Министерство образования Республики Башкортостан

Государственное бюджетноепрофессиональное образовательное учреждение

Уфимский колледж статистики, информатики и

вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора  по учебной работе  З.З. Курмашева  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

Разработка Среды тестирования в Winccoa

Пояснительная записка к дипломному проекту

|  |  |
| --- | --- |
| Рецензент  С.О. Никитина  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. | Руководитель  О.В. Фатхулова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |
|  | Выпускник гр. 4П-1  И.М. Фаттахутдинов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

2018

АННОТАЦИЯ

Пояснительная записка к дипломному проекту содержит постановку и программу решения задачи «Разработка среды тестирования в WinCCOA».

Среда тестирования «Tester»разработана в среде разработки человеко-машинного интерфейса SimaticWindowsControlCenterOpenArchitectureна языке Control, предназначена для работы в операционной системе MS Windows 7\8\8.1\10 и отлажена на данных контрольного примера.

СОДЕРЖАНИЕ

лист

[Введение 4](#_Toc516232447)

[1 Постановка задачи 6](#_Toc516232448)

[1.1 Описание предметной области 6](#_Toc516232449)

[1.2 UML диаграмма 7](#_Toc516232450)

[1.3 Описание входной информации 9](#_Toc516232451)

[1.4 Описание выходной информации 10](#_Toc516232452)

[1.5 Общие требования к программному продукту 10](#_Toc516232453)

[1.6 Описание структуры файлов 11](#_Toc516232454)

[1.7 Контрольный пример 13](#_Toc516232455)

[2 Экспериментальный раздел 15](#_Toc516232456)

[2.1 Описание программного продукта 15](#_Toc516232457)

[2.2 Протокол тестирования 21](#_Toc516232458)

[2.3 Руководство пользователя 22](#_Toc516232459)

[3 Экономический раздел 32](#_Toc516232460)

[3.1 Расчет затрат на разработку 32](#_Toc516232461)

[3.2 Расчет цены и минимального количества копий тиражирования 35](#_Toc516232462)

[Заключение 39](#_Toc516232463)

[Приложение А. Содержимое файлов контрольного примера 40](#_Toc516232464)

[Приложение Б. Исходный код программного продукта 43](#_Toc516232465)

[Список использованных источников 44](#_Toc516232466)

ВВЕДЕНИЕ

Тестирование SCADA систем отнимает много времени и ресурсов, человеко-часов.Для этого было принято решение, разработать данное программное решение,которое упрощает и ускоряет процесс тестирования, а также значительно повышает порог прохождения нормативов, обнаружение ошибок в системах, сокращая человеческий фактор.

На данный момент на ранке программных продуктов отсутствует подобное решение, удовлетворяющее требования компании «СкадТех». Это и стало решающим фактором для начала разработки данного программного продукта.

Разрабатываемое программное решение должно быть реализовано в среде разработки человеко-машинного интерфейса (далее среда разработки HMI) SimaticWindowsControlCenterOpenArchitecture (далее WinCCOA) и представлять из себя гибкую среду разработки, отладки, воспроизведения интеграционных, системных тестов, а также формирования результатов и ведения журнала тестирования.

Целью дипломного проекта является создание программного продукта, решающий данную проблему.

Необходимо решить следующие задачи для достижения цели:

* постановка задач;
* проектирование;
* кодирование;
* отладка;
* тестирование;
* документирование;
* сопровождение;
* расширение и повторное проектирование.

Описание разделов пояснительной записки:

1. постановка задачи – описание бизнес-процессов, входной и выходной информации, бизнес-правила, задач, которые будут решаться в программном продукте;
2. экспериментальный раздел – описание реализованного программного продукта: модульная схема задачи, описание модулей, протокол тестирования, руководство пользователя;
3. экономический раздел – описание расчета необходимых затрат на создание программного продукта, расчета ценообразования, расчета необходимого количества копий.

1 Постановка задачи

1.1 Описание предметной области

Пусть требуется разработать модульно-подпроектное приложение, позволяющее создавать, редактировать, воспроизводить интеграционные и системные тесты, формировать результаты тестов и записывать ихв журнал. Система должна самостоятельно обнаруживать все тесты в директории, позволяя их разделять на плейлисты и категории.

Вся информация о плейлистах, категориях, тестах должна храниться на диске в определенных директориях и доступна для редактирования пользователям через сторонние текстовые редакторы.

Плейлисты представляют из себя файл с перечислением идентификаторов тестов, которые входят в его состав. В самом программном продукте они предназначены для создания сценария тестирования, которые будут воспроизводиться при запуске. Через интерфейс должна быть предусмотрена возможность создания, редактирования содержимого и удаления плейлистов.

Плейлисты характеризуются следующими параметрами:

* идентификатор – наименование файла;
* наименование – наименование файла без расширения;
* идентификаторы тестов.

Категории представляют из себя файл с перечислением идентификаторов тестов, которые входят в его состав. В самом программном продукте они предназначены для разделения по характерам и типам тестов, для удобного упорядочивания наборов тестов. Через интерфейс программного продукта должна быть предусмотрена возможность создания, редактирования содержимого и удаления категорий. Категории характеризуются следующими параметрами:

* идентификатор – наименование файла;
* наименование – наименование файла без расширения;
* тесты – содержимое файла, представляющее из себя перечисление идентификаторов тестов, которые входят в его состав.

Тест должен представлять из себя файл с написанным скриптом на языке программирования Control, который встроен в среду разработки HMIWinCCOA. Должна быть предусмотрена возможность создания, редактирования и удаления тестов из интерфейса программного продукта.

Тесты характеризуются следующими параметрами:

* идентификатор–наименование файла;
* наименование –наименование файла без расширения;
* скрипт теста – содержимое файла.

С данным программным продуктом должны работать следующие группы пользователей:

* тестировщики;
* интеграторы;
* разработчики ПО.

Данное решение должно помочь тестировщикам, интеграторам, разработчикам тем, что они могут запускать любые сценарии тестов в целях проверки работоспособности SCADA систем, без взаимодействия с реальными объектами, что может вызвать неисправимые последствия.

1.2 UMLдиаграмма

На рисунках 1.2.1, 1.2.2, 1.2.3 представлены диаграмма использования, диаграмма классов, диаграмма деятельности соответственно.

