

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт  
Факультет приборостроения,  
информационных технологий и  
электроники

Кафедра  
«Информационная безопасность  
систем и технологий»

Специальность 10.05.02 –

Специализация -

Информационная безопасность  
телекоммуникационных систем  
Разработка защищенных  
телекоммуникационных систем

ДИПЛОМНЫЙ ПРОЕКТ

на тему

МЕТОДИКА ВОССТАНОВЛЕНИЯ АЛГОРИТМА ПРОГРАММ ДЛЯ  
ОЦЕНКИ ВОЗМОЖНОСТИ ПОЯВЛЕНИЯ ОПАСНЫХ СОБЫТИЙ В  
ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ СПЕЦИАЛЬНОГО  
НАЗНАЧЕНИЯ

Студент \_\_\_\_\_ Цыбизов Михаил Алексеевич

Руководитель \_\_\_\_\_ Боровиков А.Ю.

Нормоконтролёр \_\_\_\_\_ Фатеев А.Г.

Рецензент \_\_\_\_\_ Давыдов Д.В.

Работа допущена к защите (протокол заседания кафедры от \_\_\_\_\_ № \_\_\_\_\_)

Заведующий кафедрой \_\_\_\_\_ С.Л. Зефилов

Работа защищена с отметкой \_\_\_\_\_ (протокол заседания ГЭК  
от \_\_\_\_\_ № \_\_\_\_\_)

Секретарь ГЭК \_\_\_\_\_ А.Г. Фатеев

Пенза 2017

## РЕФЕРАТ

Отчет 110 с., 18 рис., 8 источников, 2 прил.

ТЕЛЕКОММУНИКАЦИОННАЯ СИСТЕМА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ, СХЕМА ИНФОРМАЦИОННЫХ ПОТОКОВ, ВОССТАНОВЛЕНИЕ АЛГОРИТМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ОПАСНОЕ СОБЫТИЕ

Объектом дипломного проекта является программа технического средства.

Целью дипломного проекта является разработка методики восстановления алгоритма программ для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения.

В процессе работы была разработана структура телекоммуникационной системы специального назначения, приведено обоснование необходимости восстановления алгоритма работы программного обеспечения технического средства, методы поиска опасных конструкций в программном обеспечении, описание используемых при восстановлении алгоритма работы программных средств, выбрано техническое средство для восстановления алгоритма его программы, описан процесс восстановления алгоритма работы программы на примере выбранного технического средства, сделаны выводы по результатам восстановления алгоритма программы.

В результате работы была разработана методика восстановления алгоритма программ для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения.

## СОДЕРЖАНИЕ

|                                                                                                                   |    |
|-------------------------------------------------------------------------------------------------------------------|----|
| Введение.....                                                                                                     | 8  |
| 1 Типовая архитектура построения телекоммуникационной системы специального назначения.....                        | 11 |
| 2 Выбор технического средства и обоснование необходимости восстановления алгоритма работы его программы .....     | 16 |
| 2.1 Выбор технического средства.....                                                                              | 16 |
| 2.2 Обоснование необходимости восстановления алгоритма работы программного обеспечения технического средства..... | 18 |
| 2.2.1 Проблема защиты программ.....                                                                               | 18 |
| 2.2.2 Ограничения Руководящего документа.....                                                                     | 19 |
| 2.2.3 Инструментальная база испытаний.....                                                                        | 21 |
| 2.3 Методы поиска опасных конструкций в программном обеспечении.....                                              | 22 |
| 3 Описание используемых при восстановлении алгоритма работы программных средств.....                              | 29 |
| 3.1 Среда разработки Eclipse.....                                                                                 | 29 |
| 3.2 Плагин средства контроля версий Git.....                                                                      | 31 |
| 3.3 Плагин системы документирования программного кода Doxygen.....                                                | 33 |
| 3.4 Плагин пакета утилит по автоматической визуализации графов Graphviz.....                                      | 35 |
| 4 Описание процесса настройки и использования IDE Eclipse в процессе восстановления алгоритма программы.....      | 38 |
| 4.1 Установка и настройка IDE Eclipse.....                                                                        | 38 |
| 4.2 Работа с проектами в IDE Eclipse.....                                                                         | 38 |
| 4.3 Конфигурация плагина Doxygen.....                                                                             | 39 |
| 4.4 Общие сведения о Doxygen.....                                                                                 | 40 |
| 4.5 Использование средства документирования кода Doxygen.....                                                     | 44 |
| 4.6 Полезные функции IDE Eclipse.....                                                                             | 49 |

|                                                                                                               |     |
|---------------------------------------------------------------------------------------------------------------|-----|
| 5 Методика восстановления алгоритма программы.....                                                            | 55  |
| 5.1 Общие положения.....                                                                                      | 55  |
| 5.2 Цель методики.....                                                                                        | 55  |
| 5.3 Основные положения методики.....                                                                          | 55  |
| 5.3.1 Развертывание необходимого программного обеспечения.....                                                | 55  |
| 5.3.2 Создание и настройка проекта исходного кода программного обеспечения.....                               | 56  |
| 5.3.3 Документирование исходного кода.....                                                                    | 56  |
| 5.3.4 Редактирование отчета.....                                                                              | 58  |
| Заключение.....                                                                                               | 61  |
| Список использованных источников.....                                                                         | 62  |
| Приложение А. Листинг макроса, редактирующего отчет в соответствии с требованиями группы стандартов ЕСПД..... | 64  |
| Приложение Б. Протокол проверки ВКР на оригинальность.....                                                    | 110 |

## ОПРЕДЕЛЕНИЯ

В настоящем отчете применяют следующие термины с соответствующими определениями.

Автоматизированная система – система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций [1].

Сеть специального назначения – сеть электросвязи, создаваемая для производственных и специальных потребностей органов государственной власти [2].

Информационные ресурсы – отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах [1].

Контролируемая зона – пространство (территория, здание, часть здания), в котором исключено неконтролируемое пребывание лиц, не имеющих постоянного или разового допуска, и посторонних транспортных средств [1].

Конфиденциальная информация – документированная информация, доступ к которой ограничивается в соответствии с законодательством Российской Федерации [1].

Межсетевой экран – локальное (однокомпонентное) или функционально-распределенное программное (программно-аппаратное) средство (комплекс), реализующее контроль за информацией, поступающей в АС и (или) выходящей из АС [1].

Защита информации – деятельность по предотвращению утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию [3].

Аудит – процесс записи информации о происходящих с каким-то объектом (или в рамках какого-то процесса) событиях в журнал (например, в файл) [3].

Опасное событие – событие, в результате которого может быть причинен вред [5].

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

|        |   |                                                      |
|--------|---|------------------------------------------------------|
| АРМ    | – | автоматизированное рабочее место                     |
| АС     | – | автоматизированная система                           |
| БИ     | – | безопасность информации                              |
| ЗИ     | – | защита информации                                    |
| КИ     | – | конфиденциальная информация                          |
| ИБ     | – | информационная безопасность                          |
| ИС     | – | информационная система                               |
| МСЭ    | – | межсетевой экран                                     |
| НИ     | – | несекретная информация                               |
| НСД    | – | несанкционированный доступ                           |
| ПО     | – | программное обеспечение                              |
| СПО    | – | специальное программное обеспечение                  |
| ТСК СН | – | телекоммуникационная система специального назначения |
| ТС     | – | техническое средство                                 |

## ВВЕДЕНИЕ

Информация в настоящее время стала важным стратегическим ресурсом, одним из основных достоинств экономически развитой страны. Быстрое развитие информатизации в России, проникновение ее во все сферы жизненно важных интересов личности, общества и государства вызвали кроме несомненных преимуществ также появление некоторых значимых проблем. Одной из них стала необходимость защиты информации. В настоящее время экономический потенциал все в большей степени определяется уровнем развития информационной структуры, пропорционально растет потенциальная уязвимость экономики от информационных воздействий.

Одной из задач обеспечения информационной безопасности Российской Федерации является необходимость обеспечения защиты информации при внедрении новейших информационных технологий в государственные системы обработки информации. Такие системы объединяются под названием телекоммуникационных систем специального назначения, создаваемых в интересах органов государственной власти и управления [3].

В связи с большими темпами развития вычислительной техники средства обработки информации стали многофункциональными и технически сложными. Поэтому, характеристики надежности этих средств в настоящее время ухудшаются, что приводит к увеличению вероятности появления отказов. В текущих условиях эксплуатации технически сложных средств необходимо при анализе технических средств по требованиям безопасности информации производить не только анализ аппаратного обеспечения, но и анализ программного обеспечения.

При анализе возможных угроз безопасности информации и их источников большое внимание необходимо уделять угрозам природного происхождения, связанным со сбоями и неисправностями технических

средств. Технические сбои и неисправности могут приводить не только к нарушению целостности и доступности информации, но и к нарушению её конфиденциальности. При анализе телекоммуникационной системы специального назначения на соответствие требованиям к безопасности информации в том числе производится анализ программного обеспечения технических средств, участвующих в процессе обмена конфиденциальной информацией.

Программное обеспечение - неотъемлемая часть компьютерной системы. Аппаратные компоненты функционируют под управлением установленного на них программного обеспечения, поэтому функции аппаратных компонентов полностью зависят от программного обеспечения. Сфера применения конкретного технического средства определяется созданным для него программным обеспечением. При сертификации технического средства проводится анализ аппаратного и программного обеспечений технического средства для того, чтобы убедиться в отсутствии недокументированных возможностей АО и ПО и в соответствии АО и ПО реально документированным возможностям технического средства.

Анализ программного обеспечения — процедура исследования, имеющая цели:

- показать разработчикам и заказчикам, что программа соответствует требованиям;
- определить условия, в которых поведение программы является ошибочным, нежелательным или не соответствующим спецификации.

В настоящее время для анализа программного обеспечения на наличие уязвимостей могут использоваться средства статического или динамического анализа. Помимо прочего, одной из важных задач является процесс восстановления алгоритма программного обеспечения. Этот метод позволяет выявить опасные программные конструкции, использование которых угрозами информационной безопасности телекоммуникационной системы

специального назначения может привести к нежелательным событиям – инцидентам информационной безопасности, в частности, к утечке конфиденциальной информации.

При анализе программного обеспечения необходимо восстановить алгоритм работы программы, убедиться в том, что восстановленный алгоритм работы программы соответствует реально документированным возможностям программы, не содержится опасных функциональных конструкций в программном коде, используются методы обработки ошибок, возникающих в процессе функционирования программы.

Существующие методики анализа программного обеспечения технических средств, включающие просмотр программного кода, не предусматривают восстановление алгоритма программ для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения, возникающих в результате сбоев и неисправностей. В связи с этим, необходимо разработать методический подход к процессу восстановления алгоритма программ в рамках анализа телекоммуникационной системы специального назначения на соответствие требованиям к безопасности информации с учетом возможности появления опасных событий в телекоммуникационных системах специального назначения, возникающих в результате сбоев и неисправностей. Данный подход должен предусматривать методы автоматизации процесса восстановления алгоритма программ, так как этот процесс является довольно трудоемким.

В рамках дипломного проекта была разработана методика восстановления алгоритма программ для оценки возможности появления опасных событий в телекоммуникационной системе специального назначения.

## 1 Типовая архитектура построения телекоммуникационной системы специального назначения

Телекоммуникационная система специального назначения (ТКС СН) – комплекс технических средств (ТС), обеспечивающих передачу информации на любые расстояния, предназначенный исключительно для выделенной группы должностных лиц с использованием гарантированных мер обеспечения информационной безопасности.

К ТКС СН предъявляется ряд требований по защите информации, в частности, по обеспечению конфиденциальности принимаемой, обрабатываемой, передаваемой и хранимой информации.

Типовая архитектура построения телекоммуникационной системы специального назначения представлена на рисунке 1.

Рассматриваемая ТКС СН находится в одном здании, в пределах контролируемой зоны. Подсеть №1 обрабатывает информацию ограниченного доступа (конфиденциальную) и общедоступную (несекретную) информацию. Подсеть №2 обрабатывает только общедоступную информацию. Подсеть №3 обрабатывает только конфиденциальную информацию.

Взаимодействие подобных сетей должно осуществляться с использованием межсетевого экрана, требования к которому определяются РД Гостехкомиссии России «Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации» [4].

Телекоммуникационная система специального назначения имеет структуру, описываемую в технической документации на ТКС СН, в состав которой включаются руководство по эксплуатации, электрические схемы, иные документы.

В состав рассматриваемой телекоммуникационной системы специального назначения входят автоматизированные рабочие места (АРМ) операторов, коммутатор Ethernet, маршрутизатор.

В данной ТКС СН обрабатывается информация двух видов: конфиденциальная (КИ) и общедоступная (ОИ).

Уровни конфиденциальности информационных ресурсов, обрабатываемых ТКС СН: конфиденциально, несекретно.

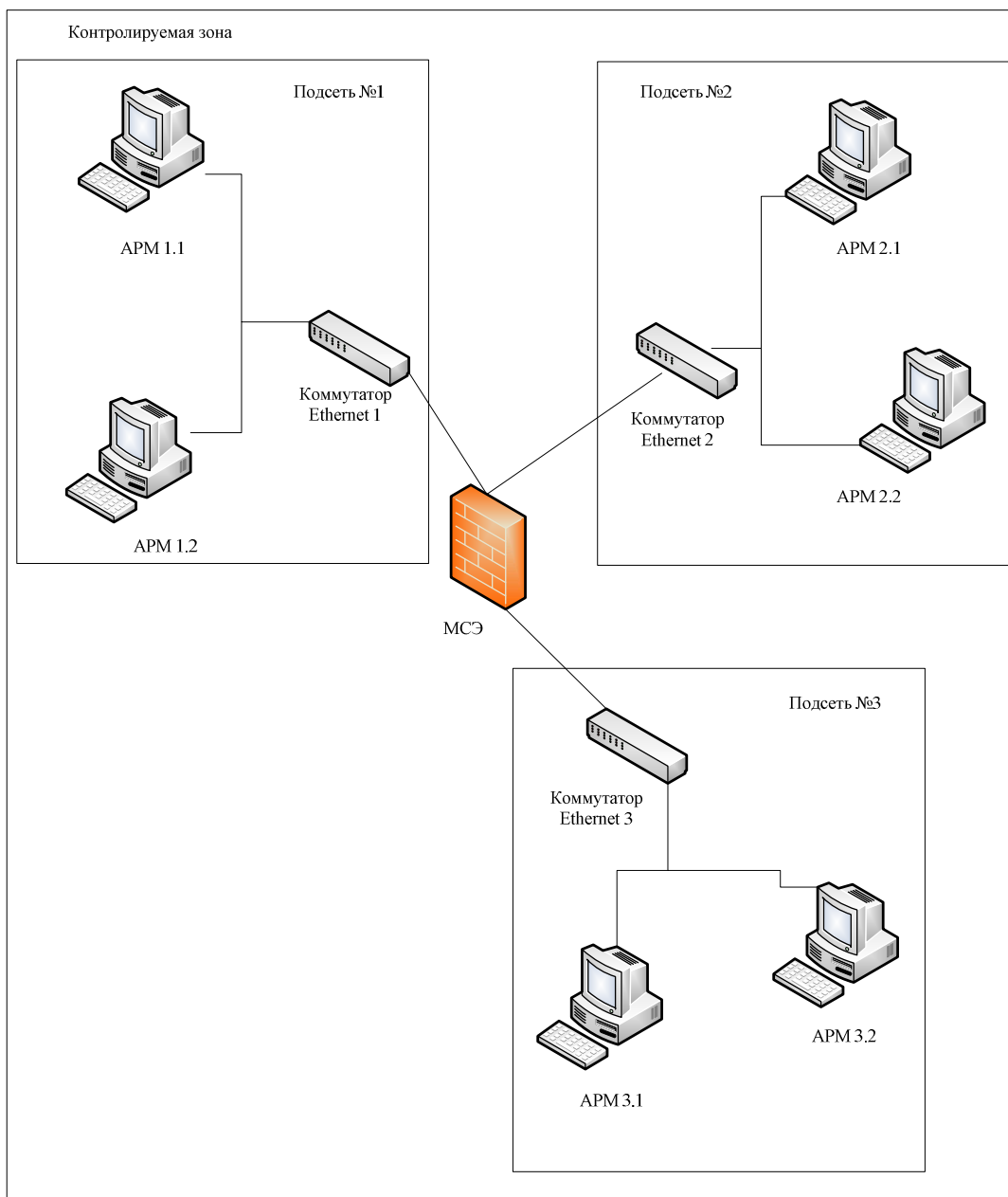


Рисунок 1 – Структура телекоммуникационной системы специального назначения

ТКС СН состоит из трех подсетей, обрабатывающих информацию различных уровней конфиденциальности:

- подсеть №1 обрабатывает и общедоступную, и конфиденциальную информацию;
- подсеть №2 обрабатывает только общедоступную информацию, попадание конфиденциальной информации в данную подсеть запрещено;
- подсеть №3 обрабатывает только конфиденциальную информацию.

