

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
Алтайский государственный технический университет им.
И.И. Ползунова

Старолетов С.М.

Верификация и тестирование программного обеспечения

Методические указания по проведению
лабораторных работ
для студентов направления 09.03.04
«Программная инженерия»
(бакалавриат)

Барнаул 2015

Старолетов С. М. Верификация и тестирование программного обеспечения: Методические указания по проведению лабораторных работ .— Барнаул: АлтГТУ, 2015. – 24с.

Методические указания по проведению лабораторных работ представляют собой материал и задания по дисциплине “Верификация и тестирование программного обеспечения”, предназначены для студентов направления 09.03.04 «Программная инженерия», бакалавриат.

(с) АлтГТУ, 2015

Содержание

Выписка из стандарта дисциплины	4
Общие сведения о лабораторном практикуме	5
Тема 1. Системы баг-трекинга – 4 часа	7
Тема 2. Модульное тестирование — 6 часов	9
Тема 3. Разработка через тестирование (TDD) – 6 часов	11
Тема 4. Функциональное тестирование – 4 часа	15
Тема 5. Работа с системой Unitesk – 6 часов.....	17
Тема 6. Работа с системой Spec Explorer – 6 часов	18
Тема 7. Верификация при помощи верификатора Spin - 7 часов	19
Литература.....	20
Приложение 1. Список вопросов для собеседования	23

Выписка из стандарта дисциплины

(специальность 09.03.04 “Программная инженерия”,

бакалавриат

Цели освоения дисциплины:

Целью освоения дисциплины "Верификация и тестирование программного обеспечения" является развитие культуры индустриальной разработки программ-ного обеспечения, в соответствии с которыми обучающийся должен быть спосо-бен при проектировании программных систем применять различные методы верификации и тестирования так, чтобы гарантировать качество создаваемого программного продукта.

Трудоемкость дисциплины: 3 ЗЕ (108ч.)

Содержание дисциплины:

Тема 1. Системы баг-трекинга

Тема 2. Модульное тестирование

Тема 3. Разработка через тестирование (TDD)

Тема 4. Функциональное тестирование

Тема 5. Работа с системой Unitesk

Тема 6. Работа с системой Spec Explorer

Тема 7. Верификация при помощи верификатора Spin

Компетенции:

- готовность применять основные методы и инструменты разработки программного обеспечения способность проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности (ПК-1);
- владение концепциями и атрибутами качества программного обеспечения (надежности, без-опасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества (ПК-4);
- готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-3).

Форма промежуточной аттестации: зачет

Общие сведения о лабораторном практикуме

Цель лабораторного практикума

Лабораторный практикум по дисциплине «Верификация и тестирование и программного обеспечения» служит для практического закрепления теоретических навыков, полученных в ходе изучения лекционного материала и формирования необходимых компетенций.

Цели выполнения лабораторных работ:

- закрепление теоретических знаний по дисциплине и применение этих знаний для тестирования и верификации программ;
- развитие навыков работы с современными методологиями и программными средствами для верификации и тестирования .

Организация лабораторного практикума

Для посещения лабораторных занятий студенческая группа делится на две подгруппы, посещающие отведенные для них часы. Это позволяет использовать индивидуальный подход к обучению, отвечать на возникшие вопросы и способствует лучшему усвоению материала.

На каждую лабораторную работу выдается общее задание, соответствующее изучаемому теоретическому материалу, и индивидуальный вариант задания (обычно подбирается студентом самостоятельно и утверждается преподавателем). Студент должен в соответствии с темой решить поставленную задачу и оформить отчет о проделанной работе.

Для каждой лабораторной работы предусмотрены сроки ее выполнения и защиты. Каждая работа оценивается по стобалльной

шкале. В случае выполнения и/или защиты работы после установленного срока, оценка может быть снижена.