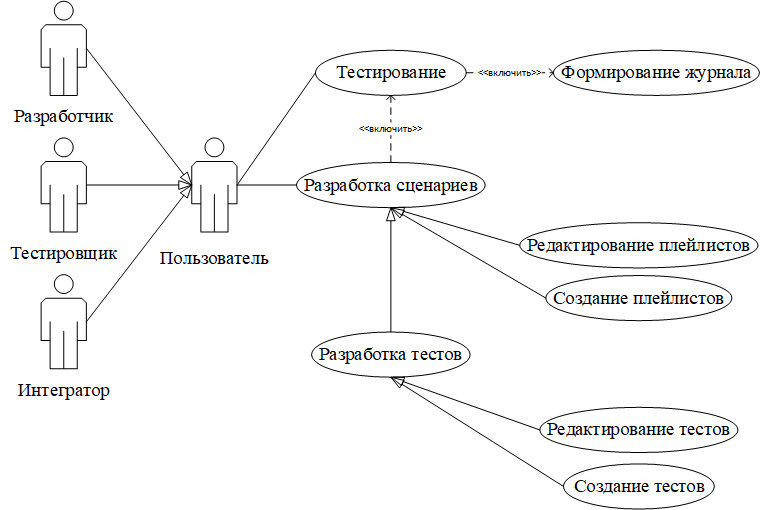


Рисунок 1.2.1 – Диаграммаиспользования

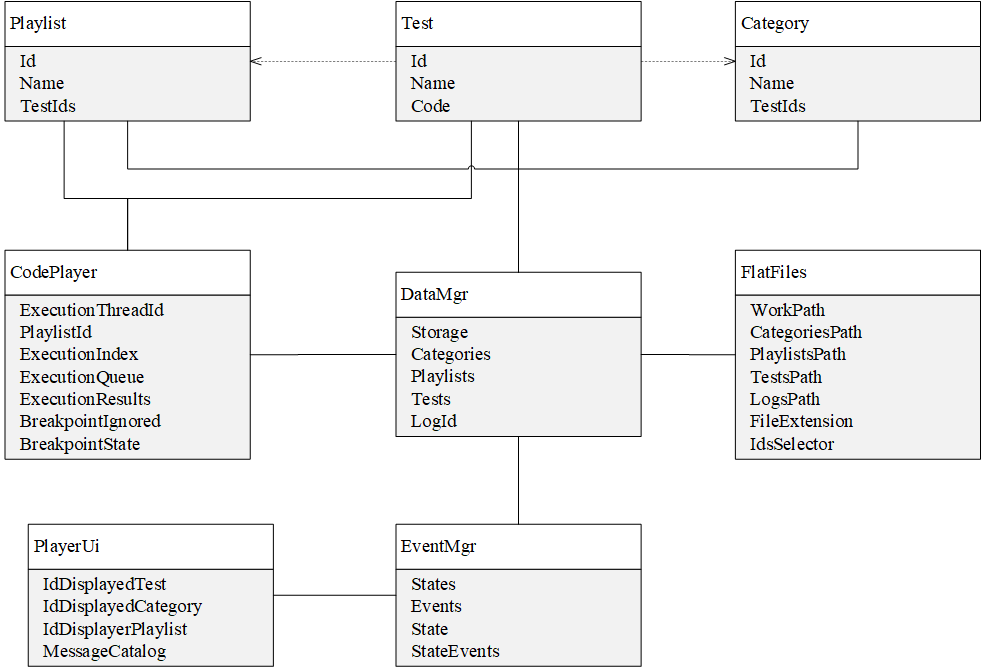


Рисунок 1.2.2 – Диаграмма классов

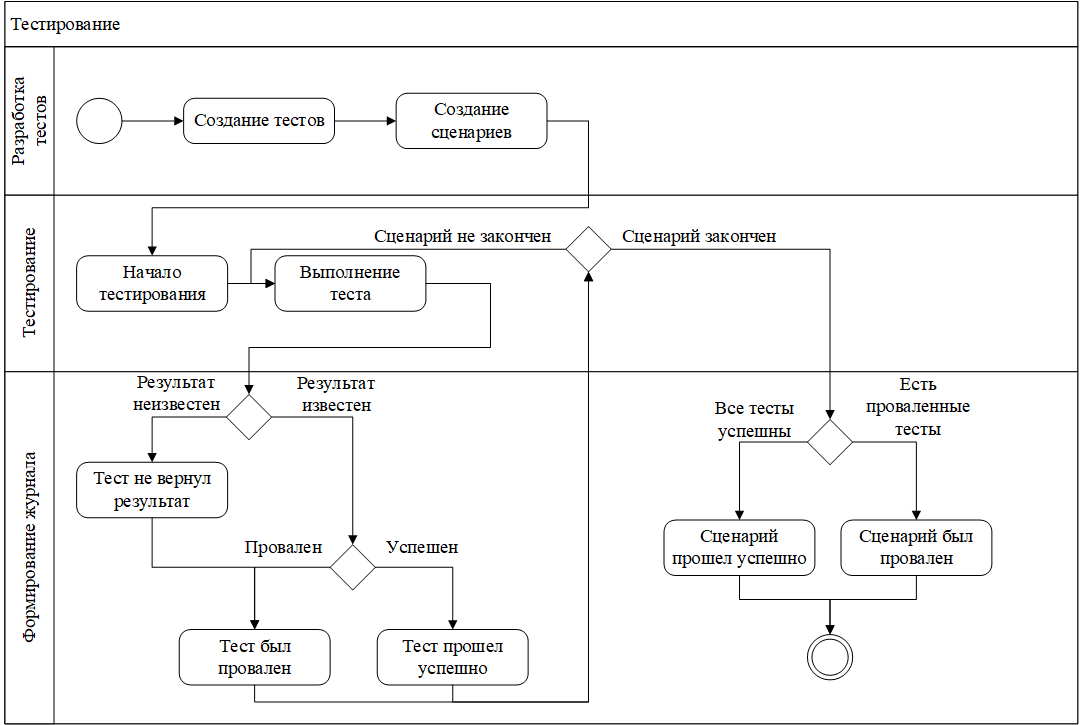


Рисунок 1.2.3 – Диаграмма деятельности

1.3 Описание входной информации

Основным источником информации являются файлы категорий, плейлистов и тестов, хранящиеся в собственных директориях, имеющие иерархическую структуру, представленную на рисунке 1.3.1. Полный путь до директории с файлами передается как параметр запуска главной панели программного продукта с ключом «$DataPath:», Например, «$DataPath:C:\TesterData\».

Описание входных файлов представлено в таблице 1.3.1.

Таблица 1.3.1 – Описание входных файлов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование файла | Шифр файла | Тип файла | Источник поступления |
| Файл теста | \*.txt | Текстовый | Разработчик |
| Файл категории | \*.txt | Текстовый | Разработчик |
| Файл плейлиста | \*.txt | Текстовый | Разработчик |

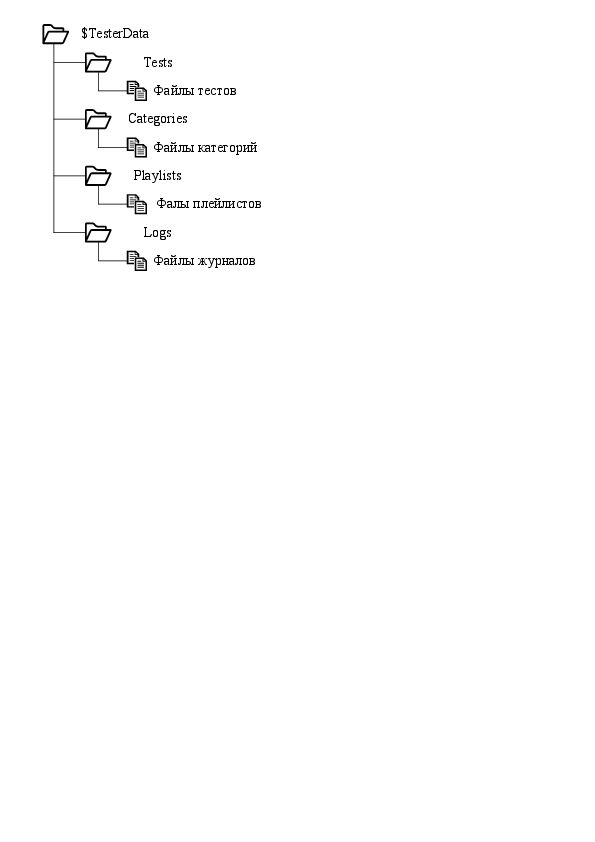


Рисунок 1.3.1 – Иерархия директорий

1.4 Описание выходной информации

Выходной информацией являются файлы категории, плейлисты, тестов, которые были созданы или отредактированы в интерфейсе программного продукта. А также во время проигрывания сценария тестов и после его завершения формируются результаты выполнения. Эти результаты отображаются в специальном поле «Журнал». Его можно сохранить в файл в директории журналов, присвоив ему имя. Описание выходного файла представлено в таблице 1.4.1. Здесь после таблицы описание выходного файла надо таблицу описание реквизитов выходного файла (см методичку)

Таблица 1.4.1 – Описание выходных файлов

|  |  |  |  |
| --- | --- | --- | --- |
| Название файла | Шифр файла | Тип файла | Источник приема |
| Журнал | \*.txt | Текстовый | - |

1.5 Общие требования к программному продукту

Программный продукт разрабатывается как модуль-подпроект для SCADA систем, разворачивать его необходимо непосредственно на стенде с тестируемой системой. В связи с этим воспроизведение тестов может привести к сбою в работе тестируемойсистемы и его алгоритмов, рекомендуется запускать программный продукт и воспроизводить сценарии тестов на резервном стенде (при наличии), выведенном из работы. Перед проведением тестирования рекомендуется выполнить процедуру резервного копирования данных тестируемойсистемы.

Для корректной работы стенд должен иметь следующие параметры, или выше:

* микропроцессор Pentium M 1.6GHz или совместимый;
* оперативная память объемом 2 Гбайт;
* HDD, 40 Гбайт;
* операционную систему от Windows Server 2003 SP2 или от Windows XP;
* среду разработки HMIWinCCOA, версии 3.14;

Минимальное количество персонала, требуемого для работы программного продукта, должно составлять не менее 1 штатной единицы – тестировщик или интегратор или разработчик. Пользователь должен иметь среднетехническое профильное образование, сертификаты компании, опыт программирования на языке Control.