Подсеть №1 состоит из двух АРМ (АРМ 1.1 и АРМ 1.2), обрабатывающих общедоступную информацию, предназначенную для обмена с АРМ подсети №2, и конфиденциальную информацию, предназначенную для обмена с АРМ подсети №3.

Подсеть №2 состоит из двух АРМ (2.1 и 2.2), обрабатывающих только общедоступную информацию.

Подсеть №3 состоит из двух АРМ (3.1 и 3.2), обрабатывающих только конфиденциальную информацию.

Взаимодействие между подсетями осуществляется с помощью межсетевого экрана, располагаемого между подсетями. В соответствии со списком правил фильтрации межсетевой экран разрешает или запрещает передачу информации из одной части сети в другую.

Размещение средств ТКС СН осуществляется в пределах контролируемой зоны.

Конфиденциальная информация может передаваться от технических средств АРМ 1.1 и АРМ 1.2 подсети №1 к техническим средствам АРМ 3.1 и АРМ 3.2 подсети №3. Передача конфиденциальной информации техническим средствам подсети №2 запрещена.

Таким образом, в данной телекоммуникационной сети специального назначения организованы следующие тракты прохождения конфиденциальной информации:

- АРМ 1.1 ↔ Eth №1 ↔ МСЭ ↔ Eth №3 ↔ АРМ 3.1;
- АРМ 1.2 ↔ Eth №1 ↔ МСЭ ↔ Eth №3 ↔ АРМ 3.1;
- АРМ 1.1 ↔ Eth №1 ↔ МСЭ ↔ Eth №3 ↔ АРМ 3.2;
- АРМ 1.2 ↔ Eth №1 ↔ МСЭ ↔ Eth №3 ↔ АРМ 3.2.

Общедоступная информация может передаваться от технических средств АРМ 1.1 и АРМ 1.2 подсети №1 техническим средствам АРМ 2.1 и АРМ 2.2 подсети №2 и обратно. Передача общедоступной информации техническим средствам подсети №3 запрещена.

Таким образом, в данной телекоммуникационной сети специального назначения организованы следующие тракты прохождения общедоступной информации:

- АРМ 1.1 ↔ Eth №1 ↔ МСЭ ↔ Eth №2 ↔ АРМ 2.1;
- АРМ 1.2 ↔ Eth №1 ↔ МСЭ ↔ Eth №2 ↔ АРМ 2.2;
- АРМ 1.1 ↔ Eth №1 ↔ МСЭ ↔ Eth №2 ↔ АРМ 2.1;
- АРМ 1.2 ↔ Eth №1 ↔ МСЭ ↔ Eth №2 ↔ АРМ 2.2.

Информационные потоки, возникающие при взаимодействии объектов ТКС СН, приведены на рисунке 2.

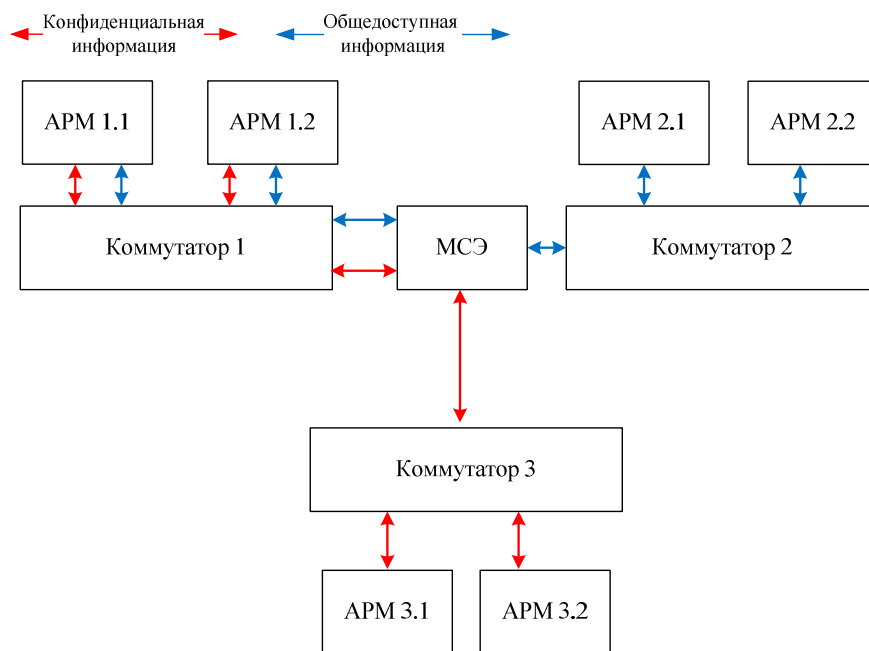


Рисунок 2 – Информационные потоки в ТКС СН

Таким образом, по результатам выполнения первого этапа была описана типовая архитектура построения ТКС СН, входящие в нее технические средства, тракты прохождения информации. Данная информация необходима для последующего выбора технического средства для восстановления алгоритма работы программы.

2 Выбор технического средства и обоснование необходимости восстановления алгоритма работы его программы

### 2.1 Выбор технического средства

В данной ТКС СН циркулирует информация как общедоступная, так и конфиденциальная. Передача конфиденциальной информации на технические средства подсети №2 запрещена. В связи с этим, необходимо определить перечень опасных событий, которые могут возникнуть в процессе обмена конфиденциальной информацией внутри телекоммуникационной системы специального назначения.

К примеру, опасными событиями в процессе обмена информацией внутри рассматриваемой телекоммуникационной системы специального назначения могут являться:

- ошибочная обработка межсетевым экраном конфиденциальной информации, поступившей из подсети №1, и дальнейшая передача ее в тракт обмена информацией между межсетевым экраном и коммутатором 2 подсети №2, что влечет за собой возникновение канала утечки конфиденциальной информации;

- ошибочная обработка межсетевым экраном конфиденциальной информации, поступившей из подсети №3, и дальнейшая передача ее в тракт обмена информацией между межсетевым экраном и коммутатором 2 подсети №2, что влечет за собой возникновение канала утечки конфиденциальной информации.

Таким образом, в данной ТКС СН с учетом особенностей ее построения необходимо обеспечить уверенность в том, что в процессе обмена информацией между подсетями №1 и №3 обеспечивается конфиденциальность передаваемых сведений.

В рассматриваемой телекоммуникационной системе специального назначения появление каналов утечки конфиденциальной информации возможно только в случае нештатной работы аппаратного или программного

обеспечения межсетевого экрана. Отклонения межсетевого экрана от штатного функционирования возможны в результате сбоев и неисправностей, возникающих в процессе его работы.

С целью выявления возможности возникновения подобных событий проводится анализ аппаратного и программного обеспечения технического средства. Так как функционирование аппаратного обеспечения технического средства происходит под управлением программного обеспечения, особое внимание стоит уделить анализу программного обеспечения. Программное обеспечение может быть встроенным, системным или специальным.

В рассматриваемой ТКС СН в межсетевом экране установлено только специальное программное обеспечение, выполняющее основные функциональные задачи, решаемые техническим средством. Необходимо установить, что программное обеспечение межсетевого экрана не имеет в своем составе опасных конструкций, которые могут привести к возникновению указанных выше опасных событий, опасных алгоритмов работы, и алгоритм его программы соответствует реально документированным возможностям. Для этого проводится восстановление алгоритма работы его программы.

Таким образом, техническим средством для восстановления алгоритма его программы выбран межсетевой экран.

Межсетевой экран – комплексное программное или аппаратное обеспечение, которое позволяет на необходимом уровне безопасности контролировать количество и качество сетевых пакетов, проходящих через него. Межсетевой экран осуществляет анализ сетевого трафика, следуя определенному набору правил, в соответствии с которыми выполняется фильтрация всех данных.

Основная задача межсетевого экрана – защита локальных узлов или общих компьютерных сетей от несанкционированного доступа, который может использовать данные в своих целях либо нанести вред владельцу сети.

Фильтрация сетевого трафика может производиться на любом из уровней модели OSI. В качестве критериев может быть использована информация с разных уровней: номера портов, содержимое поля данных, адрес отправителя/получателя.

## 2.2 Обоснование необходимости восстановления алгоритма работы программы технического средства

### 2.2.1 Проблема защиты программ

Сегодняшняя нормативная, методическая и инструментальная база выявления недеklarированных возможностей программ не позволяет эффективно обеспечивать безопасность программных ресурсов. Данной проблемой занимаются испытательные лаборатории Минобороны и ФСТЭК России, однако большинство уязвимостей обнаруживаются не в соответствии с нормативными документами. Имеется тенденция увеличения числа уязвимостей в программном коде.

Защитой программ занимаются лаборатории Минобороны и ФСТЭК России. В своей работе они опираются на Руководящий документ Гостехкомиссии России по контролю над недеklarированными возможностями. Однако аналитики имеют большой опыт выявления закладок и ошибок кода, что происходит не в соответствии с этим документом, а скорее вопреки ему. Существующая нормативная, методическая и инструментальная база выявления недеklarированных возможностей в программном обеспечении не позволяет эффективно обеспечивать безопасность программ.

Причины появления уязвимостей в программном обеспечении заключаются в высокой структурной сложности кода, быстрому развитию версий программ и легкости модификации кода. Можно добавить и проблему достоверного определения программных закладок, созданных преднамеренно, несовершенство нормативной и методической баз, отставание инструментальной базы сертификационных испытаний. В отличие

от средств антивирусного контроля, в настоящее время отсутствуют средства гарантированного выявления программных закладок в структурно сложном программном обеспечении. На сегодняшний момент так и не обеспечены разработки математического аппарата для оценки степени безопасности программного обеспечения, основанного на сертификационных испытаниях с целью подтвердить отсутствие программных закладок. В связи с этим происходит снижение качества разработки программных средств в отечественных государственных учреждениях, как и уровня доверия к зарубежным продуктам.

Основой теории информационной безопасности являются понятия угрозы (наличие уязвимости) и риска (возможность реализации угрозы). Под уязвимостью программного кода понимается дефект программного обеспечения, который может потенциально снизить степень его безопасности. Значимые классификационные признаки уязвимостей кода: преднамеренность внесения уязвимости кода (логическая ошибка кодирования, программная закладка), уровень выявленных уязвимостей и уровень функционирования (в исходном и выполняемом коде, в среде), подсистема безопасности, компрометируемая уязвимостью (парольная, криптографическая и т.д.).

#### 2.2.2 Ограничения Руководящего документа

Руководящий документ по недекларированным возможностям был создан в 90-х годах, чтобы решить проблему контроля в России иностранных продуктов, и, без сомнения, имеет большое значение. Методы, которые определены в РД, берут свое начало в теории надежности работы программ, так что вопросы защиты кода, разобранные в документе, отражены не достаточно четко. В соответствии с РД одними из основных видов проверок, которые должны быть выполнены в испытательной лаборатории, являются структурированный статический и динамический анализ исходных кодов.

Достоинствами документа является то, что документ содержит требования предоставлять исходный код и документацию для осуществления контроля избыточности данных (что позволяет исключить определенные элементы программных закладок), и определения полномаршрутного тестирования. Однако существуют и серьезные недостатки в РД:

- значительная сложность вычислений статического и динамического анализа. Так, с увеличением сложности программ динамический анализ становится неразрешимой задачей и превращается в формальную процедуру, занимающую много времени и ресурсов у экспертов;

- отсутствие или недостаточность проверок, непосредственно связанных с безопасностью изделия. Например, не указывается сигнатурный анализ в требованиях к испытаниям программ ниже второго уровня контроля (т.е. программ, которые обрабатывают секретную и конфиденциальную информацию). Не выдвигается требований к содержимому базы потенциально опасных конструкций, что делает неэффективными методы сигнатурного анализа. Нет механизмов нахождения ошибок кодирования, которые связаны с переполнением буфера, гонками, вызовом функций из чужого адресного пространства. Отсутствует очистка памяти и т.д.;

- неопределенность действий экспертов и достоверности получаемых результатов. Отсутствуют способы, нацеленные на построение перечня маршрутов при выполнении функциональных объектов (ветвей). Нет механизмов определения полноты покрытия и его достаточности при динамическом анализе. Отсутствуют рекомендации по нахождению недекларированных возможностей при анализе, например, матрицы связей по информации.

Самыми небезопасными остаются продукты, прошедшие сертификацию по третьему или четвертому уровню контроля. Четвертый уровень сводится к проверке возможности компиляции и сборки

программного продукта, а также отсутствия избыточных файлов в его дистрибутиве. Таким образом, возникает лишь иллюзия проверки программы в соответствии с требованиями безопасности. Для третьего уровня подход выявления недеklarированных возможностей представляет собой слишком длительное изучение структуры программы, которое не включает в себя анализ функций, процедур и методов на признаки программных закладок и некорректностей кодирования кода.

### 2.2.3 Инструментальная база испытаний

Чтобы получить аттестат аккредитации ФСТЭК по выявлению недеklarированных возможностей, испытательная лаборатория должна иметь сертифицированные инструментальные средства («АИСТ», ЕМУ).

Возникает вопрос, целесообразно ли делать обязательными к применению инструментальные средства, основанные на несовершенной нормативной базе и не позволяющие эффективно выявлять уязвимости кода? Исследования показывают, что даже в тестовой программе, содержащей несколько простейших закладок, ни одна из них не была выявлена с помощью указанных средств.

Для анализа кода на языке С рекомендуется программа «АИСТ-С». Однако, несмотря на ее очевидные достоинства, установлено, что она некорректно обрабатывает структуры объектно-ориентированных программ, например, не анализирует конструктор и деструктор. Не анализируются и исходные тексты, подключаемые директивами `#include` — лишь проверяется исходный текст программы. Мало того, нормальная работа анализатора возможна, только если объем исходных текстов не превышает нескольких мегабайт: блок-схемы программ строятся некорректно уже для программ объемом более нескольких килобайт. Те же ограничения относятся к построению матрицы связей по информации, да и отчеты не в полной мере соответствуют требованиям РД по недеklarированным возможностям. Но главное отставание от реалий дня состоит в том, что это средство вообще не

имеет базы сигнатур и умеет анализировать лишь программы на языке C/C++.

### 2.3 Методы поиска опасных конструкций в программном обеспечении

Хорошо известно, что любая информационная атака реализуется злоумышленником на основе той или иной уязвимости, которая присутствует в атакуемой системе. Согласно ГОСТ Р 50922-2006 под уязвимостью понимается свойство информационной системы, обуславливающее возможность реализации угроз безопасности обрабатываемой в ней информации [3].

Для выявления уязвимостей кода необходим комплексный подход. Используя критерии вероятности реализации уязвимости и ее последствий, можно выработать рекомендации по уровням контроля над уязвимостями кода для применения программного обеспечения в системах конкретного класса.

Аудит программного кода по требованиям безопасности представляет собой структурное тестирование ПО с целью выявления уязвимостей, реализация которых может снизить уровень целостности, доступности и конфиденциальности системы. Условием проведения аудита безопасности кода является наличие исходных текстов программ и их спецификаций.

Важной особенностью технологий security code reviews является то, что основная задача аудита - выявление не всех возможных уязвимостей, а только уязвимостей кода, которые могут быть использованы злоумышленником.

В общем плане аудит безопасности кода является итерационным процессом, включающим мероприятия по планированию, проведению анализа, выработке рекомендаций по доработке программы и документации, а также развитию методик и средств выявления и анализа уязвимостей.

Уязвимости программного кода могут являться некорректностями кодирования или ошибками проектирования, а также иметь злоумышленный

или непреднамеренный характер. Однако, согласно открытым публикациям, основные приемы security code reviews, используемые при проверках кода, ориентированы на выявление некорректностей кодирования подсистем безопасности. Основные классы указанных уязвимостей:

- переполнение, чтение и запись вне буфера;
- выход вычислений за пределы определенного диапазона при преобразовании переменных числового типа;
- формирование отрицательного значения длины строк байт или количества элементов массива;
- некорректное приведение типов;
- отсутствие инициализации данных;
- утечка, нехватка, использование освобожденной памяти;
- ошибки определения времени и синхронизации;
- ошибки блокировок в многопоточных средах и др.

Данные классы уязвимостей могут быть использованы при проведении атак на отказ в обслуживании или выполнении нелегитимного кода.

Очевидно, что выявление некорректностей кодирования полностью не решает проблему обеспечения безопасности кода. К важному классу уязвимостей относят ошибки проектирования подсистем защиты и преднамеренные (логические) программные закладки. Наиболее известными из них являются: наличие недекларированных отладочных и деструктивных функций, неверные реализации протоколов или алгоритмов шифрования, использование собственных механизмов псевдобезопасности, наличие встроенных мастер-паролей.

С каждым годом увеличивается количество уязвимостей, которые обнаруживаются в прикладном программном обеспечении. При этом растет не только их количество, но и их разновидность.