Разработка программного обеспечения

Требования, предъявляемые к программам:

- программы должны быть работоспособными;
- тесты должны покрывать ключевые особенности программ;
- отчет должен содержать постановку задачи, описание решения задачи и текст программы с тестами (если необходимо ее написание), все необходимые снимки с экрана.

Тема 1. Системы баг-трекинга – 4 часа

Цели и задачи работы: изучение систем баг-трекинга и командной работы по организации тестирования

Теоретические сведения о работе приведены в литературе [1-11] и конспекте лекций (тема 1)

Задание к работе:

1. Ознакомиться с понятиями система «баг-трекинга», «план тестирования», «тестовый набор»
2. Описать функциональность своего программного обеспечения в виде предусловий, постусловий, инвариантов и use-case диаграмм
3. Протестировать программное обеспечения одноклассников

Описание используемых средств для выполнения работы:

программный продукт, веб-браузер

Методика выполнения работы:

1. Разделиться на группы из 3х человек
2. Выбрать и утвердить у преподавателя какой-то программный продукт, написанный самостоятельно на одном из курсов одним из членов команды
3. Ознакомиться с открытыми системами отслеживания ошибок (баг-трекинга).
4. Ознакомиться с системой контроля версий git и средой github.
5. Залить свое программное обеспечение, диаграммы вариантов использования и документацию на основной функционал в виде предусловий, постусловий, инвариантов
6. Разработать план тестирования и тестовые наборы для программ коллег

7. Протестировать программные продукты коллег и отметить в системе баг-трекинга (в данном случае в issues репозитория github) найденные ошибки и предложения по усовершенствованию

Требования к отчету:

Отчет должен содержать титульный лист, задание, описание программного продукта, план тестирования, информацию, внесенную в систему баг-трекинга.

Контрольные вопросы по лабораторной приведены в приложении.

Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Тема 2. Модульное тестирование — 6 часов

Цели и задачи работы: изучение методологии модульного тестирования

Теоретические сведения о работе приведены в литературе [1-11] и конспекте лекций по теме 2.

Задание к работе:

1. Изучить способы подключения фреймворка для модульного тестирования в проект
2. Ознакомиться со способами создания модульных тестов и написать их
3. Провести регрессивное тестирования

Описание используемых средств для выполнения работы : среда программирования с подключенным фреймворком для модульного тестирования (Nunit, Juint, Gtest).

Методика выполнения работы:

1. Добавить к существующему программному проекту фреймворк для проведения модульного тестирования (Nunit для проектов на C#, GTest для проектов на C++, Junit — для Java)
2. Создать тесты для основных модулей (классов) программы
3. Проверить их выполнение
4. Добавить в программу дополнительную функциональность (по согласованию с преподавателем)
5. Убедиться, что после добавления дополнительной функциональности все предыдущие тесты работают

Требования к отчету:

Отчет должен содержать титульный лист, модульные тесты и результаты их выполнения.

Контрольные вопросы по лабораторной приведены в приложении.

Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Тема 3. Разработка через тестирование (TDD) – 6 часов

Цели и задачи работы: Разработка учебного программного обеспечения с нуля при помощи методологии TDD

Теоретические сведения о работе приведены в литературе [1-11] и конспекте лекций (тема 3).

Задание к работе:

1. Изучить методологию TDD (Test-driven development) [1, 4]
2. Реализовать прототип программной системы на основе данной методологии

Варианты заданий:

1. Игра Поле чудес (угадать загаданное слово)
2. Игра Города (база городов, игроки по очереди вводят город за отведенное время, проверяется на правильность)
3. Игра Крестики-нолики
4. Научный калькулятор
5. Матричный калькулятор
6. Записная книга
7. Программа-будильник (заданная мелодия в заданное время)
8. Создатель Мемов (добавить текст на одну из predetermined картинок и сохранить)
9. Поисковик забытого кода на диске (поиск в заданных папках в файлах с исходниками)
10. Конвертор различных величин (американская система, старорусская, си)
11. Бесследный уничтожитель файлов и папок (забивать файл несколько раз заданными символами, потом уничтожить) ->
12. Электронный терапевт (есть некая база, пользователю задаются вопросы и ставится диагноз)