1.6 Описание структуры файлов

Все файлы программного продукта должны храниться в иерархической структуре, изображенной на рисунке 1.6.1. Расширение файлов должно быть \*.txt. Директория «Tests» предназначена для хранения тестовых файлов,«Categories» - категорий, «Playlists» - плейлистов, «Logs» - журналов результатов выполнения тестов.

Скрипты тестов разрабатываются на языке программирования Control. При создании, редактировании необходимо учитывать то, что скрипт теста должны возвращать результат выполненияследующими константами:

* CodePlayer\_TestSucceeded – тестпрошелуспешно;
* CodePlayer\_TestFailed– тестбыл провален.

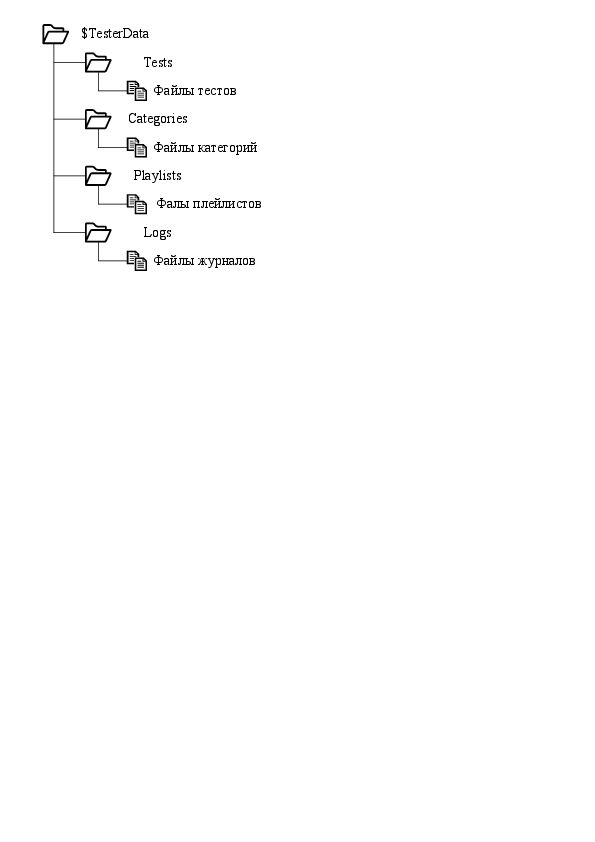


Рисунок 1.6.1 – Иерархия файлов

Иначе после выполнения скрипта тест будет считаться как неуспешным, и в журнал результатов выполнения тестов выйдет соответствующая запись о том, что результат теста неизвестен.

При создании, редактировании плейлистов необходимо учитывать то, что тесты будут воспроизводиться в той последовательности, в которой они записаны в файле плейлиста, а также каждый идентификатор должен находиться на новой строчке, в противном случае возникнет ошибка чтения файла

При создании, редактировании категорий порядок тестов в файле не имеет значения, можно располагать в любой последовательности, но также необходимо учитывать, что каждый идентификатор теста должен находиться на новой строчке, в противном случае возникнет ошибка чтения файла.

1.7 Контрольный пример

Входные файлы для контрольного примера представлены в таблице 1.7.1, содержание файлов находится в Приложении А.

Параметр запуска - полный путь до директории с файлами «D:\TesterData\», «$DataPath:D:\TesterData\».

Таблица 1.7.1 – Входные файлы контрольного примера

|  |  |  |
| --- | --- | --- |
| Наименование | Директория | Тип файла |
| test1.txt | \Tests | Тест |
| test2.txt | \Tests | Тест |
| test3.txt | \Tests | Тест |
| test4.txt | \Tests | Тест |
| test5.txt | \Tests | Тест |
| test6.txt | \Tests | Тест |
| category1.txt | \Categories | Категория |
| category2.txt | \Categories | Категория |
| category3.txt | \Categories | Категория |
| playlist1.txt | \Playlists | Плейлист |
| playlist2.txt | \Playlists | Плейлист |

Выходные данные:

1. при воспроизведении плейлиста с названием «playlsit1» в журнале результатов должны появиться следующиесообщения:
   1. тест «test1»- прошел успешно;
   2. тест «test2» - прошел успешно;
   3. тест «test3»- прошел неуспешно;
2. при воспроизведении плейлиста с названием «playlist2» в журнале результатов должны появится следующие сообщения:
3. тест «test4» - прошел неуспешно;
4. тест «test5» - прошел, неизвестный результат;
5. тест «test6» - прошел успешно.

2 Экспериментальный раздел

2.1 Описание программного продукта

В качестве среды для разработки приложения была выбрана система разработки HMIWinCC OA, так как это было главным критерием. Она основана на программном обеспечении от компании Siemens. В качестве языка программирования был выбран язык Control, который входит в систему WinCC OA. Также необходимо было соблюдать методологию разработки программного обеспечения«Разработка через тестирование» (TestDrivenDevelop), заключающиеся в разработке через тестирование функционала, реализовать архитектуру разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента «Модель-Представление-Контроллер» (Model-View-Controller), структура данной архитектуры отображена на рисунке 2.1.1. В качестве системы контроля версии использовался ApacheSubversion, разработанный ApacheSoftwareFoundation.



Рисунок 2.1.1 – Структура архитектуры  
«Модель-Представление-Контроллер»

Запуск программного продукта осуществляется путем открытия главной панели «tester/user.main.xml», с параметром запуска «$DataPath:», который указывает на директорию с файлами.

Описание модулей и их процедур представлено в таблице 2.1.1.

Таблица 2.1.1– Описание модулей и их процедур

|  |  |
| --- | --- |
| Процедуры | Назначение |
| 1 | 2 |
| player.ui – Модуль работы с интерфейсом пользователя | |
| PlayerUi\_InitEvent | Инициализация интерфейса |
| PlayerUi\_ClearEvent | Очищение интерфейса |
| PlayerUi\_ShowDataEvent | Отображение данных |
| PlayerUi\_UpdateDataEvent | Обновление данных в интерфейсе |
| PlayerUi\_SelectLogFileEvent | Выбор файла журнала |
| PlayerUi\_AppendMessageToLogEvent | Добавление сообщения в журнал |
| PlayerUi\_TurnOnPlayingModeEvent | Включение режима воспроизведения для интерфейса |
| PlayerUi\_TurnOffPlayingModeEvent | Выключение режима воспроизведения для интерфейса |
| PlayerUi\_TurnOnBreakpointModeEvent | Включение режима ожидания точки остановки |
| PlayerUi\_TurnOffBreakpointModeEvent | Выключение режима ожидания точки остановки |
| PlayerUi\_TestExecutionEvent | Отображение информации о воспроизводимом тесте |
| PlayerUi\_ShowMessage\_CreateTestEvent | Сообщение создания теста |
| PlayerUi\_ShowMessage\_RenameTestEvent | Сообщение переименования теста |
| PlayerUi\_ShowMessage\_DeleteTestsEvent | Сообщение удаления теста |
| PlayerUi\_ShowMessage\_DeleteTestsFromCategoryEvent | Сообщение удаления теста из категории |
| PlayerUi\_ShowMessage\_DeleteTestsFromPlaylistEvent | Сообщение удаления теста из плейлиста |
| PlayerUi\_ShowMessage\_CreateCategoryEvent | Сообщение создания категории |
| PlayerUi\_ShowMessage\_DeleteCategoryEvent | Сообщение удалении категории |
| PlayerUi\_ShowMessage\_CreatePlaylistEvent | Сообщение создания плейлиста |