Выделяют несколько методов аудита безопасности кода:

- просмотр (инспекция) кода вручную;
- статический анализ кода по шаблону;
- динамический анализ выполнения кода.

Поиск уязвимостей в ПО может осуществляться в ручном режиме либо с использованием средств автоматизации. В настоящее время для выявления уязвимостей ПО могут использоваться средства статического или динамического анализа.

Статический анализ предполагает поиск потенциально опасных конструкций в исходном коде программного обеспечения. Данный вид анализа потенциально позволяет выявлять технологические уязвимости, которые возникают на этапе разработки ПО вследствие халатности или злого умысла разработчика.

Процесс внедрения средств статического анализа требует вовлечения не только подразделения по защите информации, но и подразделения, которое отвечает за разработку программного обеспечения. Это обусловлено тем, что только разработчики смогут провести анализ результатов сканирования исходного кода и внести в исходный код приложения необходимые корректировки.

Средство для статического анализа запускает процесс сканирования исходного кода в момент централизованной сборки всего приложения. Как правило, этот процесс осуществляется в конце рабочего дня. Информация о выявленных в процессе сканирования уязвимостях поступает в сервер управления, к которому имеет доступ представитель отдела разработки, ответственный за обеспечение информационной безопасности ПО. Проанализировав результаты сканирования, он выделяет актуальные уязвимости и создает по ним соответствующие заявки в системе отслеживания ошибок (Bug Tracker). На следующем этапе руководитель всего процесса разработки ПО назначает ответственных разработчиков за устранение выявленных уязвимостей на основе созданных заявок. После

этого эти разработчики вносят изменения в исходный код с целью устранения выявленных уязвимостей. Далее этот процесс повторяется на регулярной основе до завершения процесса разработки приложения. При этом с заданной периодичностью представитель подразделения по защите информации осуществляет мониторинг того, что все уязвимости, выявляемые системой статического анализа, успешно устраняются командой разработчиков.

Динамический анализ предполагает проведение тестирования уже скомпилированного ПО и функционирующего в определенной среде. Для решения данной задачи, в частности, могут использоваться специализированные сканеры безопасности, которые моделируют возможные атаки злоумышленников, а также анализируют конфигурационные файлы ПО. Динамический анализ позволяет выявлять не только технологические, но и эксплуатационные уязвимости, связанные с неправильной настройкой ПО. К таким уязвимостям можно отнести: использование слабых и нестойких к угадыванию паролей доступа, наличие неиспользуемых учетных записей, неправильно настроенные функции безопасности ПО и др. В качестве примеров продуктов, реализующих функционал динамического анализа, можно привести решения Max Patrol (компании Positive Technologies) и Qualys (компании Qualys, Inc.).

Еще одной разновидностью динамического анализа является фаззинг. Фаззинг представляет собой процесс передачи на вход программе намеренно некорректных данных с целью вызова ситуации сбоя или ошибки. Таким образом, эта технология позволяет обнаруживать уязвимости в анализируемом приложении.

Использование средств динамического анализа не требует наличия исходного кода ПО, а также вовлечения разработчиков для проведения анализа безопасности приложения.

На сегодняшний день более 80% атак злоумышленников базируются на использовании уязвимостей в прикладном ПО. Своевременное выявление и устранение этих уязвимостей позволит предотвратить возможные атаки нарушителей. Для эффективного решения данной задачи рекомендуется использовать современные инструментальные средства анализа безопасности ПО, которые могут значительно автоматизировать процесс поиска и устранения слабых мест в ПО организации.

Одними из причин появления возможности утечки конфиденциальной информации из-за использования уязвимостей в программном обеспечении являются аппаратные и программные сбои.

Аппаратные сбои – выход из строя или сбои в работе комплектующих технического средства. Со временем механические и электрические части технического средства изнашиваются и приходят в негодность. Причиной аппаратных сбоев могут быть перегрев, короткое замыкание, переизбыток влаги, работа рядом с техническим средством источников тепла и электромагнитного излучения. Так как в настоящее время функционирование аппаратных компонентов технических средств производится под управлением установленного на них программного обеспечения, фактически выполняющего задачи технического средства.

Программные сбои – ошибки запуска или некорректная работа программного обеспечения. Программные сбои могут возникать по следующим причинам:

- ошибки в коде программ. Не всегда разработчики могут предусмотреть все возможные сценарии использования программ или просто допускают ошибки. Также возможна нестабильная работа программ, которые подверглись изменению третьими лицами, взломанные или подверженные заражению вирусом;

- программная несовместимость. Операционные системы меняются и выходят новые версии программ, совместимые с новыми ОС, но

не всегда разработчики занимаются поддержкой и развитием своего ПО. Любая запускаемая программа требует определенной среды для своей стабильной работы, поэтому у них есть свои системные требования, которые прописываются разработчиком – определенная операционная система, её разрядность и наличие в ней дополнительных компонентов, если они требуются;

- конфликт в работе с другими программами. Большинство устанавливаемых программ вносят изменения на компьютере, делают запись в реестре, ставят себя в автозагрузку или регистрируют службу обновления. Так или иначе операционная система изменяется, что может послужить причиной дальнейших конфликтов или нестабильной работы системы в целом. Самый распространенный случай – работа программы блокируется антивирусом.

Метод просмотра кода вручную считается самым эффективным с точки зрения полноты и точности проверок. Только эксперт способен обнаружить сложные уязвимости и маскируемые программные закладки, наличие опасных функциональных конструкций программного кода, наличие опасных алгоритмов программы при функционировании внутри ТКС СН, а также дать рекомендации по исправлению уязвимостей или ограничению возможности их реализации. Данный метод позволяет также оценить соответствие восстановленного алгоритма программы реально документированным возможностям, определить наличие функциональных конструкций программного кода, позволяющих обеспечить обработку ошибок, возникающих в процессе функционирования программы.

К недостаткам метода относят высокую трудоемкость и требования к квалификации и опыту экспертов. Для исключения субъективизма к работе экспертов может быть привлечены независимые группы тестирования, что является еще более затратным.

Существующие методики восстановления алгоритма программ не предполагают анализ опасных алгоритмов программ при функционировании технического средства внутри ТКС СН, и анализ алгоритмов программ, реализующих реакции на ошибки, возникающие в результате сбоев и неисправностей аппаратного обеспечения технических средств из состава телекоммуникационных систем специального назначения.

Для усовершенствования анализа программы технического средства из состава ТКС СН, рассмотренной в разделе 1 настоящего дипломного проекта, необходимо применить метод ручной инспекции кода с документированием его участков. Для обеспечения более эффективной навигации по коду, а также обеспечения эффективной работы аналитиков в команде в процессе документирования программного кода используются различные программные средства, описание которых представлено в разделе 3.

Таким образом, по результатам выполнения второго этапа выбрано техническое средство из состава ТКС СН для анализа его программы, приведено обоснование необходимости восстановления алгоритма его программы для оценки возможности появления опасных событий в рассматриваемой ТКС СН.

### 3 Описание используемых при восстановлении алгоритма работы программных средств

#### 3.1 Среда разработки Eclipse

Eclipse — свободная интегрированная среда разработки модульных кроссплатформенных приложений.

Платформа Eclipse создана для построения интегрированных сред разработки (IDE). Она может использоваться для создания различных сквозных вычислительных решений для множества сред выполнения программ.

Платформа Eclipse представляет собой фундамент для построения и запуска интегрированных инструментов разработки сквозного программного обеспечения. Платформа состоит из компонентов с открытым исходным кодом, используемым поставщиками инструментов для построения решений, встраиваемых в интегрированную рабочую область. Платформа Eclipse объединяет в себе технологию, выражающуюся в правильно оформленном дизайне и среде реализации.

Платформа Eclipse была построена на следующих принципах:

- содействие в однородной интеграции внутренних инструментов, а также между различными типами контента и поставщиками инструментов;
- поддержка создания разнообразных инструментов;
- поддержка неограниченного множества поставщиков инструментов, включая независимых поставщиков ПО;
- поддержка инструментов для работы с любыми форматами (включая html, java, c, jsp, ejb, xml и gif);
- поддержка сред разработки приложений с графическим интерфейсом (GUI) и без него;
- запуск на широком диапазоне операционных систем, включая Windows и Linux;

- использование популярности языка java для написания инструментов.

Одна из интересных особенностей Eclipse – скорость, с которой обнаруживаются ошибки, и с которой разработчики могут реагировать на нужды и пожелания пользователей. Eclipse Consortium каждый месяц выпускают новые релизы, которые включают в себя стабильные версии различных возможностей, которыми интересуются пользователи. Среди возможностей, включенных в релиз 2.1 M3, можно выделить следующие, наиболее значимые:

- настраиваемые связи "горячих" комбинаций клавиш, включая набор связей, схожий с тем, что есть в Emacs;
- возможность автоматической вставки TODO, FIXME и других тэговых комментариев в Java-код, в различные элементы и списки задач;
- улучшенная поддержка CVS, включая возможность определять "working sets", ограничивающих количество отображаемых проектов на панели CVS Repositories;
- новые заготовки в Java-редакторе, которые могут использоваться для показа исходного кода, Javadoc, подробных сообщений об ошибках для каждой из строк кода или выражения в "выпадающем" (tooltip) окне;
- поддержка некоторых новых рефакторингов;
- новая панель Threads and Monitors в отладчике, которая позволяет определить, какие из потоков блокируют ресурсы, какие из них находятся в состоянии ожидания, и иную полезную для пользователя информацию при отладке мультитепотоковых приложений;
- более тесная интеграция с инструментом сборки Jakarta Ant.

Последняя особенность делает Eclipse хорошо скомпонованной средой разработки Java-приложений и, в частности, разработки Web-приложений. Eclipse объединяет в себе все инструменты, необходимые разработчику для успешной разработки Web-приложений: расширяемую IDE, стандартный

компилятор, возможность удаленной отладки, поддержка Ant, тестирование с помощью JUnit, plug-in'ы для взаимодействия, практически, со всеми серверами приложений, а также EJB-контейнер.

Для осуществления наиболее эффективного использования среды разработки Eclipse при осуществлении анализа программного кода подключены плагины следующих средств: средства контроля версий Git, системы документирования программного кода Doxygen, пакета утилит по автоматической визуализации графов Graphviz.

### 3.2 Плагин средства контроля версий Git

Git — мощная и сложная распределенная система контроля версий.

Git поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки. Git предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой.

Удалённый доступ к репозиториям Git обеспечивается git-daemon, SSH- или HTTP-сервером. TCP-сервис git-daemon входит в дистрибутив Git и является наряду с SSH наиболее распространённым и надёжным методом доступа. Метод доступа по HTTP, несмотря на ряд ограничений, очень популярен в контролируемых сетях, потому что позволяет использовать существующие конфигурации сетевых фильтров.

Главное отличие Git'a от любых других СКВ (например, Subversion и ей подобных) — это то, как Git смотрит на свои данные. В принципе, большинство других систем хранит информацию как список изменений (патчей) для файлов.

Git считает хранимые данные набором слепков файловой системы. Каждый раз, когда пользователь фиксирует текущую версию проекта, Git сохраняет слепок того, как выглядят все файлы проекта на данный момент. Ради эффективности, если файл не изменялся, Git не сохраняет файл снова, а

делает ссылку на ранее сохраненный файл. То, как Git подходит к хранению данных, представлено на рисунке 3.

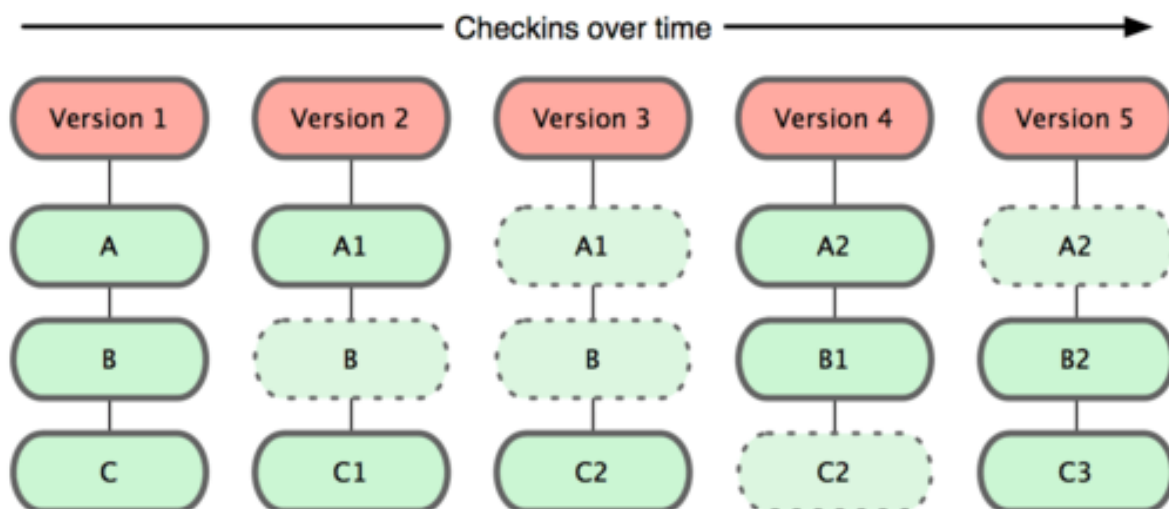


Рисунок 3 – Принцип хранения состояний проекта Git во времени

Это важное отличие Git'a от практически всех других систем контроля версий. Git похож на небольшую файловую систему с невероятно мощными инструментами, работающими поверх нее.

Для осуществления большинства операций в Git'e необходимы только локальные файлы и ресурсы, т.е. обычно информация с других компьютеров из сети не нужна. Так как вся история проекта хранится локально на диске, большинство операций оказываются практически мгновенными.

Чтобы показать всю историю для проекта, Git'у не нужно скачивать ее с сервера, он может читать ее прямо из локального репозитория.

Перед сохранением каждого файла Git вычисляет его контрольную сумму, и она становится меткой этого файла. Невозможно изменить содержимое файла или каталога таким образом, чтобы Git не узнал об этом. Эта функциональность является встроенной в фундамент Git'a и является важной составляющей его устройства. Если информация потеряется при передаче или повредится на диске, Git всегда это выявит.

Механизм, используемый Git'ом для вычисления контрольных сумм, называется SHA-1 хешем. Это строка из 40 шестнадцатеричных символов (0-

9 и a-f), вычисляемая в Git'e на основе содержимого файла или структуры каталога. Фактически, в своей базе данных Git сохраняет всё не по именам файлов, а по хешам их содержимого.

Для наиболее эффективного процесса восстановления алгоритма программного обеспечения необходима командная работа нескольких пользователей. Git позволяет осуществить подобное путем синхронизации локальных репозиториях пользователей. Это позволяет сразу нескольким пользователям выполнять анализ программного кода, не конфликтуя в процессе его описания.

### 3.3 Плагин системы документирования программного кода Doxygen

Doxygen генерирует документацию на основе набора исходных текстов и также может быть настроен для извлечения структуры программы из недокументированных исходных кодов. Возможно составление графов зависимостей программных объектов, диаграмм классов и исходных кодов с гиперссылками.

Doxygen имеет встроенную поддержку генерации документации в формате HTML, LATEX, man, RTF и XML. Также вывод может быть легко сконвертирован в CHM, PostScript, PDF.

Для html-представления документации, размещаемого на web-серверах, существует удобный способ организации поиска (при помощи создаваемого Doxygen'ом PHP-модуля) и ссылок на внешнюю документацию.

Документация кода в Doxygen осуществляется при помощи документирующего блока. При этом существует два подхода к его размещению:

- он может быть размещён перед или после объявления или определения класса, члена класса, функции, пространства имён и т.д.;
- его можно располагать в произвольном месте (и даже другом файле), но для этого потребуется явно указать в нём, к какому элементу кода

он относится. Этот подход не будет рассмотрен, поскольку даже разработчики рекомендуют его избегать.

Структурно, любой документирующий блок является комментарием, просто оформленным специальным образом, поэтому естественно, что его вид зависит от используемого языка.

Всего существует два основных типа документирующих блоков: многострочный блок и однострочный блок. Текст, написанный в однострочном блоке, относится к краткому описанию документируемого элемента, а текст, написанный в многострочном блоке, относится к подробному описанию.

В начале файла следует добавлять документирующий блок, который содержит описание назначения файла. Для того, чтобы указать предназначения блока файлу, стоит воспользоваться командой `@file`. В качестве параметра к данной команде можно указать путь к файлу. По умолчанию блок будет принадлежать тому файлу, в который он вписан.

Во время документирования функций и методов необходимо включать в документацию также описание входных параметров, возвращаемых значений и возможных исключений.

Документирование классов может осуществляться предварением их документирующим блоком. Задача документирования классов упрощается тем, что большое количество информации Doxygen получает, используя синтаксис языка. Он автоматически определяет методы класса, уровни доступа к функциям, дружественные функции и так далее.