13. Генератор паттернов (генерация исходного кода по заданному паттерну с предпросмотром)
14. Игра “Угадай мелодию” (3 игрока, по очереди проигрывается мелодия и спрашивается)
15. Календарь с возможностью запуска программ в заданное время (планировщик)
16. Электронный поэт - генератор стихов из заданных фрагментов (задаются фрагменты и способы рифмы, из них случайно генерируется произведение)
17. Конвентор валют с возможностью предсказания курсов (несколько валют и история курса, можно узнать курс на след. дни экстраполяцией)
18. Хранитель паролей (сайты, кредитные карты)
19. Шифратор и дешифратор файлов в заданной папке (шифруем все файлы и расшифровываем)
20. Игра морской бой
21. Программа для конверсии графических файлов (размеры+ разные форматы)
22. Игра в дурака (2 игрока)
23. Приложение для создания миксов мелодий из нескольких файлов (выбираем файлы, начало и конец для каждого и получаем итоговый микс)
24. Игра “Угадай паттерн” (игрокам предоставляется код или UML схема и надо угадать паттерн)
25. Приложение для генерации скриншотов программ, с возможностью представления их, как будто они работают под разными ОС
26. Социальная сеть (логинка, добавление в друзья с заданным аккаунтом, посылка сообщения другу)

- 27.Soundcloud (список песен с возможностью добавления комментария на каждую секунду песни и отображение их в процессе проигрывания)
- 28.Система выбора и подстановки причесок для фотографий людей (операции с изображениями)
- 29.Настраиваемые часы
- 30.Редактор праздничного стола (категории еды, питья, расчет цены, сервировка на N человек)
- 31.Генератор поздравительных текстов для различных праздников
- 32.Генератор мелодий по заданным аккордам
- 33.Программа для напоминания о днях рождения друзей
- 34.Программа – консольный друг

Описание используемых средств для выполнения работы : среда программирования с подключенным фреймворком для модульного тестирования

Методика выполнения работы:

- 1. Создать пустой проект с подключенным фреймворком для модульного тестирования
- 2. Спроектировать основной класс для приложения
- 3. Написать тесты для него
- 4. Написать реализацию методов основного класса приложения, так, чтобы выполнялись написанные в п.3 тесты
- 5. Написать интерфейсы для вспомогательных классов приложения
- 6. Написать тесты для этих классов
- 7. Написать реализацию для всех методов вспомогательных классов, так, чтобы выполнялись все написанные тесты

8. Повторять создание тестов и реализации до тех пор, пока заданная и обсужденная с преподавателем функциональность не будет реализована, все вспомогательные классы не дописаны и все модульные тесты проходят

Требования к отчету:

Отчет должен содержать титульный лист, постановку задачи, исходный текст программы, тесты.

Контрольные вопросы по лабораторной приведены в приложении.

Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Тема 4. Функциональное тестирование – 4 часа

Цели и задачи работы: Проведение функционального тестирования программного обеспечения

Теоретические сведения о работе приведены в литературе [1-11] и конспекте лекций (тема 4, тема 5).

Задание к работе:

1. Разработать основные сценарии для функциональной работы приложения с использованием интерфейса пользователя
2. Провести функциональное тестирование средствами IBM Functional tester или Selenium

Описание используемых средств для выполнения работы :

программа IBM Functional Tester (пробная версия) или Selenium (по выбору студента)

Методика выполнения работы:

1. Ознакомиться с программой IBM Rational Functional Tester / Selenium
2. Выбрать и согласовать с преподавателем свое программное обеспечение, написанное ранее на языке C# или Java/ Web фреймворк
3. Создать план тестирования, включающий основные функциональные возможности своего ПО
4. Провести функциональное тестирование при помощи IBM Functional tester / Selenium
5. Добавить некоторую функциональность в приложение (обсудить с преподавателем)
6. Провести регрессивное тестирование.