Продолжение таблицы 2.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| PlayerUi\_ShowMessage\_DeletePlaylistEvent | Сообщение удаления плейлиста |
| PlayerUi\_ShowMessage\_SaveLogEvent | Сообщение сохранения журнала |
| data.mgr – Модуль работы с данными | |
| DataMgr\_InitEvent | Инициализация модуля |
| DataMgr\_ClearEvent | Очищение модуля |
| DataMgr\_SetSelectedLogFileEvent | Установки файла журнала |
| DataMgr\_SaveLogEvent | Сохранение журнала в файл |
| DataMgr\_CreateTestEvent | Создание теста |
| DataMgr\_DeleteTestsEvent | Удаление теста |
| DataMgr\_RanameTestEvent | Переименование теста |
| DataMgr\_RanameTestWithReplaceEvent | Переименование теста с заменой |
| DataMgr\_SetTestCodeEvent | Сохранение скрипта теста |
| DataMgr\_CreateCategoryEvent | Создание категории |
| DataMgr\_DeleteCategoryEvent | Удаление категории |
| DataMgr\_AppendTestToCategoryEvent | Добавление теста в категорию |
| DataMgr\_DeleteTestsFromCategoryEvent | Удаление теста из категории |
| DataMgr\_CreatePlaylistEvent | Создание плейлиста |
| DataMgr\_DeletePlaylistEvent | Удаление плейлиста |
| DataMgr\_AppendTestsToPlaylistEvent | Добавление теста в плейлист |
| DataMgr\_DeleteTestsFromPlaylistEvent | Удаление теста из плейлиста |
| DataMgr\_StartPlayingEvent | Начало воспроизведения сценария тестов |
| DataMgr\_PlaylistExeutionStartedEvent | Воспроизведение сценария тестов началось |
| DataMgr\_TestExecutionStartedEvent | Выполнение теста началось |
| DataMgr\_TestExecutionEndedEvent | Выполнение теста завершилось |
| DataMgr\_PlaylistExecutionEndedEvent | Выполнение сценария теста завершилось |
| DataMgr\_PauseOnBreakpointEvent | Остановка выполнения теста на точке остановки |
| DataMgr\_UnpauseOnBreakpointEvent | Продолжение выполнения теста после точки остановки |
| DataMgr\_Checking\_CreateTestEvent | Проверка создания теста |

Продолжение таблицы 2.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| DataMgr\_Checking\_RenameTestEvent | Проверка переименования теста |
| DataMgr\_Checking\_CreatePlaylistEvent | Проверка создания плейлиста |
| DataMgr\_Checking\_SaveLogEvent | Проверка сохранения журнала |
| code.player – Модуль воспроизведения скриптов тестов | |
| CodePlayer\_InitEvent | Инициализация модуля |
| CodePlayer\_ShiftBreakpointStateEvent | Смена состояния ожидания точки остановки теста |
| CodePlayer\_StartExecutionEvent | Начало выполнения сценария тестов |
| CodePlayer\_StopExecutionEvent | Остановки выполнения сценария тестов |
| CodePlayer\_PauseOnBreakpointEvent | Остановка выполнения теста на точке остановки |
| CodePlayer\_UnpauseOnBreakpointEvent | Продолжение выполнения теста после точки остановки |
| event.mgr – Модуль реагирования на события в модулях | |
| Tester\_InitEvents | Регистрация всех событий модулей |
| Tester\_Setup | Страт модуля |
| EventMgr\_InitProjEvent | Инициализация модуля |
| EventMgr\_ClearProjEvent | Очистка модуля |
| EventMgr\_ChangeLangEvent | Изменение языка модуля |
| EventMgr\_CancelMessageEvent | Обработка события «отмена» в интерфейсе |

2.2 Протокол тестирования

В ходе тестирования программного продукта на корректных и некорректных данных не было обнаружено ошибок, которые влияли бы на работу самого программного продукта и всей системы.

Разработанный программный продукт удовлетворяет всем функциональным требованиям, имеет дружественный интерфейс, исключает появлении системных ошибок.

При тестировании программного продукта на данных контрольного примера создание, редактирование, удаление категорий, плейлистов, тестов, а также воспроизведение сценариев происходит корректно.

Результаты тестирования полностью совпадают с ожидаемыми результатами, представленными на рисунках2.2.1 и 2.2.2.