Одним из действенных инструментов структуризации документации является создание функциональных групп, так как это один из наиболее удобных способов навигации по ней.

Для объявления функциональной группы следует использовать команду `@defgroup`, которую необходимо заключить в документирующий блок.

Для модуля идентификатором является какое-либо уникальное слово, которое написано на латинице и будет затем использовано как указатель при обращении к модулю. Заголовком модуля может быть любое слово либо предложение, которое будет отображено в сгенерированной документации.

При описании модуля можно дополнить его кратким и подробным описанием, что позволяет раскрыть назначение того или иного модуля.

Пользователю предоставляется возможность построения графов и диаграмм на основе исходного кода. Графы предназначены продемонстрировать, как взаимодействуют те или иные элементы кода. Графы могут являться и инструментом навигации по коду, так как над элементами графа проставляются ссылки на разделы документации.

Ради упрощения манипуляций с файлом конфигурации (а он содержит довольно много настроек), существует несколько утилит с графическим интерфейсом. Одна из них, `doxuwizard`, поставляется вместе с `Doxugen`.

### 3.4 Плагин пакета утилит по автоматической визуализации графов Graphviz

Graphviz (сокращение от англ. Graph Visualization Software) — пакет утилит автоматической визуализации графов, заданных в виде описания на языке DOT, а также других TUI и GUI программ, виджетов и библиотек, используемых при разработке программного обеспечения для визуализации структурированных данных.

Graphviz состоит из нескольких утилит командной строки, а также утилит с графическим интерфейсом, которые могут обрабатывать файлы на языке DOT. Помимо них Graphviz включает в себя также виджеты и библиотеки, облегчающие создание графов и программ для построения графов, включая следующие:

- `dot` — инструмент для формирования многоуровневого графа с возможностью вывода изображения результирующего графа во множестве форматов (PNG, PDF, PostScript, SVG и т. д.);

- neato — инструмент для формирования графа на основе «пружинной» («spring model», «energy minimised») модели;
- twopi — инструмент для формирования графа на основе «радиальной» модели;
- circo — инструмент для формирования графа на основе «круговой» модели;
- fdp — инструмент для формирования ненаправленного графа на основе «fdp» модели;
- dotty — GUI для формирования графов;
- lefty — программируемый GUI-виджет (на языке EZ).

Пример построения графа с использованием данного программного средства представлен на рисунке 4.

В пакет утилит входит программа «dot», являющаяся автоматическим визуализатором ориентированных графов, принимающая на вход текстовый файл на языке DOT с представлением графа в виде смежных списков, а на выходе формирующая граф в виде графического, векторного или текстового файла.

Входной файл для программы «dot» является обычным текстовым файлом на специальном языке описания.

Программа «dot» сама распознаёт все связи графа и упорядочивает его таким образом, чтобы было наименьшее количество пересечений.

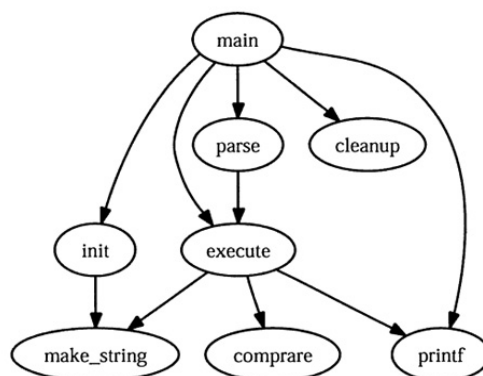


Рисунок 4 – Пример визуализации графа GraphViz

Таким образом, на третьем этапе рассмотрены программные средства, используемые в процессе восстановления алгоритма программы технического средства для оценки возможности появления опасных событий в ТКС СН.

## 4 Описание процесса настройки и использования IDE Eclipse в процессе восстановления алгоритма программы

### 4.1 Установка и настройка IDE Eclipse

При восстановлении алгоритма работы программы в рамках СИА используется интегрированная среда разработки Eclipse CDT.

Приложение eclipse.exe распаковывает в выбираемую директорию portable-версию IDE Eclipse, а также каталог «doxyssettings», в котором находятся файлы, необходимые для корректного функционирования плагина Doxygen в среде разработки IDE Eclipse. Папку «doxyssettings» необходимо распаковать в основную папку с проектом ПО перед началом работы с ним.

Приложение eclipse.exe также устанавливает следующее программное обеспечение:

- Doxygen – средство автоматического документирования исходного кода (версия 1.8.10);
- Graphviz – средство построения графов (версия 2.28);
- Git – средство контроля версий (версия 2.7.4).

После запуска приложения eclipse.exe требуется выбрать путь распаковки архива с portable-версией среды разработки IDE Eclipse (версия Mars2) и для каталога «doxyssettings».

Папка «doxyssettings» содержит следующие файлы и папки:

- Doxyfile – конфигурационный файл программы Doxygen;
- rtfstyle.cfg – конфигурационный файл программы Doxygen;
- файл шаблона стилей ЕСПД;
- StyleChng.bas – макрос, для редактирования сгенерированной документации, который необходимо импортировать в Microsoft Word.

### 4.2 Работа с проектами в IDE Eclipse

IDE Eclipse поддерживает концепцию workspace (рабочее пространство). Workspace является контейнером проектов, который имеет определенный набор установок (Preferences) и метаданных. Можно

использовать как единое рабочее пространство для всех проектов, так и несколько рабочих пространств – например, для нескольких однотипных проектов программного обеспечения – одно рабочее пространство.

При первом запуске Eclipse необходимо выбрать путь к папке с рабочим пространством. Важно, чтобы путь к папке с рабочим пространством не содержал названий директорий с использованием кириллицы, а также пробелов.

Для того чтобы начать работу с новым проектом, необходимо создать новый проект с использованием комбинации команд File->New ->C/C++ project->, ввести название проекта (название проекта так же не должно содержать кириллицы и пробелов), выполнить Next->Next->Finish. После этого в рабочем пространстве создается папка с названием проекта, содержащая конфигурационные файлы Eclipse. В эту папку необходимо скопировать исходные коды проекта ПО, а также каталог doxysettings.

#### 4.3 Конфигурация плагина Doxygen

При первом запуске Eclipse указать путь к исполняемому файлу Doxygen в окне настроек Window->Preferences->Doxygen->Add. В появившемся окне выбрать актуальную версию Doxygen.

В окне Windows->Preference-> C/C++-> Editor в поле Workspace default выбрать «Doxygen».

Среда Eclipse предоставляет интерфейс для редактирования файла конфигурации Doxygen. В папке «Добавить в проект» уже содержится файл конфигурации, содержащий основные оптимальные настройки для проекта на языке C, однако в зависимости от особенностей модулей исходного кода, некоторые настройки рекомендуется изменить:

- Project Name – имя проекта, с которым проводятся работы;
- Optimize\_output\_for\_C – устанавливается в случае, если работы проводятся с исходным кодом, написанном на языке Си;

- Input\_Encoding – кодировка модулей исходного кода (например – UTF-8);
- Dot\_Path – путь к исполняемому файлу GraphViz;
- Input – путь к модулям проекта, необходимо выбрать путь проекта.

При необходимости, можно изменить и другие параметры. Описание всех параметров содержится в файле конфигурации.

#### 4.4 Общие сведения о Doxygen

Doxygen генерирует программную документацию на основе комментариев исходного кода специального формата.

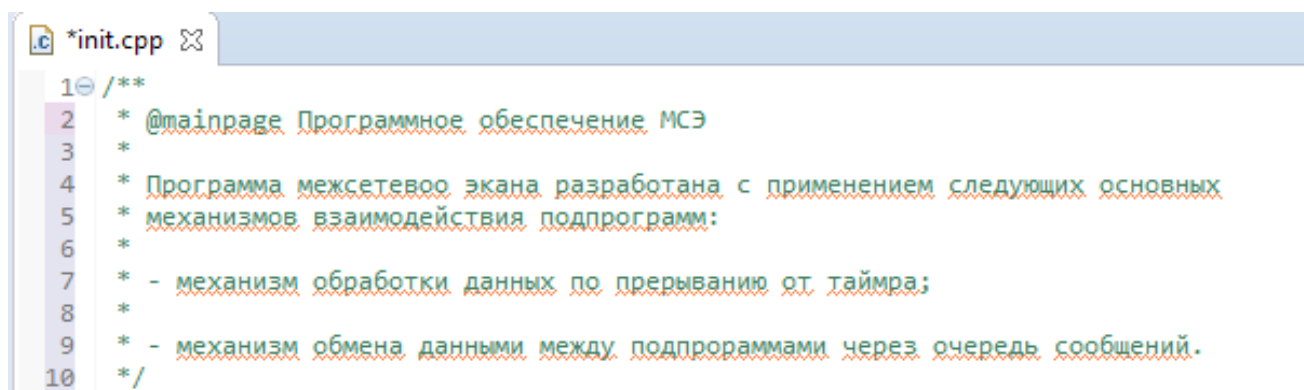
Принцип документирования состоит в следующем: на вход генератора подается код, включающий комментарии специального вида, а на выходе генерируется оформленная документация. В результате сборки документации создается документ формата .rtf, экспортируемый в раздел «Восстановление алгоритма работы» отчета СИА.

Помимо специального комментария «`/**`» могут так же использоваться «`/*!`», для оформления блоков комментариев и «`///», «///» для оформления строк комментариев.`

Теги комментариев Doxygen начинаются с ключевых символов «`@`» или «`\`», после которых идет идентификатор тега. Список всех тегов, используемых при документировании кода в рамках СИА, приведен ниже.

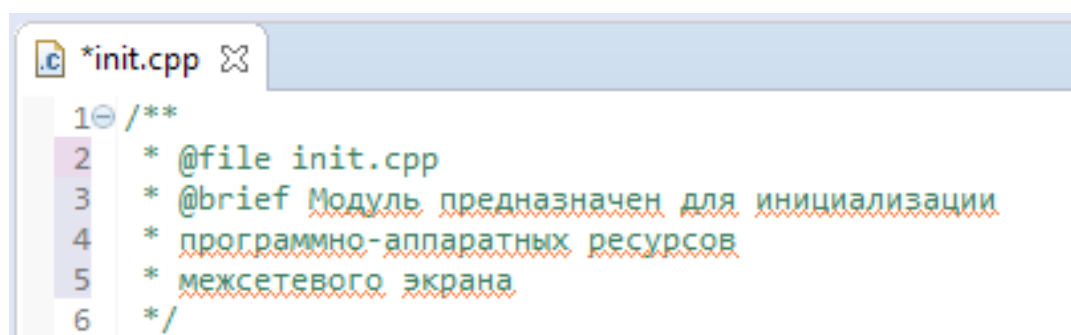
- @mainpage – краткое описание программы, её назначение. Тег является опциональным; информация, помеченная данным тегом, не включается в отчет. Пример использования тега показан на рисунке 5;
- @file – описание программного модуля. Используется для создания таблицы из подраздела анализа ПО «Состав СПО изделия»;
- @brief – краткое описание сущности (функции, структуры, класса, файла и др.). Данный тег используется в двух случаях: при описании

файла исходного кода, а также при описании функций, при восстановлении алгоритма работы ПО. Пример использования тега показан на рисунке 6;



```
1 /**
2  * @mainpage Программное обеспечение МСЭ
3  *
4  * Программа межсетевого экрана разработана с применением следующих основных
5  * механизмов взаимодействия подпрограмм:
6  *
7  * - механизм обработки данных по прерыванию от таймера;
8  *
9  * - механизм обмена данными между подпрограммами через очередь сообщений.
10 */
```

Рисунок 5 – Пример использования тега @mainpage



```
1 /**
2  * @file init.cpp
3  * @brief Модуль предназначен для инициализации
4  * программно-аппаратных ресурсов
5  * межсетевого экрана
6  */
```

Рисунок 6 – Пример использования тегов @file и @brief

– @defgroup id Название функциональной задачи – определение новой группы функций, объединенных идентификатором id. Используется для создания идентификатора восстанавливаемой ветки алгоритма программы. В процессе описания ПО в рамках СИА, необходимо выделить все функциональные задачи, которое решает ПО и определить соответствующие им группы в файле, содержащем точку входа в программу. Пример использования тега @defgroup показан на рисунке 7;

– @ingroup id – добавление функции в группу id. В рамках СИА необходимо для каждой определенной функциональной задачи отметить данным тегом все функции, реализующие эту задачу. Пример использования тега показан на рисунке 8;

– @addtogroup id @{ ... @} – добавление функций, содержащихся между скобками @{ ... @} к созданной группе id. Тег является аналогом @ingroup с возможностью добавления в группу нескольких функций;

```
1  /**
2   * @file init.cpp
3   * @brief Модуль содержит главную функцию программы
4   *
5   * @defgroup init Инициализация программно-аппаратных
6   * ресурсов
7   * @brief В данной функциональной задаче выполняется
8   * инициализация программно-аппаратных ресурсов
9   * межсетевого экрана.
10  *
11  * @defgroup receive Прием данных от АРМ
12  * @brief В данной функциональной задаче выполняется
13  * прием сообщений от АРМ.
14  */
```

Рисунок 7 – Пример использования тега @defgroup

```
32 /**
33  * @ingroup init
34  * @brief Функция предназначена для инициализации
35  * таймера ожидания сообщения от АРМ 1.1
36  */
37 void InitTimer1()
38 {
39     char name[MAX_NAME_LEN + 1];
```

Рисунок 8 – Пример использования тега @ingroup

– @weakgroup id @{ ... @} – добавление функций, содержащихся между скобками @{ ... @} к созданной группе id. Является аналогом @addtogroup;

– @param – описание параметра функции id. При описании функции в рамках СИА необходимо описать ее параметры и возвращаемое значение;

– @return – описание возвращаемого значения функции. Пример использования тегов @param и @return показан на рисунке 9;

– @retval val – описание возвращаемого значения/значений функции. Является аналогом @return. Пример использования тега показан на рисунке 10;

```
21 /**
22  * @ingroup rcv
23  * @brief Функция предназначена для приема данных от АРМ
24  * @param *host_name указатель на структуру, содержащую
25  * адрес хоста
26  * @return len длина принятого сообщения
27  *
28  *
29  */
30 void RcvFromArm(const char *host_name)
31 {
32     char dst;
```

Рисунок 9 – Пример использования тегов @param и @return

– @short – краткое описание сущности (функции, структуры, класса, файла и др.). Аналог @brief. В рамках СИА данный тег используется в случае описания функциональной задачи (краткое описание);

– @details – детальное описание сущности (функции, структуры, класса, файла и др.). Данный тег используется в случае описания функциональной задачи (подробное описание, восстановление алгоритма выполнения функциональной задачи). Тег @details аналогичен двум переносам строки после тега @brief;

```
130 /**
131  * @ingroup work
132  * @brief Функция предназначена для обработки сообщения от АРМ
133  * @param *buf указатель на структуру, содержащую
134  * принятое сообщение
135  * @retval 0 сообщение не обработано
136  * @retval 1 сообщение успешно обработано
137  */
138 void TransArm(const char *buf)
139 {
```

Рисунок 10 – Пример использования тега @retval

– @todo – описание запланированной задачи;

- @a <word>, @c <word>, <em> <text> </em> – выделение слова (a)
- курсивом, (c) – полужирным шрифтом, выделение фрагмента текста курсивом;
- @warning, @attention – оформление блока «Предупреждение»;
- @bug – оформление блока «Ошибка»;
- @sa – оформление блока «См. также», в который включаются ссылки на задокументированные модули, функции, структуры;
- @cond, @endcond – оформление блока, исключаемого из документации;
- @fn – индикатор описания функции;
- @version – описание версии проекта;
- @li – оформление блока текста в виде маркированного списка.

При оформлении маркированного списка внутри описания функциональной задачи вместо тега @li можно использовать символ «—».

#### 4.5 Использование средства документирования кода Doxygen

При анализе программного обеспечения средство документирования Doxygen может использоваться для формирования документации по нескольким разделам, к примеру, разделам «Состав ПО», «Восстановление алгоритма работы» и т.д.

Перед началом документирования программного обеспечения необходимо проверить файлы исходного кода на наличие комментариев разработчика, и при их наличии отформатировать эти комментарии. Это можно сделать следующим образом:

- открыть проект в IDE Eclipse, выполнить команду поиска по файлам (Ctrl+N);
- в качестве значений для поиска последовательно ввести «/\*\*», «/\*!», «/\*!», «///», для каждого значения выполнить команду «Replace»;

– если значения найдены, заменить на «/\*», «/\*», «//» и «//» соответственно.

Комментарии разработчика необходимо отформатировать для корректного оформления документации, так как плагин Doxygen при ее формировании воспринимает данные теги как собственные, и, соответственно, включает комментарии разработчика в текст генерируемого отчета.

При формировании подраздела «Состав ПО» при анализе программного обеспечения необходимо привести таблицу с перечнем модулей исходного кода и описанием каждого модуля. Для создания такой таблицы необходимо для каждого файла исходного кода текущего проекта программного обеспечения написать комментарий в формате @file.