Требования к отчету:

Отчет должен содержать титульный лист, снимки с экрана процесса тестирования

Контрольные вопросы по лабораторной приведены в приложении.

Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Тема 5. Работа с системой Unitesk – 6 часов

Цели и задачи работы: : изучение системы Unitesk

Теоретические сведения о работе приведены в литературе [1-11] и конспекте лекций (тема 6).

Задание к работе: самостоятельно выполнить задачу согласно методике

Описание используемых средств для выполнения работы :
система Unitesk

Методика выполнения работы:

1. Ознакомиться с понятиями Unitesk
2. Выбрать собственное программное обеспечение, содержащее некоторые вычисления и согласовать с преподавателем
3. Написать контрактные спецификации для вычислительных методов
4. Проверить правильность программного обеспечения согласно написанным спецификациям в среде Unitesk

Требования к отчету:

Отчет должен содержать титульный лист, текст программы, текст спецификации, снимки с экрана при верификации.

Контрольные вопросы по лабораторной приведены в приложении.

Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Тема 6. Работа с системой Spec Explorer – 6 часов

Цели и задачи работы: ознакомление с системой Spec Explorer

Теоретические сведения о работе приведены в литературе [1-11] и конспекте лекций (тема 7).

Задание к работе: проведение верификации в соответствии с методикой

Описание используемых средств для выполнения работы: Visual Studio 2010 с Spec Explorer

Методика выполнения работы:

1. Ознакомиться с принципами языка Spec#
2. Создать контрактные спецификации для вычислительных методов аналогично работе 5

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст спецификации, результаты тестирования

Контрольные вопросы по лабораторной приведены в приложении.

Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Тема 7. Верификация при помощи верификатора Spin - 7 часов

Цели и задачи работы: Верификация при помощи верификатора Spin

Теоретические сведения о работе приведены в источниках [1-11] и в конспекте лекций по дисциплине (тема 8).

Задание к работе: верифицировать модель согласно методике

Описание используемых средств для выполнения работы :
программа Spin[9]

Методика выполнения работы:

1. Выбрать собственное программное обеспечения, которое можно будет представить в виде автоматной модели и согласовать с преподавателем
2. Выделить логические состояния в программе
3. Создать автоматную модель исходя из состояний
4. Описать модель средствами языка Promela
5. Задать требования к модели в виде темпоральных предикатов
6. Провести верификацию модели средствами Spin

Требования к отчету:

Отчет должен содержать титульный лист, задание, модель на Promela, результаты верификации

Контрольные вопросы по лабораторной приведены в приложении

Б. Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Литература

1. Старолетов С.М. Тестирование и верификация программного обеспечения. - Барнаул: Изд-во АлтГТУ, 2015
2. Старолетов, С.М. Крючкова, Е.Н. Моделирование распределенных много-компонентных программных систем и их тестирование на основе автоматных вероятностных моделей. - Барнаул : Изд-во АлтГТУ, 107 с., 2011, – 3 экз.
3. Крючкова Е.Н.. Математическая логика и теория алгоритмов [Электронный ресурс] - Барнаул.: Изд-во АлтГТУ, 2013. – 216 с. – режим доступа - http://new.elib.altstu.ru/eum/download/pm/Kruchkova_ml.pdf
4. Кент Бек. Экстремальное программирование: разработка через тестирование(Test-driven Development by Example). Питер - ISBN 5-8046-0051-6, 2003 г. – 224 с.
5. Системная информатика: Сб. науч. тр. / Под ред. Марчука А.Г. - Н.: Изд-во СО РАН, 2004. - Вып.9: Формальные методы и модели информатики. - 359 с — 1 экз
6. Материалы сайта "Интернет университет информационных технологий":
 - 6.1. Курс «Маглинец, Ю. Анализ требований к автоматизированным информационным системам», сетевой адрес: <http://www.intuit.ru/studies/courses/2188/174/info>
 - 6.2. Курс «Котляров, В. Основы тестирования программного обеспечения», сетевой адрес: <http://www.intuit.ru/studies/courses/48/48/info>
 - 6.3. Курс «Налютин, Н. Синицин С. Верификация программного обеспечения», сетевой адрес: <http://www.intuit.ru/studies/courses/1040/209/info>