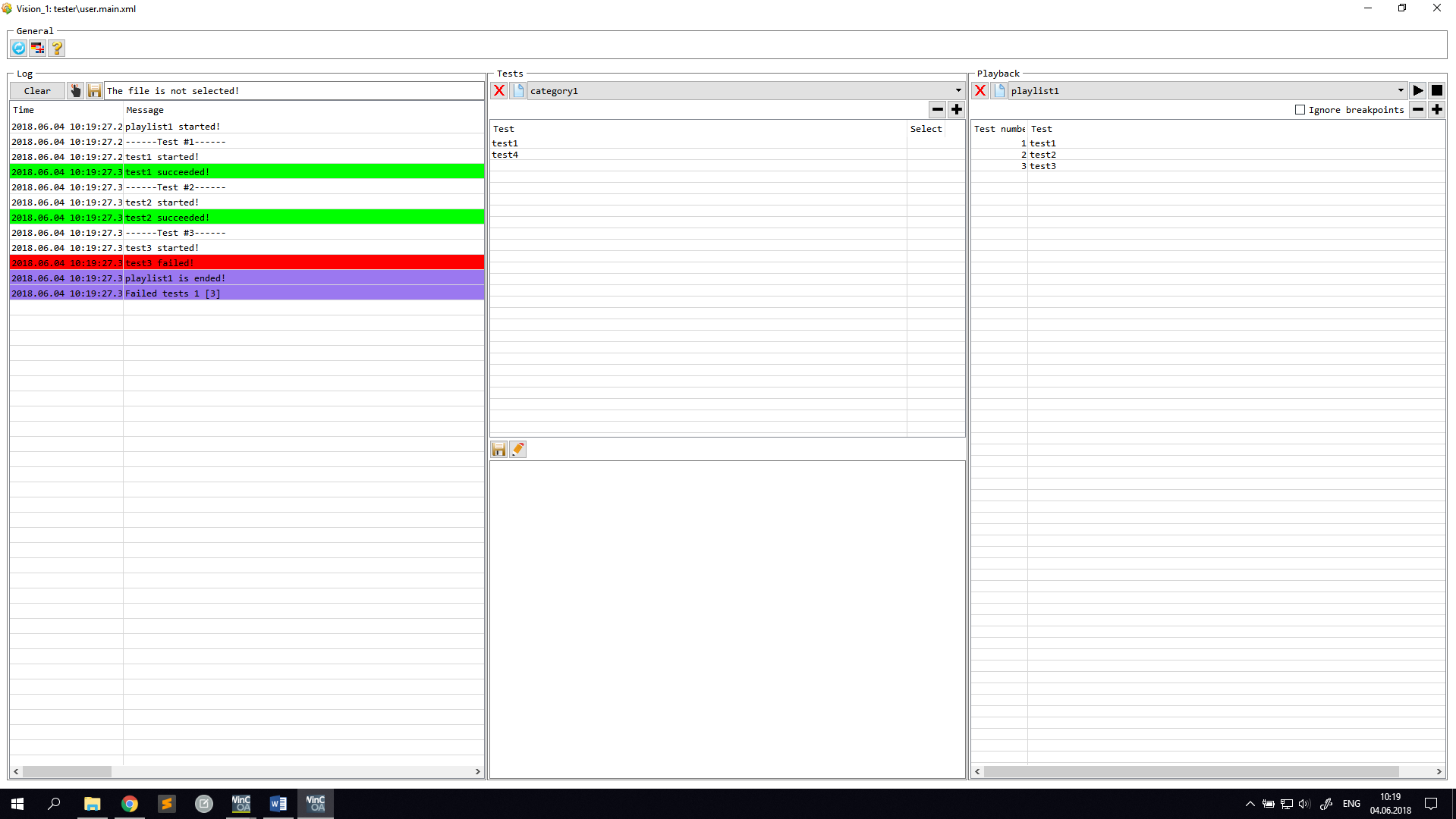


Рисунок 2.2.1 – Результат тестирования воспроизведения сценария тестов

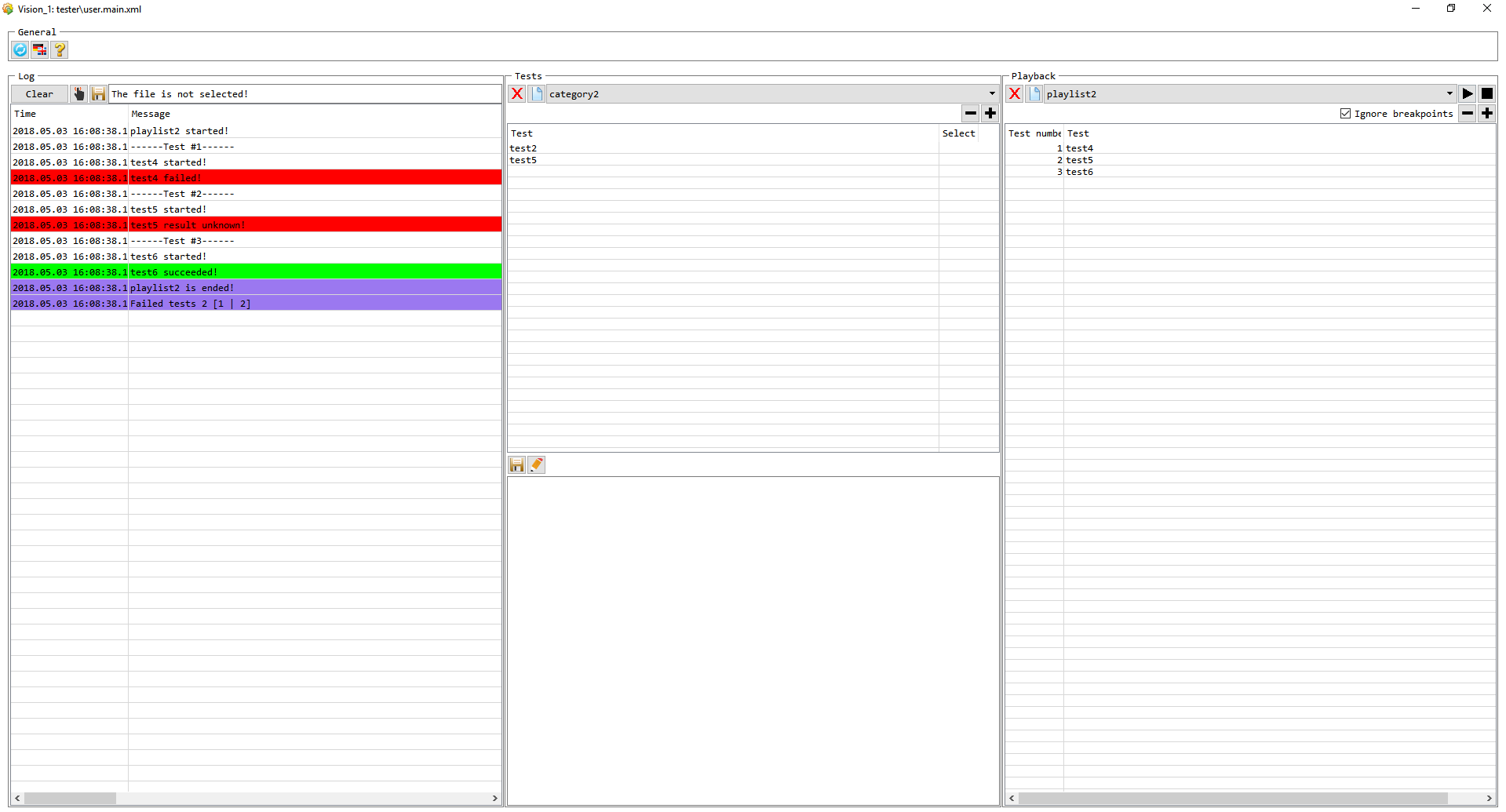


Рисунок 2.2.2 – Результат тестирования воспроизведения сценария тестов

2.3 Руководство пользователя

Программный продукт является инструментом для проверки функционирования системы, её менеджеров и алгоритмов.

Тестирование выполняется посредством запуска проверок из интерфейса программного продукта, в который также выводится информация о воспроизведении сценариев тестов и результаты проверки.

Все файлы для тестера подгружаются из каталога ресурсов, который содержит в себе четыре обязательных раздела:

1. категории (categories/) (рисунок 2.3.1).

Данный раздел предназначен для хранения категорий, которые будут доступны в выпадающем списке области «Тесты».

Раздел включает в себя перечень категорий в формате \*.txt, содержимым которых является список входящих в категорию тестов.

Список тестов формируется из наименований файлов тестов.

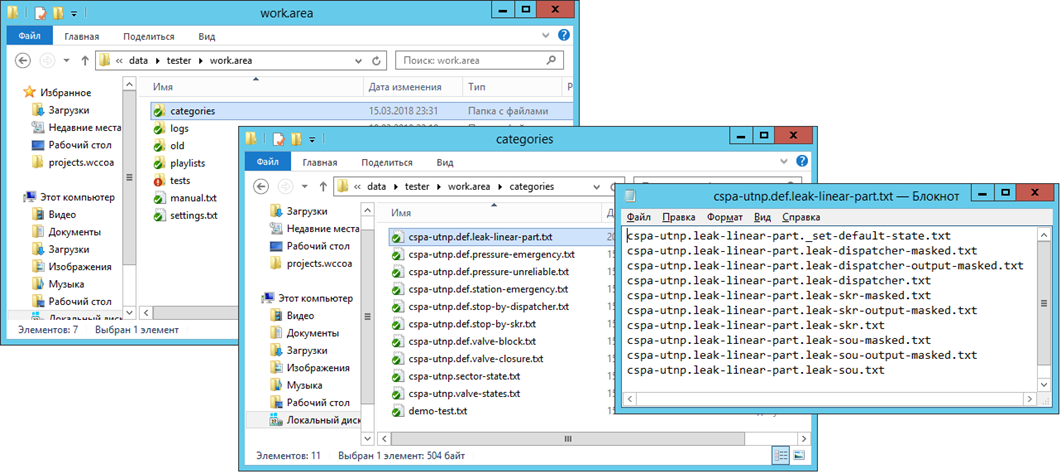


Рисунок 2.3.1 – Раздел категорий с тестами

1. логи (logs/).

Раздел предназначен для хранения данных таблицы с результатами выполненных плейлистов.

1. плейлисты (playlists/).

Данный раздел предназначен для хранения плейлистов, которые будут доступны в выпадающем списке области «Воспроизведение».

Раздел включает в себя перечень плейлистов в формате \*.txt, содержимым которых является список входящих в плейлист тестов.

Список тестов формируется из наименований файлов тестов.

1. тесты (tests/).

Данный раздел предназначен для хранения всех тестов, которые можно добавить в категории и плейлисты, с последующей возможностью их воспроизведения.

Каждый тест представляет собой сценарий, написанный на языке CTRL, который будет воспроизведен при запуске тестирования.

Также содержимое сценария можно просмотреть и отредактировать в самом интерфейсе программного продукта.

Файл с настройками цветов сообщений, отображаемых в журнале, находится в каталоге colorDB. Файл доступен для редактирования в текстовом редакторе.

Файлы с настройками текстов сообщений, отображаемых в журнале, находятся в каталогах msg\en\_US.utf8 (для английского языка) и msg\ru\_RU.utf8 (для русского языка). Файлы также доступны для редактирования в текстовом редакторе.

Интерфейс главной панели представлен на рисунке 2.3.2.