Далее необходимо выбрать модули для анализа программы. Рассматриваемая программа межсетевого экрана из состава рассмотренной на первом этапе ТКС СН состоит из модулей, схема взаимодействия которых представлена на рисунке 11.

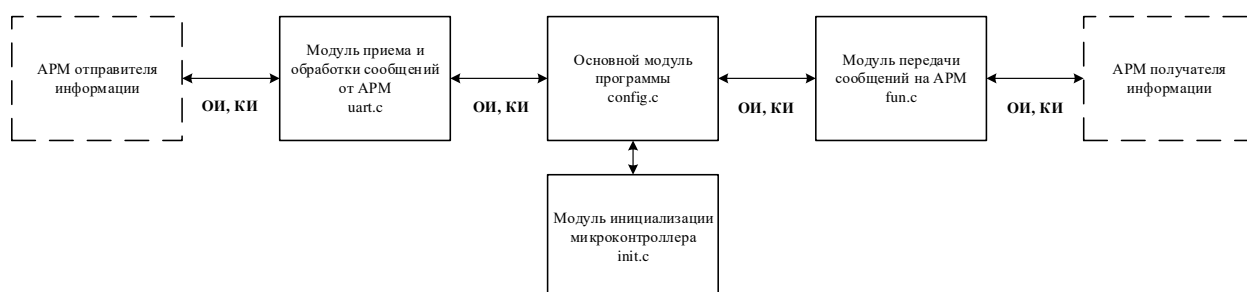


Рисунок 11 – Схема взаимодействия модулей программы межсетевого экрана

После того, как в каждый файл исходного кода добавлен комментарий с описанием модуля, необходимо сгенерировать документацию. Для этого, в IDE Eclipse на панели быстрого доступа нужно выбрать функцию “BuildDocumentation” (@). При этом в каталоге «doxysettings» сгенерируется файл DoxygenOut/html/index.html. Во вкладке «Файлы» данного документа расположена таблица для экспорта в отчет СИА. Для того, чтобы экспортировать таблицу, необходимо после нажатия правой кнопкой мыши

выбрать опцию «Export Table to Excel». После этого необходимо скопировать таблицу в документ Microsoft Word и запустить макрос «TabModules». Макрос отредактирует таблицу в соответствии с требованиями группы стандартов ЕСПД и требованиями к оформлению отчетов в рамках СИА. По окончании редактирования таблицы она подлежит включению в отчет о проведении анализа исходного кода программного обеспечения в разделе «Состав ПО».

При формировании подраздела «Восстановление алгоритма работы», согласно требованиям к анализу ПО в рамках СИА, необходимо привести описание основных функциональных задач, решаемых ПО, а также функций, реализующих указанные задачи.

Для восстановления алгоритма работы программы, в отдельно создаваемом файле определяется перечень основных функциональных задач, реализованных в программе. Для каждой функциональной задачи из перечня необходимо привести подробное описание. Для каждой функциональной задачи определяется специальный тег с помощью команды \defgroup.

С помощью команды \brief приводится краткое описание функциональной задачи, затем с помощью команды \details – ее подробное описание. Пример описания функциональной задачи приведен на рисунке 12.

```
22 /**
23  * @defgroup init Инициализация устройства
24  * @brief В процессе выполнения данной функциональной задачи
25  * осуществляется инициализация программно-аппаратных
26  * компонентов микроконтроллера межсетевых экранов в следующей
27  * последовательности: инициализация векторов прерываний,
28  * задания тактовой частоты, установка значения
29  * таймера по умолчанию...
30  *
31  * В результате выполнения функциональной задачи микроконтроллер
32  * инициализируется и переходит в штатный режим функционирования.
33  */
```

Рисунок 12 – Пример описания функциональной задачи

После определения тега функциональной задачи, необходимо отметить этим тегом все функции реализующие данную задачу. Это можно сделать с помощью команды @ingroup.

Одна и та же функция не может быть отнесена к нескольким группам с использованием специальных команд Doxygen, поэтому при определении принадлежности функции двум и более группам, необходимо включить ее в группу, объявленную в файле ранее остальных, после чего уже в сгенерированном отчете (по окончании документирования кода) вручную добавить ее в число функций каждой из функциональных задач, которым данная функция также принадлежит. Если же таких функций больше четырех, их следует выносить в отдельную функциональную задачу, которая включает в себя функции общего назначения.

Также необходимо объявить функциональную задачу, в которой будет указана точка входа в программу, описать механизмы взаимодействия частей программы, описать псевдокод основной функции, указать основные задачи функционирования СПО.

Для каждой из выделенных тегом функций должно быть приведено описание этих функций, включающее в себя:

- краткое описание функции (тег @brief);
- входные параметры (тег @param);
- возвращаемое значение (тег @retval);
- описание алгоритма работы (тег @details можно опустить, начав описание с новой строки, относительно предыдущей команды).

Пример описания функции приведен на рисунке 13.

После того, как описаны все функции, выделены этапы, решающие основные функциональные задачи программы, необходимо сгенерировать отчет. Для этого, в IDE Eclipse на панели быстрого доступа выбрать функцию «Build Documentation» (@). При этом в папке проекта

«%директория\_проекта%/doxysettings/DoxygenOut/rtf»

появится

сгенерированный файл refman.rtf.

```
47 /**
48  * @brief Функция предназначена для инициализации
49  * программно-аппаратных ресурсов технического средства
50  *
51  * @param status признак состояния технического средства
52  * @retval 0 успешное завершение функции
53  *
54  * Данная функция определяет активные аппаратно-программные
55  * ресурсы модуля, после чего для этих ресурсов вызываются
56  * соответствующие функции инициализации.
57  *
58  * В результате выполнения функции выполняется инициализация
59  * программно-аппаратных ресурсов технического средства.
60  */
61 void init(int status)
62 {
```

Рисунок 13 – Пример описания функции

После того, как описаны все функции, выделены этапы, решающие основные функциональные задачи программы, необходимо сгенерировать отчет. Для этого, в IDE Eclipse на панели быстрого доступа выбрать функцию «Build Documentation» (@). При этом в папке проекта «%директория\_проекта%/doxysettings/DoxygenOut/rtf» появится сгенерированный файл refman.rtf.

После этого необходимо открыть документ и запустить макрос «StyleChng», дождаться сообщения "Редактирование завершено". Отчет автоматически будет отредактирован в соответствии с группой стандартов ЕСПД. Отредактированному отчету присваивается имя refman.doc.

В сформированный в результате выполнения скрипта отчет refman.doc необходимо после таблиц в соответствующих подразделах описания функциональных задач вставить графы вызовов, построенные в процессе анализа ПО.

Принцип построения отчета с помощью средства документирования кода показан на рисунке 14 в виде UML-диаграммы.

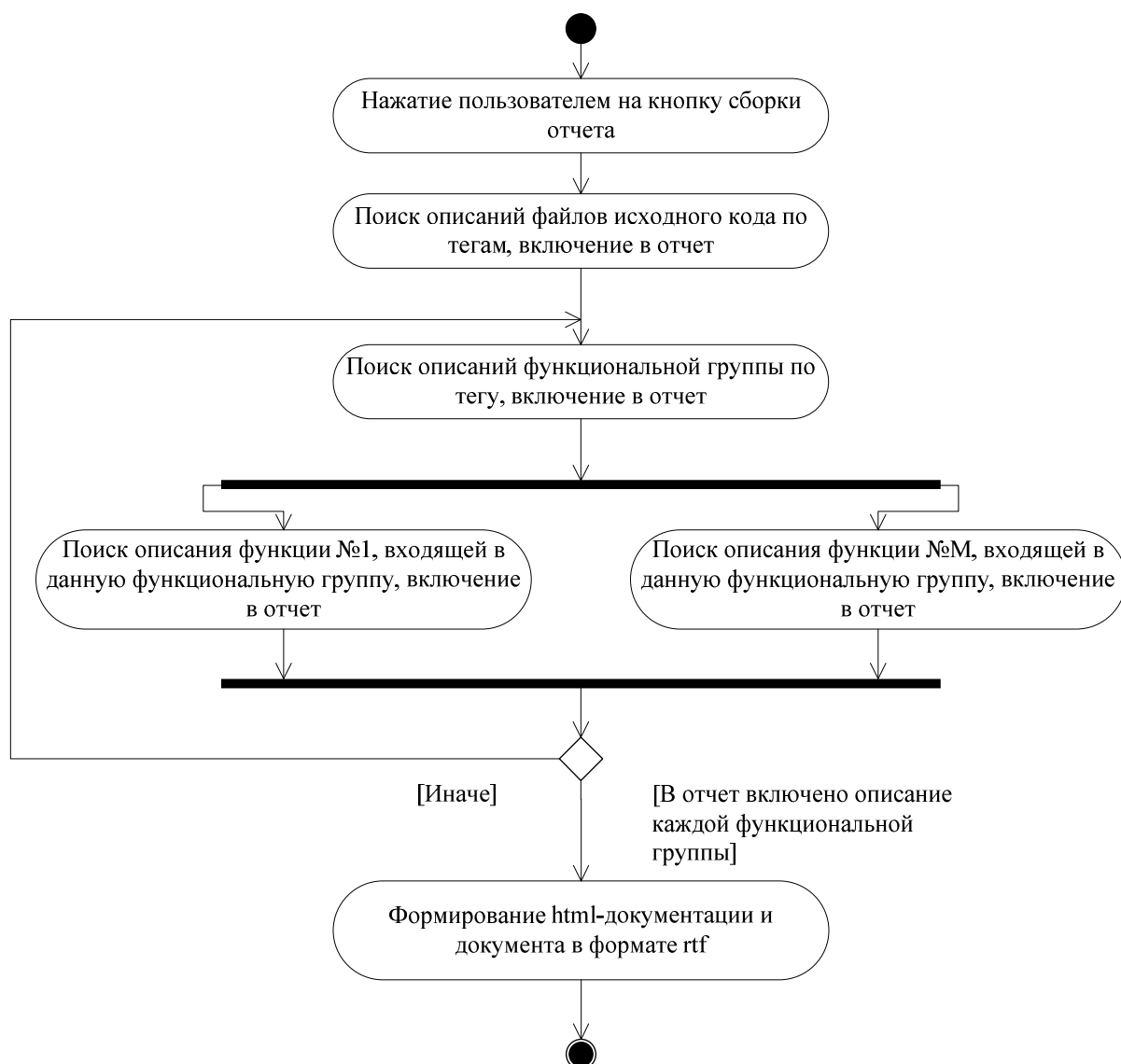


Рисунок 14 – UML-диаграмма последовательности действий программного средства документирования кода

#### 4.6 Полезные функции IDE Eclipse

При комментировании функции в формате Doxygen, IDE Eclipse автоматически создает блок комментария с тегами входных и возвращаемых значений. После этого пустые поля тегов заполняются описанием сущностей. Пример автоматически созданного блока комментария приведен на рисунке 15.

```

46 /**
47  *
48  * @param pwf1w1b
49  * @param word
50  * @param byte
51  * @return
52  */
53 int qpac1W1B(pwf1w1b, word, byte)

```

Рисунок 15 – Пример автоматически созданного блока комментария

Помимо этого можно использовать механизм шаблонов IDE Eclipse для создания собственных шаблонов. Для создания шаблона необходимо перейти в окно Windows ->Preference->C/C++->Editor->Templates, выбрать опцию «New». В окне New Template в поле Name ввести идентификатор шаблона, в поле Pattern - определить сам шаблон. Пример шаблона приведен на рисунке 16.

После того, как шаблон создан, его можно использовать следующим образом: ввести идентификатор шаблона (в рассмотренном примере filedoc), после чего, комбинацией клавиш ctrl+space заменить его на созданный шаблон. Поле \${file} изменится на название текущего файла, назначение которого необходимо описать в следующем поле.

По окончании процесса описания программного кода тегами Doxygen выполняется автоматическая сборка отчета Doxygen. Помимо отчета в формате .rtf генерируется документация и в формате .html для более наглядного отображения выделенных функциональных задач и входящих в них методов.

В процессе описания программного кода целесообразно использовать возможности средства контроля версий Git. Это позволяет синхронизировать процесс с остальными участниками анализа программного обеспечения во избежание повторного описания уже документированных элементов программного кода.

Восстановление алгоритма программы технического средства для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения позволяет выявить опасные функциональные конструкции в составе программы технического средства из состава телекоммуникационной системы специального назначения.

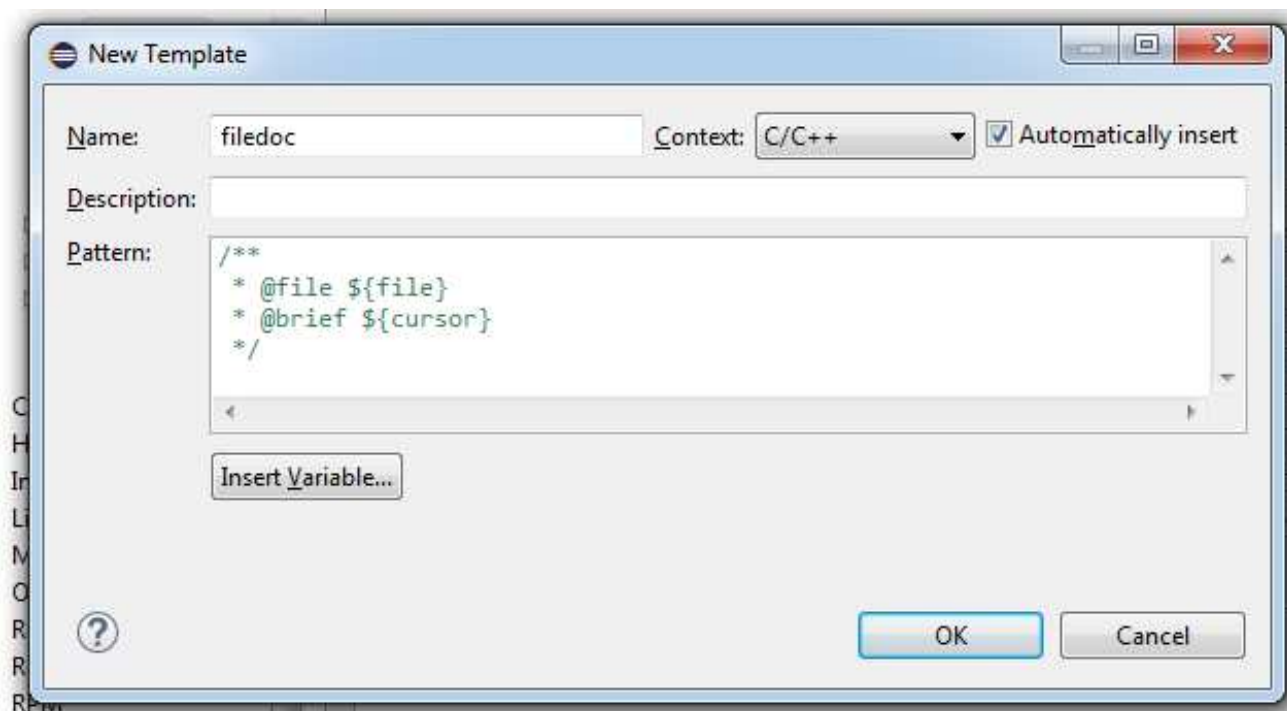


Рисунок 16 – Пример создания шаблона

Полученная при восстановлении алгоритма программы информация подлежит дальнейшему анализу. Рассматриваются возможные последствия нештатного функционирования программы в процессе реализации основных функциональных задач.

При восстановлении алгоритма программы межсетевого экрана были выделены следующие функциональные задачи:

- инициализация аппаратно-программных ресурсов микроконтроллера межсетевого экрана;
- прием информации для передачи и команд от АРМ;
- обработка пакетов информации, отправка сообщений-квитанций на АРМ-отправитель информации;

- трансляция информации к АРМ-получателю.

Выявлены следующие возможные последствия нештатного функционирования программы при реализации основных функциональных задач:

- отклонения от штатного алгоритма работы при инициализации аппаратно-программных ресурсов микроконтроллера могут привести только к некорректной инициализации микроконтроллера и, как следствие, к ошибке при загрузке межсетевого экрана, что не является опасным событием;

- отклонения от штатного алгоритма работы при приеме информации и команд от АРМ могут привести к искажению информации и команд и, как следствие, невыполнению либо некорректному выполнению команд, что является опасным событием;

- отклонения от штатного алгоритма работы при обработке пакетов информации и отправке ответов-квитанций на АРМ-отправитель информации, и при трансляции информации к АРМ-получателю могут привести к ошибочной обработке конфиденциальной информации и дальнейшей ошибочной передаче ее на АРМ, обработка конфиденциальной информации на котором запрещена.

По окончании анализа восстановленного алгоритма программы производится построение схемы восстановленного алгоритма программы.

