предыдущего и проведение по этим отличиям регрессивного тестирования. Тестировщик разрабатывает инсталляционный пакет. В процессе тестирования релиза удостоверяется успешное прохождение предыдущего ежедневного автоматического теста, выполняется функциональное alpha- и beta-тестирование, производится тестирование инсталляции и документации, проводятся дополнительные ручные тесты. Также осуществляется консультирование заказчика по вопросам переноса и внедрения системы.

В процессе тестирования релиза тестировщик открывает новые срочные инциденты.

# 2.3. Ежедневная активность

Ежедневная активность тестирования заключается в проведении запланированных автоматических тестов на тестовом стенде.

Каждую ночь производится попытка осуществить сборку системы из исходного кода. Таким образом выполняется операция «дымового» тестирования. В случае успеха осуществляется автоматический проход запланированных тестов во всех запланированных конфигурациях. В результате формируется отчет, который рассылается всех заинтересованным лицам. Сбой дымового тестирования является критической ошибкой и подлежит немедленному анализу.

В данный вид активности также включается проведение верификации некритичных ошибок и доработок в соответствии. На каждый открытый инцидент желательно написать тесты, проверяющие его. Повторно возникающие инциденты должны повлечь за собой разработку дополнительных тестов.

Тестировщик контролирует ход тестирования и получает ежедневный отчет о ходе тестирования. Он управляет включением тестов, настраивает тестовый стенд, конфигурирует необходимые окружения для проведения тестов.

Тестировщик постоянно анализирует спецификацию требований к системе, технические задания и регулярно выдает рекомендации по улучшению тестирования, а также сообщает разработчикам о расхождениях функциональности, описанной в задании и реальных характеристик системы, если таковые появляются.

Тестировщик осуществляет также дополнительные ручные тесты, которые необходимы. Действия, повторенные более чем три раза, нуждаются в последующей автоматизации.

Задачей ежедневной активности тестирования является также постоянная автоматизация тестирования.

В процессе усовершенствования системы некоторые контрольные примеры могут перестать выполняться успешно, например, при изменении пользовательского интерфейса. Поэтому ежедневной задачей тестирования является анализ причин неудач, обновление и исправление контрольных примеров.

В процессе тестирования тестировщик открывает новые инциденты по результатам тестирования, назначая их разработчикам.

# 2.5. Полугодичная активность

Производится один раз в полгода по запросу менеджера или после сдачи очередного релиза. В результате формируется отчет о текущем положении дел с предложениями по улучшению. Данная активность включает в себя проведение:

|  |  |
| --- | --- |
| Инспекция и критический просмотр кода | Руководитель, проектировщик, главный программист |
| Тестирование удобства использования интерфейса | Тестировщик |
| Анализ качества исходного кода | Тестировщик |
| Анализ покрытия тестами кода | Тестировщик |

# ТЕСТОВЫЙ СТЕНД

# 3.1. Планировщик задач автоматизированного тестирования

Необходимо разработать программное обеспечение на любом из скриптовых языков, которое позволит тестировщику конструировать и проигрывать необходимые тестовые последовательности в заданное время.

Данное программное обеспечение осуществляет ежедневную сборку, дымовое тестирование и автоматически запускает установленные блоки тестирования. Необходимо обеспечить поддержку управления несколькими компьютерами и различными виртуальными машинами из данного планировщика.

Планировщик обеспечивает интерфейс для формирования отчетов по результатам тестирования. Данные отчета представляют собой пронумерованный и разделенный список результатов пройденных контрольных примеров. В случае ошибки теста ее текст также сохраняется в отчет. Результатом является успех или код ошибки. Минимальным заданием является блок тестов. Блок тестов реализуется как отдельный исполняемый файл, который может содержать модульные, интеграционные, нагрузочные и т.п. тесты.

Необходимо разработать шаблон такого файла. Данный шаблон должен предоставлять функции записи результатов тестирования для планировщика. Необходимо разработать механизм интеграции средства автоматизированного тестирования GUI для проведения тестирования интерфейса пользовательских приложений. После завершения всего тестирования производится автоматическая экспертная оценка по проведенному тестированию. Отчет анализируется, и формулируется общее заключение. Помимо этого, планировщик предоставляет возможность проведения экспертного анализа. Производится расчет общего процента успехов при выполнении тестов.

Планировщик осуществляет слежение за запусками, формирует лог запусков и сохраняет результаты каждого прохода в отдельной папке.

Обработанный отчет высылается по электронной почте списку заинтересованных лиц. Этот список можно конфигурировать.

# 3.2. Конфигурации оборудования

Для работы системы требуется следующие тестовые конфигурации серверов и клиентских ПК. Для исполнения тестовых процедур необходим тестовый стенд, состоящий из Конфигурация (Клиент).