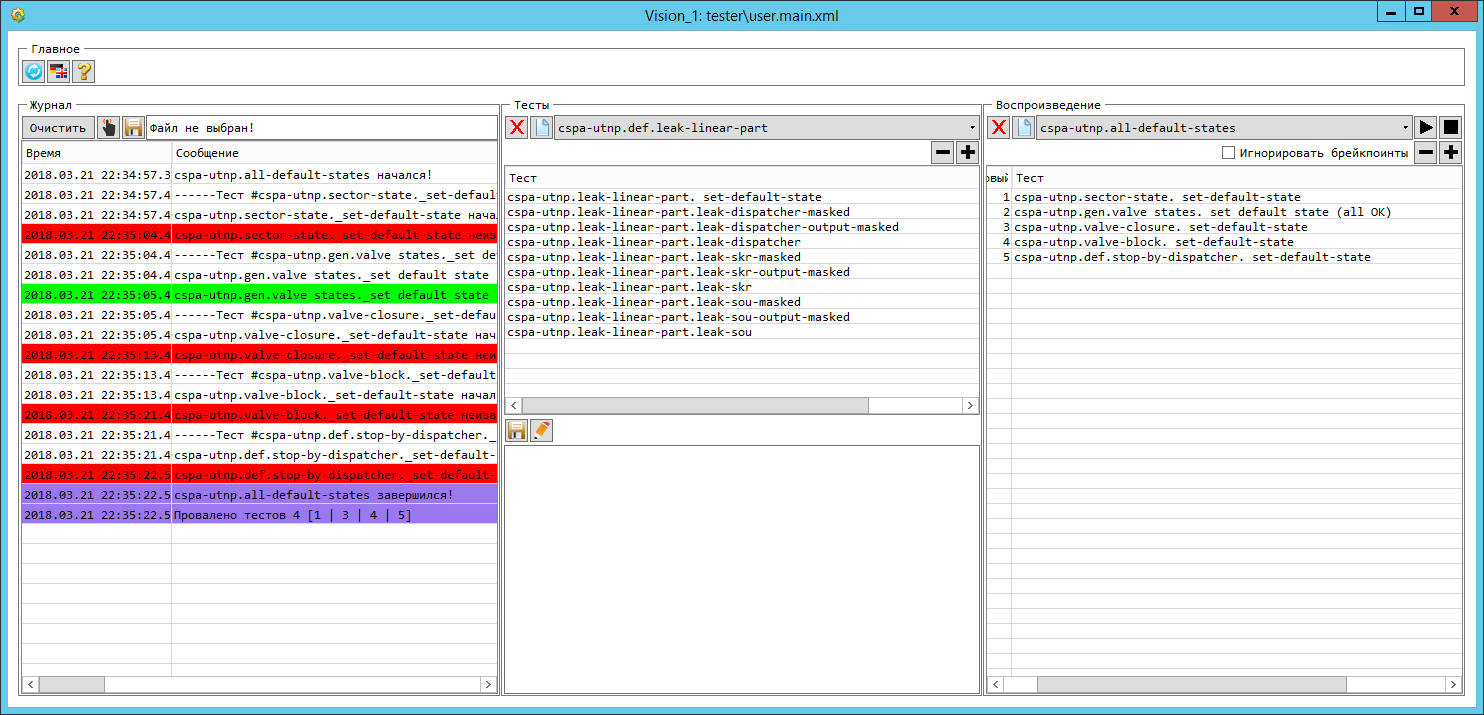


Рисунок 2.3.2 – Интерфейс главной панели

Панель состоит из следующих областей:

1. «Главное». Содержит кнопку обновления данных , используемую при необходимости перезагрузки данных с каталога (диска), кнопку смены языка  и кнопку вызова окна со справкой ;
2. «Журнал». В данной области выводится информация о ходе выполнения тестов и их результаты. Журнал представлен в виде таблицы со столбцами, предназначенными для отображения времени выполнения шага проверки и сообщения с результатом выполнения проверки на данном шаге. Кнопки, расположенные в верхней части, позволяют выполнить очистку таблицы () и сохранение результатов тестирования (→). В строке отображается имя файла, выбранного для сохранения данных таблицы;
3. «Тесты». В данной области содержатся категории с тестами. Выбрать категорию можно из выпадающего списка, при этом в таблице отобразятся все тесты, входящие в выбранную категорию. В области присутствуют кнопки, позволяющие создать  и удалить  категорию, а также кнопки, позволяющие создать  и удалить  тест из категории. При удалении теста, он останется в каталоге тестов, но будет удален в плейлистах и выбранной категории. Поле, расположенное ниже таблицы, предназначено для отображения кода выбранного теста. Поле позволяет редактировать код, также функция редактирования доступна по нажатию кнопки . Сохранение внесенных в код изменений выполняется нажатием кнопки ;
4. «Воспроизведение». В данной области содержатся плейлисты (сценарии) с тестами. Выбрать плейлист можно из выпадающего списка, при этом в таблице отобразятся все тесты, входящие в выбранный плейлист. В области присутствуют кнопки, позволяющие создать  и удалить  плейлист, а также кнопки, позволяющие добавить  и удалить  тест из плейлиста.

Установка флага «Игнорировать брейкпойнты» позволяет игнорировать прерывания, содержащиеся в скриптах тестов, и продолжать их воспроизведение. Воспроизведение выбранного плейлиста с тестами выполняется нажатием кнопки . Остановить воспроизведение можно нажав кнопку .

Программный продукт позволяет выполнить системные и интеграционные тесты.

Системное тестирование заключается в проверке функционирования системы и её алгоритмов.

Интеграционное тестирование проводится для проверки правильности совместной работы двух взаимозависимых глобальных функций. Интеграционные проверки выполняются с предварительным отключением всех других функций, которые могут повлиять на выполнение алгоритмов, заложенных в проверяемых глобальных функциях.

Процедуру тестирования в программном продуктеследует выполнять в следующем порядке:

1. в области «Воспроизведение» из выпадающего списка выбрать нужный плейлист (рисунок 2.3.3);

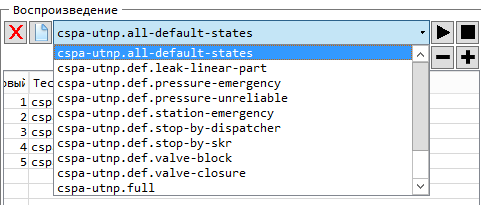


Рисунок 2.3.3 – Выбор плейлиста

1. при необходимости добавить, удалить тест. Для добавления теста необходимо выбрать ЛКМ нужный тест в области «Тесты» и нажать кнопку . Удаление теста из плейлиста осуществляется посредством выбора теста в области «Воспроизведение» и нажатия кнопки ;
2. при необходимости установить флаг «Игнорировать брейкпойнты»;
3. запустить плейлист нажатием кнопки . Начнётся воспроизведение плейлиста, во время которого будут подсвечиваться проигрываемые тесты (рисунок 2.3.4).

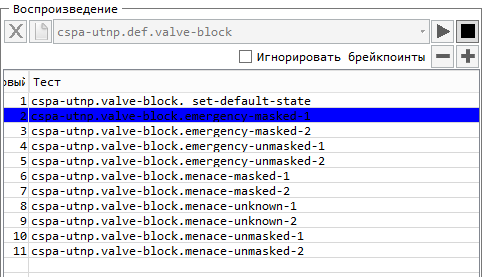


Рисунок 2.3.4 – Воспроизведение плейлиста

В ходе воспроизведения сценария тестов журнал будет заполняться информацией о выполнении тестирования. Детальное описание содержимого таблицы с сообщениями приведено на рисунке 2.3.5.

Перечень возможных отображаемых сообщений и цветовых решений приведен в таблице 2.3.1.



Рисунок 2.3.5 – Журнал с результатами выполнения сценария тестов

1. имя запускаемого теста в плейлисте;
2. сообщение о начале воспроизведения теста;
3. сообщение с результатом выполнения теста;
4. сообщение о принудительной остановке воспроизведения;
5. сообщение с результатом выполнения всего плейлиста.

Таблица 2.3.1 – Возможные отображаемые сообщения

|  |  |  |
| --- | --- | --- |
| № | Сообщение | Перевод |
| 1 | 2 | 3 |
| 1 | Waiting for signal started | Начато ожидание сигнала |
| 2 | Waiting for signal finished | Ожиданиесигналазавершено |
| 3 | Waiting for signal unsuccessful | Ожиданиесигналанекорректно |
| 4 | Testskipped | Тест пропущен, не был воспроизведен |
| 5 | Testsucceeded | Тест завершен с успехом |
| 6 | Testfailed | Тест завершен с провалом |
| 7 | Playlistsucceeded | Плейлист завершен с успехом |
| 8 | Playlistfailed | Плейлист завершен с провалом |

Воспроизведение плейлиста завершается выводом сообщения в журнал: «Плейлист завершен с успехом» или «Плейлист завершен с провалом».

Для сохранения данных таблицы с результатами проведенной проверки выполнить следующие действия:

1. нажать кнопку . В появившемся диалоговом окне выбрать ранее созданный файл либо создать новый, введя имя в строке «Имя файла» (рисунок 2.3.6). После выбора, создания файла нажать кнопку «Сохранить»;
2. путь к выбранному, созданному файлу отобразится в строке области «Журнал» (рисунок 2.3.7). Для сохранения данных таблицы в выбранный/созданный файл нажать кнопку .

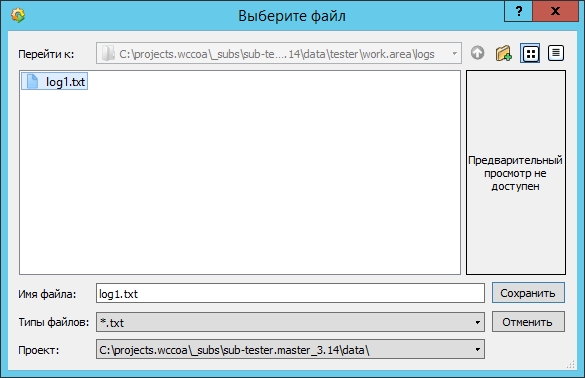


Рисунок 2.3.6 – Выбор, создание файла для сохранения журнала

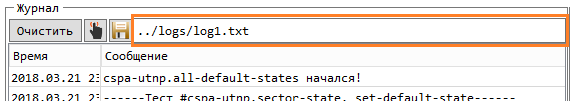


Рисунок 2.3.7 – Выбранный файл для сохранения журнала

Важно! При сохранении данных таблицы в ранее созданный файл выполняется добавление новых данных в этот файл.