Пример схемы восстановленного алгоритма программы технического средства, в частности, межсетевого экрана из состава телекоммуникационной системы специального назначения, показан на рисунке 17. В квадратных скобках на схеме указаны программные модули, в которых реализован тот или иной процесс.

В результате выполнения процесса восстановления алгоритма программного обеспечения формируется отчет, содержащий структурированное описание функциональных задач, критичных функций и прочих элементов программного кода, выводы по результатам

восстановления алгоритма программ технического средства для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения.

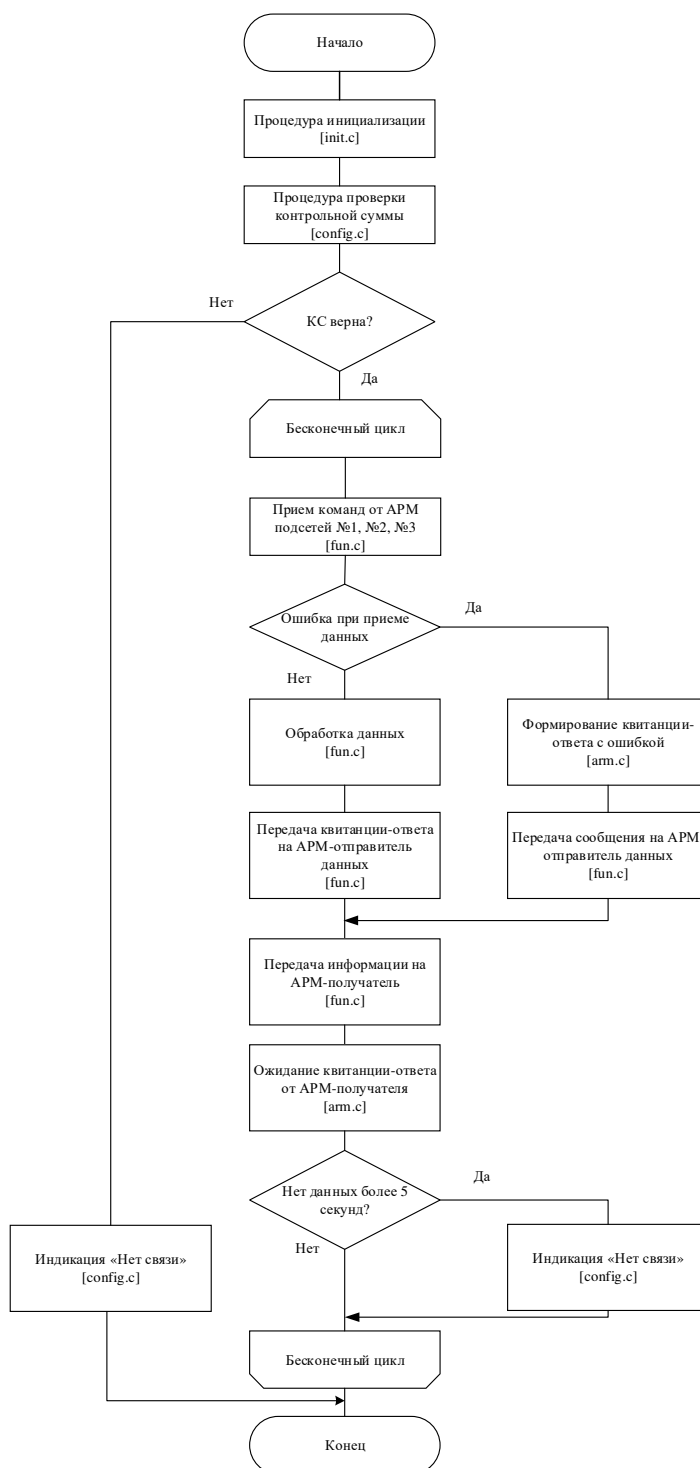


Рисунок 17 – Схема восстановленного алгоритма программы межсетевого экрана

Таким образом, на четвертом этапе было приведено описание процесса настройки и развертывания программных средств, используемых в процессе восстановления алгоритма программы технического средства для оценки возможности появления опасных событий в телекоммуникационной системе специального назначения, а также документирования кода с использованием возможностей Doxygen.

## 5 Методика восстановления алгоритма программы

### 5.1 Общие положения

Методика восстановления алгоритма программы разработана в целях обеспечения единого методологического подхода к восстановлению алгоритма программы технического средства из состава телекоммуникационной системы специального назначения.

Объектом применения методики является программный код, реализующий функционирование специального программного обеспечения технического средства, входящего в состав телекоммуникационной системы специального назначения.

Применение методики заключается в проведении процедуры восстановления алгоритма программного обеспечения на основании исходного кода данного программного обеспечения.

Данная методика предназначена для применения в рамках проведения анализа программного обеспечения на соответствие требованиям, предъявляемым к техническим средствам из состава телекоммуникационных систем специального назначения.

### 5.2 Цель методики

Целью методики является обеспечение единого методологического подхода к восстановлению алгоритма программного обеспечения технического средства из состава телекоммуникационной системы специального назначения.

### 5.3 Основные положения методики

#### 5.3.1 Развертывание необходимого программного обеспечения

Одной из задач процесса восстановления алгоритма программного обеспечения является развертывание на персональной электронно-вычислительной машине аналитика необходимого программного обеспечения, в частности:

- среды разработки IDE Eclipse для удобного просмотра исходного кода, а также навигации по нему;
- плагина средства документирования Doxygen, позволяющего документировать код внутри среды разработки IDE Eclipse, используя его функциональные возможности;
- плагина средства контроля версий Git, позволяющего организовать работу аналитиков, проводящих восстановление алгоритма программы, в команде с использованием собственных функциональных возможностей;
- плагина средства визуализации графов Graphviz, позволяющего производить построение графов, наглядно демонстрирующих алгоритм функционирования программного обеспечения;

Кроме установки приведенного выше программного обеспечения, необходимо добавить в директорию проекта исходного кода программы конфигурационные файлы Doxygen, а также подключить к Microsoft Word макрос, редактирующий документацию, сгенерированную Doxygen, по требованиям группы стандартов ЕСПД.

### 5.3.2 Создание и настройка проекта исходного кода программного обеспечения

В среде разработки Eclipse создается новый проект, в который включаются файлы исходного кода программы, а также конфигурационные файлы Doxygen и шаблон Microsoft Word, соответствующий требованиям группы стандартов ЕСПД.

### 5.3.3 Документирование исходного кода

Документирование исходного кода программного обеспечения производится с использованием встроенных возможностей средства документирования кода Doxygen.

Перед началом документирования исходного кода аналитик выделяет основные функциональные задачи, решаемые программным обеспечением, которые требуют более глубокого анализа.

После того, как основные функциональные задачи определены, следует этап рассмотрения и документирования функций, входящих в каждую из функциональных задач, а также их анализ на предмет наличия в исходном коде ошибок, закладок и потенциально уязвимых функциональных конструкций.

В процессе документирования кода аналитик должен вести ежедневный отчет о проделанной работе с использованием функциональных возможностей плагина средства контроля версий Git. Данный подход позволяет аналитику эффективно работать в команде, избегая документирования одних и тех же участков исходного кода несколькими аналитиками, а также обеспечивает возможность восстановления предыдущих версий исходного кода с внесенными комментариями в случае непредвиденных сбоев и неисправностей. Также в процессе документирования кода аналитику рекомендуется по собственному усмотрению воспроизводить взаимодействие конструкций исходного кода в виде графов с использованием средства визуализации графов Graphviz.

По результатам рассмотрения всех функциональных задач программы технического средства формируются выводы по результатам восстановления алгоритма программы технического средства, содержащие сведения о соответствии восстановленного алгоритма программы технического средства реально документированным возможностям программы, наличии опасных функциональных конструкций и потенциально уязвимых мест, наличии функциональных конструкций, обеспечивающих защиту от ошибок программы, возникающих в результате сбоев и неисправностей аппаратных компонентов.

#### 5.3.4 Редактирование отчета

По окончании документирования исходного кода программного обеспечения аналитик с использованием плагина средства документирования генерирует отчет, содержащий все описанные аналитиком функциональные конструкции программного обеспечения.

В результате выполнения процесса восстановления алгоритма программного обеспечения формируется отчет, содержащий структурированное описание функциональных задач, критичных функций и прочих элементов программного кода, выводы по результатам восстановления алгоритма программ технического средства для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения.

Для автоматического редактирования отчета в формат, соответствующий требованиям группы стандартов ЕСПД, аналитику необходимо применить соответствующий макрос, скрипт которого приведен в приложении А.

По окончании редактирования документации получен отчет, соответствующий требованиям группы стандартов ЕСПД, содержащий описание конструкций языка, участвующих в процессе выполнения выбранных для анализа функциональных задач, решаемых программным обеспечением.

UML-диаграмма последовательности действий при восстановлении алгоритма программ для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения показана на рисунке 18.

Полученная при восстановлении алгоритма программы информация подлежит дальнейшему анализу. Рассматриваются возможные последствия нештатного функционирования программы в процессе реализации основных функциональных задач.

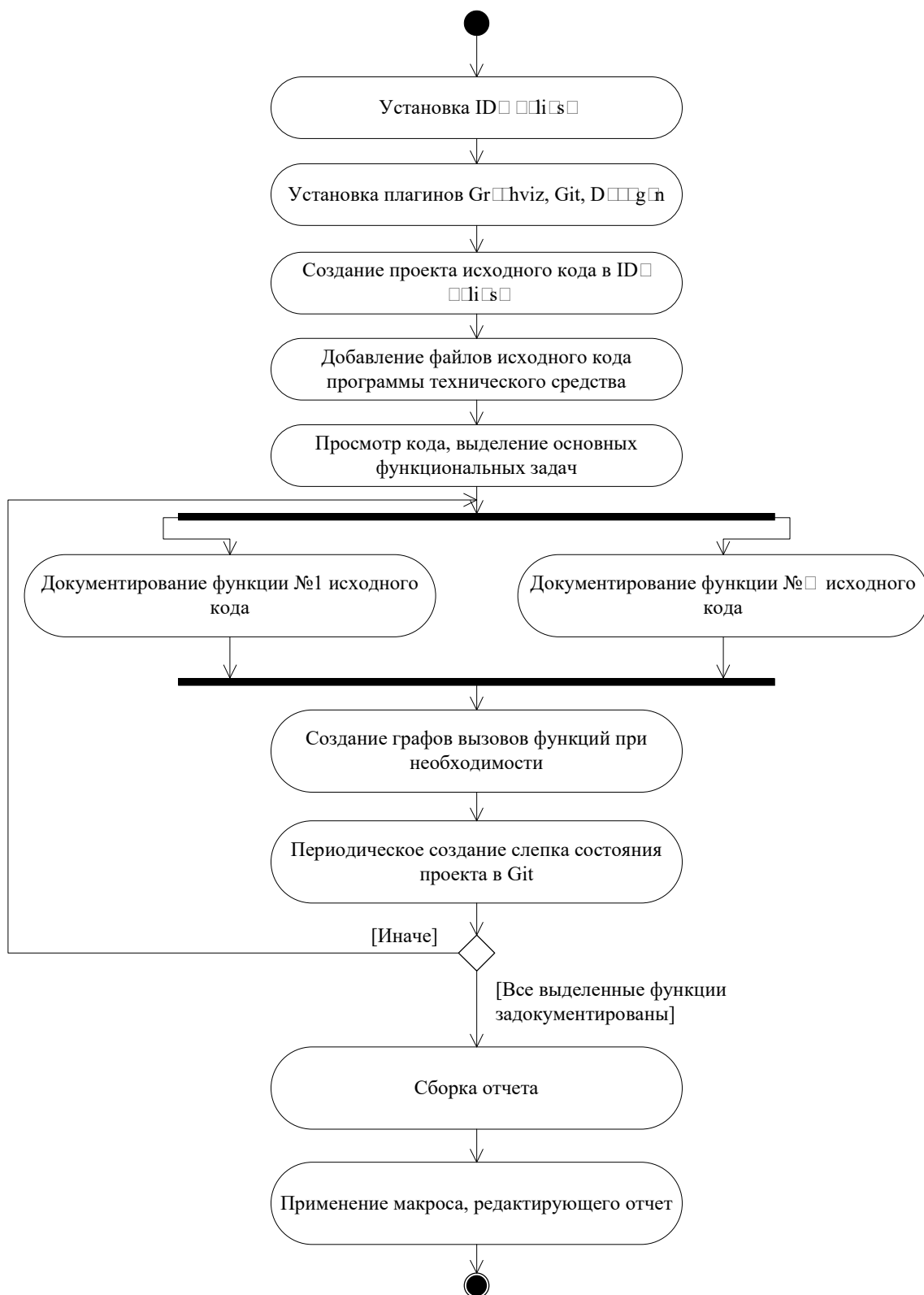


Рисунок 18 – UML-диаграмма последовательности действий при восстановлении алгоритма программы

Разработанная в рамках дипломного проекта методика позволяет автоматизировать процесс восстановления алгоритма программ. Восстановление алгоритма программы технического средства для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения позволяет сделать вывод о степени возможности возникновения канала утечки информации при ошибках программы, возникающих в результате сбоев и неисправностей в телекоммуникационной системе специального назначения.

Таким образом, методика выявления потенциальных каналов утечки информации в ТКС СН, возникающих в результате сбоев и неисправностей технических средств, может применяться при анализе ТКС СН на соответствие требованиям информационной безопасности.

## ЗАКЛЮЧЕНИЕ

В рамках дипломного проекта был разработан методический подход к восстановлению алгоритма программ для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения.

Данный методический подход основан на разработанной схеме информационных потоков ТКС СН, по результатам анализа которой определяются опасные события для данной телекоммуникационной системы специального назначения.

Восстановление алгоритма программ для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения выполняется путем документирования исходного кода в части функциональных задач программы технического средства, и последующего анализа полученных результатов и формирования выводов по результатам анализа.

Таким образом, можно сделать вывод о возможном применении данной методики восстановлению алгоритма программ для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения при анализе ТКС СН на соответствие требованиям информационной безопасности.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 ГОСТ Р 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения. [Электронный ресурс]. URL: <https://polyset.ru/GOST/all-doc/GOST/GOST-34-003-90/> (дата обращения: 03.12.2016 г.).

2 Федеральный закон «О связи» от 07.07.2003 N 126-ФЗ. [Электронный ресурс]. URL: <http://docs.cntd.ru/document/901867280> (дата обращения: 03.12.2016 г.).

3 ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. [Электронный ресурс]. URL: <http://docs.cntd.ru/document/1200058320> (дата обращения: 19.09.2016 г.).

4 Руководящий документ ФСТЭК. Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. [Электронный ресурс]. URL: <http://fstec.ru/component/attachments/download/295> (дата обращения: 20.09.2016 г.).

5 ГОСТ Р МЭК 61508-4-2012 Функциональная безопасность систем электрических, электронных, программируемых электронных связанных с безопасностью. Часть 4. Термины и определения. [Электронный ресурс]. URL: <http://www.gosthelp.ru/gost/gost44280.html> (дата обращения: 11.10.2016 г.).

6 Руководящий документ ФСТЭК. Средства вычислительной техники. Защита от несанкционированного доступа к информации. [Электронный ресурс]. URL: <http://fstec.ru/component/attachments/download/297> (дата обращения: 13.12.2016 г.).

7 Оформление документации в Doxygen. [Электронный ресурс]. URL: <https://habrahabr.ru/post/252443/> (дата обращения: 13.12.2016 г.).

8 Документируем код эффективно при помощи Doxygen. [Электронный ресурс]. URL: <https://habrahabr.ru/post/252101/> (дата обращения: 13.12.2016 г.).