7. Материалы сайта “Сайт по автоматному программированию и мотивации к творчеству”, сетевой адрес <http://is.ifmo.ru>
8. Материалы сайта "Единое окно доступа к образовательным ресурсам", сетевой адрес <http://window.edu.ru/>
 - 8.1. Вельдер С.Э., Лукин М.А., Шалыто А.А., Яминов Б.Р. Верификация автоматных программ: Учебное пособие. - СПб.: СПбГУ ИТМО, 2011. - 242 с. Сетевой адрес <http://window.edu.ru/resource/817/72817>
 - 8.2. Синицын С.В., Налютин Н.Ю. Верификация программного обеспечения: Курс лекций. - М.: МИФИ (ГУ), 2006. - 158 с. Сетевой адрес - <http://window.edu.ru/resource/700/41700>
 - 8.3. Кулямин В.В. Методы верификации программного обеспечения / Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы", 2008. - 117 с. Сетевой адрес - <http://window.edu.ru/resource/168/56168>
 - 8.4. Лаврищева Е.М., Петрухин В.А. Методы и средства инженерии программного обеспечения: Учебник. - М.: МФТИ (ГУ), 2006. - 304 с. Сетевой адрес - <http://window.edu.ru/resource/699/41699>
 - 8.5. Труды Института системного программирования РАН. Том 10. Обеспечение надежности и совместимости Linux-систем. 2006 г. / Под редакцией чл.-корр. РАН В.П. Иванникова. - М.: ИСП РАН, 2006. - 180 с. Сетевой адрес - <http://window.edu.ru/resource/972/70972>
9. Concise Promela Reference. Сетевой адрес - <http://spinroot.com/spin/Man/Quick.html>
10. Кулямин В. В.. Критерии тестового покрытия, основанные на структуре контрактных спецификаций //Труды ИСП РАН, Подход UniTESK: итоги и перспективы. С:89-107, 2008. Сетевой адрес - <http://panda.ispras.ru/~kuliamin/docs/SpecCoverageCriteria-2008-ru.pdf>

11. Spec explorer – Microsoft Research. Сетевой адрес -
<http://research.microsoft.com/en-us/downloads/b33add8c-6172-444d-b1b1-6a91323ad7cc/default.aspx>

Приложение 1. Список вопросов для собеседования

1. Качество программного обеспечения.
2. Тестирование и верификация. Место в процессе разработки программных продуктов.
3. Классификация видов тестирования.
4. Тест-кейсы и тест-наборы.
5. Системы баг-трекинга.
6. Преимущества и цели модульного тестирования.
7. Фреймворки Junit, NUnit и Gtest. Создание и запуск тестовых наборов и тестов.
8. Покрытия тестов.
9. Применение тестирования во всем процессе разработки.
10. Принципы, преимущества и недостатки подхода разработки через тестирования
11. Требования к функциональному тестированию.
12. Использование инструмента IBM Rational functional tester.
13. Необходимость нагрузочного тестирования.
14. Обзор программных средств для проведения нагрузочного тестирования.
15. Сценарии нагрузочного тестирования.
16. Особенности тестирования web-приложений.
17. Разработка, управляемая моделями (Model driven development).
18. Разработка контрактных спецификаций для программных продуктов.
19. Использование системы Unitesk.
20. Использование продукта Microsoft Spec Explorer и языка Spec# для верификации моделей программ.
21. Темпоральные логики LTL и CTL.

22. Язык Promela для описания моделей взаимодействующих программ.
23. Трассировка моделей методом случайной прогулки и задаваемой пользователем симуляцией в системе Spin.