Сохраненный файл с результатами тестирования представляет собой текстовый файл, строки которого соответствуют строкам таблицы (рисунок 2.3.8).

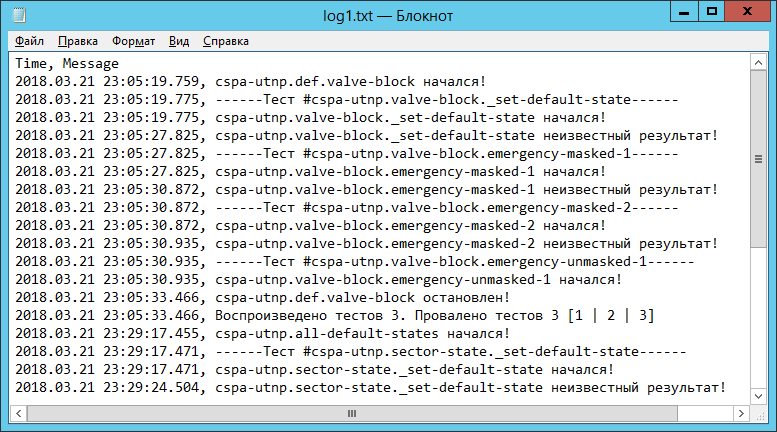


Рисунок 2.3.8 – Файл журнала

Редактирование скриптов тестов.

В интерфейсе программного продукта реализована возможность просмотра и редактирования скриптов тестов.

Область «Тесты» содержит поле (рисунок 2.3.9), предназначенное для отображения кода теста, выбранного в таблице сверху.

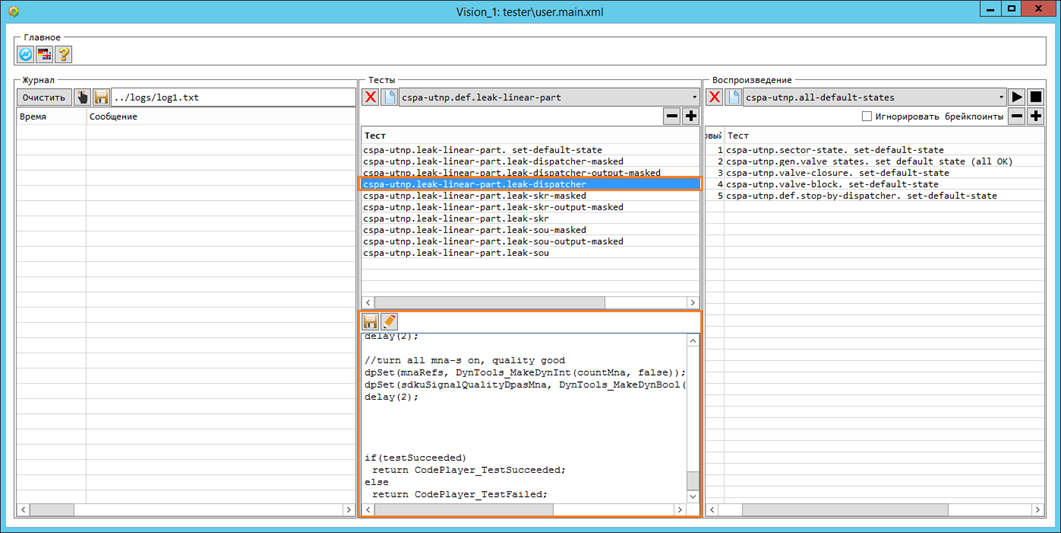


Рисунок 2.3.9 – Редактирование скриптов тестов

Поле позволяет редактировать код. Сохранение внесенных в код изменений выполняется нажатием кнопки .

Функция редактирования доступна также в текстовом редакторе, открывающемся по нажатию кнопки .

При создании, редактировании скриптов тестов необходимо учитывать то, что тесты должны возвращать результаты выполнения сценария следующими константами: CodePlayer\_TestSucceeded – тест прошел успешно, CodePlayer\_TestFailed– тест был провален.

Устранение неисправностей.

Перечень возможных проблем, возникающих при запуске и использовании программного продукта, а также методы их устранения приведены в таблице 2.3.2.

Таблица 2.3.2 – Возможные проблемы и методы их решения

|  |  |  |
| --- | --- | --- |
| № | Описание проблемы | Методы устранения |
| 1 | Программный продукт не запускается | Проверить файл конфигурации на предмет синтаксических и информационных ошибок |
| 2 | Программа открылась, но кнопки, элементы управления не работают либо отсутствуют | Проверить наличие подключенного подпроекта инструментов разработки (sub.tools).\*\*.  Проверить правильность указанного пути до каталога подпроекта (sub.tools).\*\* |
| 3 | Программа не выполняет процедуру тестирования | Убедиться в наличии активного подключения к тестируемому проекту |
| 4 | Тест завершился с ошибкой «unknown result» | Изучить лог тестируемого проекта на предмет системных сообщений о появившейся проблеме в ходе выполнения теста |

3 Экономический раздел

3.1 Расчет затрат на разработку

Расчет себестоимости машинного часа эксплуатации вычислительной и оргтехники (ВиОТ):

, (3.1.1)

где См.ч.– себестоимость машинного часа;

Зi– годовые затраты, связанные с эксплуатацией и обслуживанием ВиОТ;

Fп– годовой полезный фонд времени работы единицы оборудования;

Kг– коэффициент готовности.

, (3.1.2)

где Fн= 1943ч. (в 2018г.) – номинальныйгодовой фонд рабочего времени в часах;

– коэффициент, учитывающий время, затраченное на ремонт, настройку, обслуживание ВиОТ.

(ч.)

Для расчета годовых затрат, необходимо определить балансовую стоимость условного комплекта, необходимого для создания программного продукта. Состав условного комплекта перечислен в таблице 3.1.1.

, (3.1.3)

где – цена единицы условного комплекта;

– количество единиц условного комплекта;

– дополнительные расходы на доставку, установку, первоначальную наладку.

, (3.1.4)

Таблица 3.1.1 – Состав условного комплекта

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Кол-во, шт. | Цена, руб. |
| 1 | Ноутбук | 1 | 55290 |

Продолжение таблицы 3.1.1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Кол-во, шт. | Цена, руб. |
| 2 | Монитор | 2 | 16 490 |
| 3 | Клавиатура | 1 | 790 |
| 4 | Мышь |  | 790 |
|  | Итого: | | 89 850 |

(руб.).

(руб.).

Затраты на материалы:

, (3.1.5)

Основная и дополнительная заработная плата персонала, обслуживающего ВиОТ:

, (3.1.6)

где – оклад i-го работника, обслуживающего ВиТО;

– коэффициент дополнительной заработной платы;

– районный коэффициент;

= 20.

(руб.)

Начисления на заработную плату , подробный список перечислен в таблице 3.1.2.

, (3.1.7)

(руб.).

Таблица 3.1.2 – Состав единого страхового налога на 2018г.

|  |  |  |
| --- | --- | --- |
| № | Статья | % |
| 1 | Пенсионный фонд | 22 |
| 2 | Фонд социального страхования | 2,9 |
| 3 | Федеральный фонд обязательного медицинского страхования | 5,1 |
| 4 | Страховой налог от несчастного случая на производстве для служащих | 0,2 |
|  | Итого: | 30,2 |

Амортизационные отчисления ВиОТ:

, (3.1.8)

где – норма амортизационных отчислений.

(руб.).

Износ программных продуктов.

Условный комплект необходимых программных продуктов представлен в таблице 3.1.3:

Таблица 3.1.3 – Условный комплект программных продуктов

|  |  |
| --- | --- |
| Наименование | Цена (руб.) |
| MS Windows 10 | 14199 |
| SimaticWinCC OA | 150 000 |
| Sublime Text 3 | 2 205 |
| Visual Studio 2017 Professional | 25 000 |
| Итого: | 192 404 |

Амортизационные отчисления программных продуктов:

, (3.1.9)

где – суммарная стоимость программных продуктов;

– норма амортизационных отчислений.

(руб.)

Расходы на содержание и эксплуатацию оборудования.

Расходы на содержание и эксплуатацию оборудования состоят из:

* затраты на ремонт и специальное обслуживание

, (3.1.10)

* затраты на электрическую энергию

, (3.1.11)

где – суммарная мощность;

– коэффициент готовности;

– стоимость кВт/ч.

(руб.).

(руб.).

Расходы на содержание и эксплуатацию оборудования:

, (3.1.12)

(руб.).