## ПРИЛОЖЕНИЕ А

(обязательное)

Листинг макроса, редактирующего отчет в соответствии с требованиями  
группы стандартов ЕСПД

```
Sub StyleChng()  
,  
' StyleChng Макрос  
,  
,  
  
Selection.Find.ClearFormatting  
Selection.Find.Replacement.ClearFormatting  
With Selection.Find  
    .Text = "^p^p"  
    .Replacement.Text = "^p"  
    .Forward = True  
    .Wrap = wdFindContinue  
    .Format = False  
    .MatchCase = False  
    .MatchWholeWord = False  
    .MatchWildcards = False  
    .MatchSoundsLike = False  
    .MatchAllWordForms = False  
End With  
Selection.Find.Execute Replace:=wdReplaceAll  
Selection.Find.ClearFormatting  
With Selection.Find  
    .Text = "Группы"  
    .Replacement.Text = "^p"  
    .Forward = True  
    .Wrap = wdFindContinue  
    .Format = False  
    .MatchCase = False  
    .MatchWholeWord = False
```

```

.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveRight Unit:=wdCharacter, Count:=1
Selection.MoveLeft Unit:=wdCharacter, Count:=6
Selection.MoveUp Unit:=wdLine, Count:=1
Selection.MoveDown Unit:=wdLine, Count:=1
Selection.MoveLeft Unit:=wdCharacter, Count:=1
Selection.HomeKey Unit:=wdStory, Extend:=wdExtend
Selection.Delete Unit:=wdCharacter, Count:=1
Selection.Delete Unit:=wdCharacter, Count:=1

Selection.Find.ClearFormatting
With Selection.Find
    .Text = "Файлы"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=2
Selection.EndKey Unit:=wdStory, Extend:=wdExtend
Selection.Delete Unit:=wdCharacter, Count:=1

Set y = CreateObject("Scripting.FileSystemObject").GetFolder(ActiveDocument.Path)
y = y.ParentFolder

```

```
y = Left(y, InStrRev(y, "\")) & "espdv3-doxxygen.dotx"
```

```
With ActiveDocument
```

```
    .UpdateStylesOnOpen = True
```

```
    .AttachedTemplate = _
```

```
    y
```

```
End With
```

```
""""Добавлено
```

```
Selection.Find.ClearFormatting
```

```
Selection.Find.Style = ActiveDocument.Styles("Стиль маркированный1")
```

```
Selection.Find.Replacement.ClearFormatting
```

```
Selection.Find.Replacement.Style = ActiveDocument.Styles("Стиль маркированный")
```

```
With Selection.Find
```

```
    .Text = ""
```

```
    .Replacement.Text = ""
```

```
    .Forward = False
```

```
    .Wrap = wdFindEnd
```

```
    .Format = True
```

```
    .MatchCase = False
```

```
    .MatchWholeWord = False
```

```
    .MatchWildcards = False
```

```
    .MatchSoundsLike = False
```

```
    .MatchAllWordForms = False
```

```
End With
```

```
Selection.Find.Execute Replace:=wdReplaceAll
```

```
Selection.Find.ClearFormatting
```

```
Selection.Find.Replacement.ClearFormatting
```

```
With Selection.Find
```

```
    .Text = "^p^p"
```

```
    .Replacement.Text = "^p"
```

```
    .Forward = True
```

```
    .Wrap = wdFindContinue
```

```
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

""""Добавлено

```
CountTab = ActiveDocument.Tables.Count
I = 1
Do While I < CountTab + 1
ActiveDocument.Tables(I).Select
Selection.Rows.ConvertToText Separator:=wdSeparateByTabs, NestedTables:= _
    True
Selection.Style = ActiveDocument.Styles("Обычный")

CountTab = CountTab - 1
Loop
```

```
Selection.Find.ClearFormatting
With Selection.Find
    .Text = "Группы"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
```

```
End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1
```

```
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
```

```
    .Text = vbTab
    .Replacement.Text = "– "
    .Forward = True
    .Wrap = wdFindAsk
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
```

```
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

```
Selection.Find.ClearFormatting
```

```
With Selection.Find
```

```
    .Text = "Группы"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
```

```
End With
Selection.Find.Execute
```

Selection.MoveLeft Unit:=wdCharacter, Count:=1

CountPar = ActiveDocument.Paragraphs.Count

Do While CountPar > 0

Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend

If Selection.Paragraphs(1).Style = ActiveDocument.Styles("Описание функций: алгоритм выполнения") Then

Selection.Style = ActiveDocument.Styles("Описание функций: алгоритм выполнения")

With Selection.ParagraphFormat

.LeftIndent = CentimetersToPoints(0)  
.RightIndent = CentimetersToPoints(0)  
.SpaceBefore = 0  
.SpaceBeforeAuto = False  
.SpaceAfter = 0  
.SpaceAfterAuto = False  
.LineSpacingRule = wdLineSpace1pt5  
.Alignment = wdAlignParagraphJustify  
.WidowControl = True  
.KeepWithNext = False  
.KeepTogether = False  
.PageBreakBefore = False  
.NoLineNumber = False  
.Hyphenation = True  
.FirstLineIndent = CentimetersToPoints(1.25)  
.OutlineLevel = wdOutlineLevelBodyText  
.CharacterUnitLeftIndent = 0  
.CharacterUnitRightIndent = 0  
.CharacterUnitFirstLineIndent = 0  
.LineUnitBefore = 0  
.LineUnitAfter = 0  
.MirrorIndents = False  
.TextboxTightWrap = wdTightNone  
.CollapsedByDefault = False

End With

Selection.Font.Bold = wdToggle

Selection.Font.Bold = wdToggle

End If

If Selection.Paragraphs(1).Style = ActiveDocument.Styles("Стиль маркированный") Then

Selection.Style = ActiveDocument.Styles("Стиль маркированный")

With Selection.ParagraphFormat

.LeftIndent = CentimetersToPoints(0)

.RightIndent = CentimetersToPoints(0)

.SpaceBefore = 0

.SpaceBeforeAuto = False

.SpaceAfter = 0

.SpaceAfterAuto = False

.LineSpacingRule = wdLineSpace1pt5

.Alignment = wdAlignParagraphJustify

.WidowControl = True

.KeepWithNext = False

.KeepTogether = False

.PageBreakBefore = False

.NoLineNumber = False

.Hyphenation = True

.FirstLineIndent = CentimetersToPoints(1.25)

.OutlineLevel = wdOutlineLevelBodyText

.CharacterUnitLeftIndent = 0

.CharacterUnitRightIndent = 0

.CharacterUnitFirstLineIndent = 0

.LineUnitBefore = 0

.LineUnitAfter = 0

.MirrorIndents = False

.TextboxTightWrap = wdTightNone

.CollapsedByDefault = False

End With

Selection.Font.Bold = wdToggle

Selection.Font.Bold = wdToggle

End If

If Selection.Paragraphs(1).Style = ActiveDocument.Styles("Описание функций: краткое описание") Then

Selection.Style = ActiveDocument.Styles("Описание функций: краткое описание")

With Selection.ParagraphFormat

.LeftIndent = CentimetersToPoints(0)

.RightIndent = CentimetersToPoints(0)

.SpaceBefore = 0

.SpaceBeforeAuto = False

.SpaceAfter = 0

.SpaceAfterAuto = False

.LineSpacingRule = wdLineSpace1pt5

.Alignment = wdAlignParagraphJustify

.WidowControl = True

.KeepWithNext = False

.KeepTogether = False

.PageBreakBefore = False

.NoLineNumber = False

.Hyphenation = True

.FirstLineIndent = CentimetersToPoints(1.25)

.OutlineLevel = wdOutlineLevelBodyText

.CharacterUnitLeftIndent = 0

.CharacterUnitRightIndent = 0

.CharacterUnitFirstLineIndent = 0

.LineUnitBefore = 0

.LineUnitAfter = 0

.MirrorIndents = False

.TextboxTightWrap = wdTightNone

.CollapsedByDefault = False

End With

Selection.Font.Bold = wdToggle

Selection.Font.Bold = wdToggle

End If

If Selection.Paragraphs(1).Style = ActiveDocument.Styles("Секции (подчеркнутый)") Then

With Selection.ParagraphFormat

.LeftIndent = CentimetersToPoints(0)  
.RightIndent = CentimetersToPoints(0)  
.SpaceBefore = 0  
.SpaceBeforeAuto = False  
.SpaceAfter = 0  
.SpaceAfterAuto = False  
.LineSpacingRule = wdLineSpace1pt5  
.Alignment = wdAlignParagraphJustify  
.WidowControl = True  
.KeepWithNext = False  
.KeepTogether = False  
.PageBreakBefore = False  
.NoLineNumber = False  
.Hyphenation = True  
.FirstLineIndent = CentimetersToPoints(1.25)  
.OutlineLevel = wdOutlineLevelBodyText  
.CharacterUnitLeftIndent = 0  
.CharacterUnitRightIndent = 0  
.CharacterUnitFirstLineIndent = 0  
.LineUnitBefore = 0  
.LineUnitAfter = 0  
.MirrorIndents = False  
.TextboxTightWrap = wdTightNone  
.CollapsedByDefault = False

End With

End If

Selection.MoveRight Unit:=wdCharacter, Count:=1

CountPar = CountPar - 1

Loop

```
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
Selection.Find.Execute
With Selection.Find
    .Text = " ^pВозвращаемые"
    .Replacement.Text = ".^pВозвращаемые"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

```
Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Обычный")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = " ^p"
    .Replacement.Text = " ^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
```

Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "..^p"

.Replacement.Text = ".^p"

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting

Selection.Find.Style = ActiveDocument.Styles("Обычный")

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = " ^p"

.Replacement.Text = ";^p"

.Forward = True

.Wrap = wdFindContinue

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute Replace:=wdReplaceAll

```
Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Обычный")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = ";;^p"
    .Replacement.Text = ";;^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

```
Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Обычный")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = ".."
    .Replacement.Text = "."
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

Selection.Find.ClearFormatting

With Selection.Find

.Text = "Группы"

.Replacement.Text = "^p"

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Selection.MoveLeft Unit:=wdCharacter, Count:=1

Selection.Find.ClearFormatting

Selection.Find.Style = ActiveDocument.Styles("Описание функций: краткое описание")

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = " ^p"

.Replacement.Text = ".^p"

.Forward = True

.Wrap = wdFindEnd

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting

With Selection.Find

.Text = "Группы"

.Replacement.Text = "^p"

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Selection.MoveLeft Unit:=wdCharacter, Count:=1

"""" Замена

Selection.Find.ClearFormatting

With Selection.Find

.Text = "Группы"

.Replacement.Text = "^p"

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Selection.MoveLeft Unit:=wdCharacter, Count:=1

Fond = 0

Do While Fond = 0

```

If Fond = 0 Then
Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Подзаголовок 3")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Функции"
    .Replacement.Text = "Перечень функций, входящих в данную функциональную
задачу"
    .Forward = True
    .Wrap = wdFindEnd
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute
If Selection.Find.Found = False Then
Fond = 1
GoTo metk2
End If
With Selection
    If .Find.Forward = True Then
        .Collapse Direction:=wdCollapseStart
    Else
        .Collapse Direction:=wdCollapseEnd
    End If
    .Find.Execute Replace:=wdReplaceOne
    If .Find.Forward = True Then
        .Collapse Direction:=wdCollapseEnd
    Else
        .Collapse Direction:=wdCollapseStart
    End If
End With

```

```
.Find.Execute  
End With  
End If  
metk2: Loop
```

```
Selection.Find.ClearFormatting  
With Selection.Find  
    .Text = "Группы"  
    .Replacement.Text = "^p"  
    .Forward = True  
    .Wrap = wdFindContinue  
    .Format = False  
    .MatchCase = False  
    .MatchWholeWord = False  
    .MatchWildcards = False  
    .MatchSoundsLike = False  
    .MatchAllWordForms = False
```

```
End With  
Selection.Find.Execute  
Selection.MoveLeft Unit:=wdCharacter, Count:=1
```

```
Selection.Find.ClearFormatting  
Selection.Find.Style = ActiveDocument.Styles("Подзаголовок 3")  
Selection.Find.Replacement.ClearFormatting  
With Selection.Find  
    .Text = "Подробное описание"  
    .Replacement.Text = "Подробное описание данной функциональной задачи"  
    .Forward = True  
    .Wrap = wdFindContinue  
    .Format = True  
    .MatchCase = False  
    .MatchWholeWord = False  
    .MatchWildcards = False  
    .MatchSoundsLike = False
```

```

        .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Секции (подчеркнутый)")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Аргументы:"
    .Replacement.Text = "Параметры:"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting
With Selection.Find
    .Text = "Группы"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With

```

```

Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1

Fond = 0
Do While Fond = 0
If Fond = 0 Then
Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Подзаголовок 3")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Перечень функций"
    .Replacement.Text = "Пр"
    .Forward = False
    .Wrap = wdFindEnd
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute
With Selection
    If .Find.Forward = True Then
        .Collapse Direction:=wdCollapseStart
    Else
        .Collapse Direction:=wdCollapseEnd
    End If
    .Find.Execute Replace:=wdReplaceOne
    If .Find.Forward = False Then
        .Collapse Direction:=wdCollapseEnd
    Else
        .Collapse Direction:=wdCollapseStart
    End If

```

```

        .Find.Execute
End With
End If
If Selection.Find.Found = False Then
    Fond = 1
End If
Loop

Fond = 0
Do While Fond = 0
    If Fond = 0 Then
        Selection.Find.ClearFormatting
        Selection.Find.Style = ActiveDocument.Styles("Подзаголовок 3")
        Selection.Find.Replacement.ClearFormatting
        With Selection.Find
            .Text = "Подробное описание функций"
            .Replacement.Text = "Перечень функций, входящих в данную функциональную
задачу"
            .Forward = False
            .Wrap = wdFindEnd
            .Format = False
            .MatchCase = False
            .MatchWholeWord = False
            .MatchWildcards = False
            .MatchSoundsLike = False
            .MatchAllWordForms = False
        End With
        Selection.Find.Execute
        With Selection
            If .Find.Forward = True Then
                .Collapse Direction:=wdCollapseStart
            Else
                .Collapse Direction:=wdCollapseEnd
            End If

```

```

.Find.Execute Replace:=wdReplaceOne
If .Find.Forward = False Then
    .Collapse Direction:=wdCollapseEnd
Else
    .Collapse Direction:=wdCollapseStart
End If
.Find.Execute
End With
End If
If Selection.Find.Found = False Then
    Fond = 1
End If
Loop

Selection.Find.ClearFormatting
With Selection.Find
    .Text = "Группы"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1

Selection.Find.ClearFormatting
With Selection.Find
    .Text = "Группы"
    .Replacement.Text = "^p"

```

```

        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute
    Selection.MoveLeft Unit:=wdCharacter, Count:=1

    Fond = 0
    Do While Fond = 0
    If Fond = 0 Then
    Selection.Find.ClearFormatting
    Selection.Find.Style = ActiveDocument.Styles("Подзаголовок 3")
    Selection.Find.Replacement.ClearFormatting
    With Selection.Find
        .Text = "Подробное описание данной функциональной задачи"
        .Replacement.Text = ""
        .Forward = True
        .Wrap = wdFindEnd
        .Format = True
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute
    Selection.MoveRight Unit:=wdCharacter, Count:=2
    Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
    Selection.Style = ActiveDocument.Styles("Обычный")

```

With Selection

If .Find.Forward = True Then

.Collapse Direction:=wdCollapseStart

Else

.Collapse Direction:=wdCollapseEnd

End If

If .Find.Forward = False Then

.Collapse Direction:=wdCollapseEnd

Else

.Collapse Direction:=wdCollapseStart

End If

End With

End If

If Selection.Find.Found = False Then

Fond = 1

End If

Loop

""""Вставить

Selection.Find.ClearFormatting

With Selection.Find

.Text = "Группы"

.Replacement.Text = "^p"

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Selection.Find.ClearFormatting

With Selection.Find

```
.Text = "Подробное описание данной функциональной задачи"  
.Replacement.Text = "^p"  
.Forward = False  
.Wrap = wdFindContinue  
.Format = False  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False
```

End With

Selection.Find.Execute

Selection.MoveLeft Unit:=wdCharacter, Count:=1

Selection.MoveDown Unit:=wdParagraph, Count:=ParCount \* 2 + 2, Extend:=wdExtend

""Добавлено

Selection.Cut

Selection.Delete Unit:=wdCharacter, Count:=1

Selection.Find.ClearFormatting

Selection.Find.Style = ActiveDocument.Styles("Подзаголовков 2")

Selection.Find.Replacement.ClearFormatting

With Selection.Find

```
.Text = ""  
.Replacement.Text = ""  
.Forward = False  
.Wrap = wdFindContinue  
.Format = True  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False
```

End With

Selection.Find.Execute

Selection.MoveLeft Unit:=wdCharacter, Count:=1

Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend

Selection.MoveRight Unit:=wdCharacter, Count:=1

Selection.MoveRight Unit:=wdCharacter, Count:=1

Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend

Selection.Delete Unit:=wdCharacter, Count:=1

Selection.MoveUp Unit:=wdLine, Count:=1

Selection.PasteAndFormat (wdFormatOriginalFormatting)

Selection.MoveDown Unit:=wdLine, Count:=1

Selection.Find.ClearFormatting

Selection.Find.Style = ActiveDocument.Styles("Секции (подчеркнутый)")

Selection.Find.Replacement.ClearFormatting

Selection.Find.Replacement.Style = ActiveDocument.Styles("Подзаголовок 3")

With Selection.Find

.Text = "Алгоритм работы:"

.Replacement.Text = \_

"Алгоритм выполнения данной функциональной задачи"

.Forward = False

.Wrap = wdFindEnd

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

With Selection

If .Find.Forward = True Then

.Collapse Direction:=wdCollapseStart

Else

.Collapse Direction:=wdCollapseEnd

```

End If
.Find.Execute Replace:=wdReplaceOne
If .Find.Forward = True Then
    .Collapse Direction:=wdCollapseEnd
Else
    .Collapse Direction:=wdCollapseStart
End If
End With

I = I + 1
Loop
With Selection.Find
    .Text = "Подробное описание данной функциональной задачи"
    .Replacement.Text = "Краткое описание данной функциональной задачи"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = ".,;"
    .Replacement.Text = "."
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False

```

```

.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1

```

```

Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find

```

```

    .Text = "Функциональные задачи"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False

```

```

End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1

```

""""Добавление названий рисунков и ссылок

```
CountTab = ActiveDocument.Tables.Count
```

```
I = 1
```

```
Do While I < CountTab + 1
```

```
ActiveDocument.Tables(I).Select
```

```
Selection.MoveRight Unit:=wdCharacter, Count:=1
```

```
Selection.TypeParagraph
```

```
Selection.TypeText Text:="Схема алгоритма выполнения данной функциональной задачи
представлена на рисунке ниже ()."
```

```

Selection.MoveUp Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Selection.MoveLeft Unit:=wdCharacter, Count:=1
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Selection.Style = ActiveDocument.Styles("Обычный")
Selection.MoveRight Unit:=wdCharacter, Count:=1
Selection.TypeParagraph
Selection.TypeParagraph
Selection.MoveUp Unit:=wdLine, Count:=1
Selection.InsertCaption Label:="Рисунок", TitleAutoText:="InsertCaption2", _
    Title:="", Position:=wdCaptionPositionBelow, ExcludeLabel:=0
Selection.MoveUp Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Selection.MoveLeft Unit:=wdCharacter, Count:=1
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Selection.Style = ActiveDocument.Styles("Название рисунка")
Selection.MoveUp Unit:=wdParagraph, Count:=2, Extend:=wdExtend
Selection.MoveLeft Unit:=wdCharacter, Count:=4
Selection.InsertCrossReference ReferenceType:="Рисунок", ReferenceKind:= _
    wdEntireCaption, ReferenceItem:=I, InsertAsHyperlink:=True, _
    IncludePosition:=False, SeparateNumbers:=False, SeparatorString:=" "
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Selection.MoveRight Unit:=wdCharacter, Count:=1
Selection.MoveLeft Unit:=wdCharacter, Count:=1
Selection.TypeText Text:=". Схема алгоритма выполнения функциональной задачи "
I = I + 1
Loop

Selection.Find.ClearFormatting
With Selection.Find
    .Text = "Функциональные задачи"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue

```

```

.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1
CountTab = ActiveDocument.Tables.Count
I = 1
Do While I < CountTab + 1
Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Подзаголовок 2")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
.Text = ""
.Replacement.Text = ""
.Forward = True
.Wrap = wdFindContinue
.Format = True
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
If I = 1 Then
Selection.Find.Execute
End If
Selection.Find.Execute
Selection.Copy
Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Название рисунка")

```

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Selection.MoveRight Unit:=wdCharacter, Count:=1

Selection.MoveLeft Unit:=wdCharacter, Count:=1

Selection.TypeText Text:="\*\*\*\*\*"

Selection.MoveLeft Unit:=wdCharacter, Count:=1

Selection.PasteAndFormat (wdFormatOriginalFormatting)

Selection.TypeBackspace

Selection.MoveRight Unit:=wdCharacter, Count:=1

Selection.MoveUp Unit:=wdParagraph, Count:=1, Extend:=wdExtend

Selection.Style = ActiveDocument.Styles("Название рисунка")

I = I + 1

Loop

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindEnd

.Format = False

.MatchCase = False

.MatchWholeWord = False

```

.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Функциональные задачи"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1

With Selection.Find
    .Text = ""
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindEnd
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False

```

```
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

```
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
```

```
With Selection.Find
    .Text = ""
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindEnd
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
```

```
Selection.Find.Execute Replace:=wdReplaceAll
```

"" Удаление ненужных стилей

```
boolA = False
For Each oSt In ActiveDocument.Styles
    On Error Resume Next
    If Not oSt.NameLocal = "Объявления функций (курсив)" And Not oSt.NameLocal =
"Заголовок 1" And Not oSt.NameLocal = "Заголовок 2" And Not oSt.NameLocal =
"Заголовок 3" And Not oSt.NameLocal = "Заголовок 4" And Not oSt.NameLocal =
"Заголовок 5" And Not oSt.NameLocal = "Заголовок 6" And Not oSt.NameLocal =
"Заголовок 7" And Not oSt.NameLocal = "Заголовок 8" And Not oSt.NameLocal =
"Заголовок 9" And Not oSt.NameLocal = "Название рисунка" And Not oSt.NameLocal =
"Название таблицы" And Not oSt.NameLocal = "Номер страницы" And Not oSt.NameLocal =
"Обычный" And Not oSt.NameLocal = "Подзаголовок 1" And Not oSt.NameLocal =
"Подзаголовок 2" And Not oSt.NameLocal = "Подзаголовок 3" And Not oSt.NameLocal =
"Секции (подчеркнутый)" And Not oSt.NameLocal = "Стиль маркированный" And Not
oSt.NameLocal = "Стиль нумерованный" And Not oSt.NameLocal = "Таблица текст" And
```

```

Not oSt.NameLocal = "Таблица шапка" And Not oSt.NameLocal = "Текст программ" And
Not oSt.NameLocal = "Функции" And Not oSt.NameLocal = "Сетка таблицы" _
And Not oSt.NameLocal = "Описание функций: алгоритм выполнения" And Not
oSt.NameLocal = "Описание функций: краткое описание" Then
    If Not oSt.BuiltIn Then
        oSt.Delete
    End If
    If Not oSt.InUse Then
        ActiveDocument.Styles(st).Visibility = True
    End If
    If oSt.BuiltIn Then
        oSt.Visibility = True
    End If
    If oSt.InUse Then
        oSt.Visibility = True
    End If
End If
Next

CountTab = ActiveDocument.Tables.Count
I = 1
Do While I < CountTab + 1
    Selection.Find.ClearFormatting
    Selection.Find.Style = ActiveDocument.Styles("Подзаголовков 2")
    Selection.Find.Replacement.ClearFormatting
    With Selection.Find
        .Text = ""
        .Replacement.Text = ""
        .Forward = True
        .Wrap = wdFindContinue
        .Format = True
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
    End With
    Selection.Find.Execute
    I = I + 1
Loop

```

```

        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    If I = 1 Then
        Selection.Find.Execute
    End If
    Selection.Find.Execute
    Selection.Copy
    Selection.Find.ClearFormatting
    Selection.Find.Style = ActiveDocument.Styles("Название рисунка")
    Selection.Find.Replacement.ClearFormatting
    With Selection.Find
        .Text = ""
        .Replacement.Text = ""
        .Forward = True
        .Wrap = wdFindContinue
        .Format = True
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute
    Selection.MoveRight Unit:=wdCharacter, Count:=1
    I = I + 1
    Loop

    Selection.Find.ClearFormatting
    Selection.Find.Replacement.ClearFormatting
    With Selection.Find
        .Text = "Функциональные задачи"
        .Replacement.Text = "^p"
        .Forward = True

```

```

.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1

Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Стиль маркированный")
Selection.Find.Replacement.ClearFormatting
Selection.Find.Replacement.Style = ActiveDocument.Styles("Стиль маркированный")
With Selection.Find
.Text = ""
.Replacement.Text = ""
.Forward = True
.Wrap = wdFindContinue
.Format = True
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
.Text = "Функциональные задачи"
.Replacement.Text = "^p"

```

```
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveLeft Unit:=wdCharacter, Count:=1
```

```
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
```

```
.Text = " "
.Replacement.Text = " "
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
```

```
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
```

```
.Text = "."
.Replacement.Text = "."
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
```

```

.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = " ;"
.Replacement.Text = ";"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = " )"
.Replacement.Text = ")"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = " ,"

```

```

.Replacement.Text = ","
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Функциональные задачи"
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.MoveRight Unit:=wdCharacter, Count:=1
Selection.MoveDown Unit:=wdLine, Count:=1
Selection.MoveRight Unit:=wdCharacter, Count:=1
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Do While 1
If Selection.Paragraphs(1).Style = ActiveDocument.Styles("Секции (подчеркнутый)") Then
Selection.Delete Unit:=wdCharacter, Count:=1

```

```

End If
Selection.MoveRight Unit:=wdCharacter, Count:=1
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
If (Selection.Paragraphs.Style = ActiveDocument.Styles("Подзаголовок 2")) Then
GoTo metkaa1
End If
Loop
metkaa1:
Selection.WholeStory
Selection.Font.Bold = wdToggle
Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Секции(подчеркнутый)")
Selection.Find.Replacement.ClearFormatting
Selection.Find.Replacement.Style = ActiveDocument.Styles("Секции(подчеркнутый)")
Selection.Find.Replacement.Font.Bold = wdToggle
With Selection.Find
.Text = "Описание:"
.Replacement.Text = "Описание:"
.Forward = True
.Wrap = wdFindContinue
.Format = True
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Секции(подчеркнутый)")
Selection.Find.Replacement.ClearFormatting
Selection.Find.Replacement.Style = ActiveDocument.Styles("Секции(подчеркнутый)")
Selection.Find.Replacement.Font.Bold = wdToggle

```

With Selection.Find

```
.Text = "Параметры:"  
.Replacement.Text = "Параметры:"  
.Forward = True  
.Wrap = wdFindContinue  
.Format = True  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False
```

End With

Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting

Selection.Find.Style = ActiveDocument.Styles("Секции(подчеркнутый)")

Selection.Find.Replacement.ClearFormatting

Selection.Find.Replacement.Style = ActiveDocument.Styles("Секции(подчеркнутый)")

Selection.Find.Replacement.Font.Bold = wdToggle

With Selection.Find

```
.Text = "Возвращаемые значения:"  
.Replacement.Text = "Возвращаемые значения:"  
.Forward = True  
.Wrap = wdFindContinue  
.Format = True  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False
```

End With

Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting

```

Selection.Find.Style = ActiveDocument.Styles("Секции(подчеркнутый)")
Selection.Find.Replacement.ClearFormatting
Selection.Find.Replacement.Style = ActiveDocument.Styles("Секции(подчеркнутый)")
Selection.Find.Replacement.Font.Bold = wdToggle
With Selection.Find
    .Text = "Алгоритм работы:"
    .Replacement.Text = "Алгоритм работы:"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Краткое описание данной функциональной задачи^p"
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindContinue
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

```

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "Алгоритм выполнения данной функциональной задачи^p"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "^pПеречень функций, входящих в данную функциональную задачу^p"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute Replace:=wdReplaceAll

Selection.Find.ClearFormatting

Selection.Find.Style = ActiveDocument.Styles("Текст программ")

```

Selection.Find.Replacement.ClearFormatting
Selection.Find.Replacement.Style = ActiveDocument.Styles("Текст программ")
With Selection.Find
    .Text = " ("
    .Replacement.Text = vbTab + "("
    .Forward = True
    .Wrap = wdFindContinue
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

```

```

Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
Selection.Find.Replacement.Style = ActiveDocument.Styles("Обычный")
With Selection.Find
    .Text = "Подробное описание функций, входящих в данную функциональную задачу"
    .Replacement.Text = "Далее представлено подробное описание функций, входящих в
данную функциональную задачу.^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll

```

Selection.Find.ClearFormatting  
Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "Функциональные задачи"  
.Replacement.Text = ""  
.Forward = True  
.Wrap = wdFindContinue  
.Format = False  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False

End With

Selection.Find.Execute

Selection.MoveLeft Unit:=wdCharacter, Count:=1

Selection.MoveDown Unit:=wdLine, Count:=2

Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend

Selection.MoveRight Unit:=wdCharacter, Count:=1

Selection.MoveRight Unit:=wdCharacter, Count:=1

Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend

CountP = 1

Do While Selection.Paragraphs(1).Style = ActiveDocument.Styles("Текст программ")

Selection.MoveRight Unit:=wdCharacter, Count:=1

Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend

CountP = CountP + 1

Loop

Selection.MoveLeft Unit:=wdCharacter, Count:=1

ActiveDocument.Tables.Add Range:=Selection.Range, NumRows:=CountP - 1,  
NumColumns:= \_  
2, DefaultTableBehavior:=wdWord9TableBehavior, AutoFitBehavior:= \_  
wdAutoFitFixed

With Selection.Tables(1)

```

If .Style <> "Сетка таблицы" Then
    .Style = "Сетка таблицы"
End If

.ApplyStyleHeadingRows = True
.ApplyStyleLastRow = False
.ApplyStyleFirstColumn = True
.ApplyStyleLastColumn = False
.ApplyStyleRowBands = True
.ApplyStyleColumnBands = False
End With

Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Подзаголовок 1")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Функциональные задачи"
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.Find.Execute

Selection.MoveLeft Unit:=wdCharacter, Count:=1
Selection.MoveDown Unit:=wdLine, Count:=2
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
Selection.MoveRight Unit:=wdCharacter, Count:=1

```

```

Selection.MoveRight Unit:=wdCharacter, Count:=1
CountPp = CountP
Do While CountPp > 1
Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdExtend
CountPp = CountPp - 1
Loop
Selection.Cut
ActiveDocument.Tables(1).Select
Selection.Rows.AllowBreakAcrossPages = False
Selection.MoveLeft Unit:=wdCharacter, Count:=1
Selection.MoveDown Unit:=wdLine, Count:=CountP - 1, Extend:=wdExtend
Selection.MoveLeft Unit:=wdWord, Count:=1, Extend:=wdExtend
Selection.Paste
ActiveDocument.Tables(1).Select
With Selection.Tables(1)
    .Borders(wdBorderLeft).LineStyle = wdLineStyleNone
    .Borders(wdBorderRight).LineStyle = wdLineStyleNone
    .Borders(wdBorderTop).LineStyle = wdLineStyleNone
    .Borders(wdBorderBottom).LineStyle = wdLineStyleNone
    .Borders(wdBorderHorizontal).LineStyle = wdLineStyleNone
    .Borders(wdBorderVertical).LineStyle = wdLineStyleNone
    .Borders(wdBorderDiagonalDown).LineStyle = wdLineStyleNone
    .Borders(wdBorderDiagonalUp).LineStyle = wdLineStyleNone
    .Borders.Shadow = False
End With
With Options
    .DefaultBorderLineStyle = wdLineStyleSingle
    .DefaultBorderLineWidth = wdLineWidth150pt
    .DefaultBorderColor = wdColorAutomatic
End With
Selection.Style = ActiveDocument.Styles("Текст программ")
Selection.Font.Size = 10
Selection.MoveRight Unit:=wdCharacter, Count:=1

```

```

Selection.Find.ClearFormatting
Selection.Find.Style = ActiveDocument.Styles("Подзаголовок 1")
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "Функциональные задачи"
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute
Selection.Find.Execute
Selection.Delete Unit:=wdCharacter, Count:=2

Set y = CreateObject("Scripting.FileSystemObject").GetFolder(ActiveDocument.Path)
y = y.ParentFolder
y = y & "\rtf"
ChangeFileOpenDirectory _
    y
ActiveDocument.SaveAs2 FileName:="refman.doc", FileFormat:= _
    wdFormatDocument, LockComments:=False, Password:="", AddToRecentFiles:= _
    True, WritePassword:="", ReadOnlyRecommended:=False, EmbedTrueTypeFonts:= _
    False, SaveNativePictureFormat:=False, SaveFormsData:=False, _
    SaveAsAOCELetter:=False, CompatibilityMode:=0
End Sub

```

## ПРИЛОЖЕНИЕ Б

(обязательное)

### Протокол проверки ВКР на оригинальность

#### Протокол

проверки на оригинальность в системе «Антиплагиат.ВУЗ» выпускной квалификационной работы на тему «Методика восстановления алгоритма программ для оценки возможности появления опасных событий в телекоммуникационных системах специального назначения» студента 6 курса М.А. Цыбизова специальности 10.05.02 Информационная безопасность телекоммуникационных систем.

Руководитель ВКР Боровиков Алексей Юрьевич

#### Информация о документе:

Имя исходного файла: ВКР\_11ПТ1\_Цыбизов\_МА.doc

Имя компании: Пензенский гос. Университет

Тип документа: Прочее

Имя документа: Группа 11ПТ1 ВКР Цыбизов МА

Дата проверки: 14.01.2017 20:15

Модули поиска: Интернет (Антиплагиат), Университетская библиотека онлайн, Научные статьи Elibrary, Диссертации и авторефераты РГБ, Модуль поиска ЭБС "Лань", Модуль поиска ЭБС БиблиоРоссика, Пензенский гос. Университет, Цитирования, Кольцо вузов

#### Текстовые статистики:

Индекс читаемости: сложный

Неизвестные слова: в пределах нормы

Макс. длина слова: в пределах нормы

Большие слова: в пределах нормы

Оригинальные блоки: 72,56%

Заимствованные блоки: 27,21%

Заимствование из "белых" источников: 0,23%

Итоговая оценка оригинальности: **72,79%**

13.01.2017 \_\_\_\_\_ А.Ю. Боровиков