Прочие расходы.

Так как используется один тип техники и количество обслуживающего персонала не превышает трех человек, следовательно:

, (3.1.13)

(руб.).

Суммарные годовые затраты:

, (3.1.14)

(руб.).

Себестоимость машинного часа, из формулы (3.1.1), составляет:

(руб.).

3.2 Расчет цены и минимального количества копий тиражирования

Фонд оплаты труда за время работы над программным продуктом – ФОТ:

, (3.2.1)

где – оклад сотрудника, работающего над продуктом;

– общее время работы над программным продуктом;

– коэффициент дополнительной заработной платы;

– районный коэффициент.

(руб.).

Начисления на ФОТ:

, (3.2.2)

(руб.).

Затраты, связанные с эксплуатацией и обслуживанием ВиОТ - :

, (3.2.3)

где – машинное время работы над программным продуктом;

– число рабочих часов в месяце;

– число условных комплектов;

– коэффициент использования оборудования.

(руб.).

Затраты на специальные программные продукты – (руб.), представлены в таблице 3.2.1.

Таблица 3.2.1 – Специальные программные продукты

|  |  |
| --- | --- |
| Наименование | Цена |
| SimaticWinCC OA | 150 000 |
| Visual Studio 2017 Professional | 25 000 |
| Итого: | 175 000 |

Затраты на хозяйственные операции и нужды – (руб.), представлены в таблице 3.2.2.

Таблица 3.2.2 – Затраты на хозяйственные нужды

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена за единицу, (руб.) | Кол-во | Всего |
| Бумага | 235 | 4 | 940 |
| Ручка | 25 | 2 | 50 |
| Блокнот | 360 | 1 | 360 |
| Карандаш | 25 | 3 | 75 |
| Итого: | | | 1425 |

Накладные расходы:

, (3.2.4)

(руб.)

Полные затраты на разработку программного продукта:

, (3.2.5)

(руб.)

Расчет установочной прибыли:

, (3.2.6)

где - установочная рентабельность.

(руб.).

Расчет величины налога на добавленную стоимость (НДС):

, (3.2.7)

(руб.).

Цена предложения разрабатываемого программного продукта:

, (3.2.8)

(руб.).

Расчет минимального количества копий, необходимых для реализации, для получения установочной прибыли:

, (3.2.9)

где – цена на рынке одной копии программного продукта;

– затраты на тиражирование одной копии программного продукта.

, (3.2.10)

где – цена носителя программного продукта;

– время копирования.

(руб.).

шт.

ЗАКЛЮЧЕНИЕ

В процессе выполнения дипломного проекта был разработан программный продукт «Среда тестирования в WinCCOA».

При этом была изучена среда разработки человеко-машинного интерфейса SimaticWindowsControlCenterOpenArchitecture. Также были изучены такие технологии разработки ПО такие как:

* TDD(TestDrivenDevelopment) – разработкачерезтестирование,
* MVC (Model-View-Control) – разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента;
* система контроля версии ApacheSubversion.

Программный продукт написан на языке Control, который встроен в среду разработки HMIWinCCOA.

Была проведено эксплуатационное и отладочное тестирование программного продукта. По результатам отладочного тестирования были устранены некоторые недостатки, в частности были обнаружены и исправлены неточности в реализации. После этого было написано руководство пользователя.

С помощью программного продукта на основании данных контрольного примера были получены результаты, которые полностью совпадают с выходной информацией контрольного примера.

Приложение А

Содержимое файлов контрольного примера

Содержимое файлаtest1.txt:

*float a = 5.7;*

*float b = 3.3;*

*int c = a + b;*

*if (c == 9)*

*return CodePlayer\_TestSucceeded;*

*else*

*return CodePlayer\_TestFailed;*

Содержимоефайлаtest2.txt:

*float a = 5.3;*

*float b = 3.7;*

*int c = a - b;*

*if (c == 1)*

*return CodePlayer\_TestSucceeded;*

*else*

*return CodePlayer\_TestFailed;*

Содержимоефайлаtest3.txt:

*float a = 1.1;*

*float b = 3.3;*

*int c = a \* b;*

*if (c == 4)*

*return CodePlayer\_TestSucceeded;*

*else*

*return CodePlayer\_TestFailed;*

Содержимоефайлаtest4.txt:

*float a = 3.3;*

*float b = 0.7;*

*int c = a / b;*

*if (c == 5)*

*return CodePlayer\_TestSucceeded;*

*else*

*return CodePlayer\_TestFailed;*

Продолжение приложения А

Содержимое файла test5.txt:

*float a = 3;*

*float b = 0;*

*int c = a / b;*

*if (c == 0)*

*return CodePlayer\_TestSucceeded;*

*else*

*return CodePlayer\_TestFailed;*

Содержимое файла test6.txt:

*float a = 0;*

*float b = 3;*

*int c = a / b;*

*if (c == 0)*

*return CodePlayer\_TestSucceeded;*

*else*

*return CodePlayer\_TestFailed;*

Содержимое файла category1.txt:

*test1.txt*

*test4.txt*

Содержимое файла category2.txt:

*test2.txt*

*test5.txt*

Содержимое файла category3.txt:

*test3.txt*

*test6.txt*

Содержимое файла palylist1.txt:

*test1.txt*

*test2.txt*

*test3.txt*

Продолжение приложения А

Содержимое файла playlist2.txt:

*test4.txt*

*test5.txt*

*test6.txt*

Приложение Б

Исходный код программного продукта

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 2.104-2006. Единая система конструкторской документации (ЕСКД). Основные надписи (с Поправками);
2. ГОСТ 7.1–2003. Библиографическая запись. Библиографическое описание;
3. ГОСТ 2.105-95. Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам;
4. ГОСТ 2.316-08. Единая система конструкторской документации (ЕСКД). Правила нанесения на чертежах надписей технических требований и таблиц;
5. ГОСТ Р7.0.12-2011. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила;
6. ГОСТ 7.82-01. Единая система конструкторской документации (ЕСКД). Библиографическая запись. Библиографическое описание электронных ресурсов;
7. ГОСТ 7.1-03. Единая система конструкторской документации (ЕСКД). Библиографическая запись. Библиографическое описание;
8. Роберт Мартин. Чистый код. Создание, анализ и рефакторинг [Текст] / Роберт Мартин –АпперСадл Ривер, Нью-Джерси, США: ПрентисХол, 2010. –464с;
9. Роберт Мартин. Чистый кодер: кодекс поведения для профессиональных программистов [Текст] / Роберт Мартин –АпперСадл Ривер, Нью-Джерси, США: ПрентисХол, 2011. –242с;
10. Роберт Мартин. Чистая архитектура: руководство мастера по структуре и дизайну программного обеспечения [Текст] / Роберт Мартин –АпперСадл Ривер, Нью-Джерси, США: ПрентисХол, 2017. –432с;
11. Роберт Мартин. Threelawsof TDD [Электронный ресурс]: Роберт Мартин –Режим доступа: URL: https://www.youtube.com/watch?v=qkblc5WRn-U;
12. Роберт Мартин. ThefutureofprogrammingРобертМартин –Режимдоступа: URL: https://www.youtube.com/watch?v=ecIWPzGEbFc;
13. ЭванДорн. Test-driven development: Write better code in less time [Электронныйресурс]: ЭванДорн –Режимдоступа: URL: https://www.youtube.com/watch?v=HhwElTL-mdI;
14. МискоХевери. Google's clean code talks [Электронныйресурс]: МискоХевери –Режимдоступа: URL:

https://www.youtube.com/playlist?list=PL693EFD059797C21E;

1. Мишель Нортон. Thetechnicaldebttrap [Электронный ресурс]: Мишель Нортон –Режим доступа: URL: https://www.youtube.com/watch?v=SfWCRl75Kas;
2. Рич Хиккей. Simplemadeeasy [Электронный ресурс]: Рич Хиккей –Режим доступа: URL: https://www.infoq.com/presentations/Simple-Made-Easy;
3. Свободная энциклопедия Википедия. Статья «Model-View-Controller» [Электронный ресурс]: Свободная энциклопедия Википедия –Режим доступа: URL: https://ru.wikipedia.org/wiki/Model-View-Controller;
4. Свободная энциклопедия Википедия. Статья «Разработка через тестирование» [Электронный ресурс]: Свободная энциклопедия Википедия –Режим доступа: URL:

https://ru.wikipedia.org/wiki/Разработка\_через\_тестирование.