

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

09.04.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В ОБРАЗОВАНИИ И
ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЯХ

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему **РАЗРАБОТКА КОНЦЕПЦИИ СОПРОВОЖДЕНИЯ И
МОДЕРНИЗАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ,
СОЗДАННОГО НА ЯЗЫКЕ SOVOL**

Студент А.Е. Стрижов

Научный С.В. Мкртычев
руководитель

Руководитель программы д.пед.н., профессор, А.Н. Ярыгин
« » 20 г.

Допустить к защите

Заведующий кафедрой к. тех.н., доцент, А.В. Очеповский

« » 20 г.

Тольятти 2016

СОДЕРЖАНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ..... | 4 |
| ГЛАВА 1. АНАЛИЗ СОВРЕМЕННОГО СОПРОВОЖДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ СИСТЕМ ОТСЛЕЖИВАНИЯ ОШИБОК | 8 |
| 1.1 Применение языка программирования COBOL | 8 |
| 1.2 Описание программного комплекса Wizard..... | 10 |
| 1.3 Обзор существующих методов сопровождения программного продукта с помощью систем отслеживания ошибок..... | 14 |
| ГЛАВА 2. ОБОСНОВАНИЕ ВЫБРАННЫХ МЕТОДОВ СОПРОВОЖДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ..... | 27 |
| 2.1 Теоретические основы организации процесса сопровождения программного обеспечения с использованием системы отслеживания ошибок..... | 27 |
| 2.2 Основные положения концепции сопровождения программного обеспечения..... | 31 |
| 2.3 Описание процесса сопровождения..... | 33 |
| 2.4 Концептуальное моделирование процесса сопровождения | 37 |
| ГЛАВА 3. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА КОМПОНЕНТОВ СИСТЕМЫ ОТСЛЕЖИВАНИЯ ОШИБОК «CR CONTROL»..... | 43 |
| 3.1 Формализованное описание процесса сопровождения программного обеспечения в компании ООО «НетКрэкер»..... | 43 |
| 3.2 Архитектура системы отслеживания ошибок CR Control | 46 |
| 3.3 Проектирование базы данных системы CR Control | 47 |
| 3.4 Описание функционала системы отслеживания ошибок «CR Control» | 52 |
| 3.5 Реализация компонентов системы отслеживания ошибок | 57 |
| 3.6 Тестирование и испытание «CR Control»..... | 67 |

| | |
|--|----|
| ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНАЯ АПРОБАЦИЯ И ВНЕДРЕНИЕ КОНЦЕПЦИИ СОПРОВОЖДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ | 71 |
| 4.1 Внедрение модулей в систему CR Control | 71 |
| 4.2 Экспериментальная оценка качества сопровождения с использованием новых модулей системы отслеживания ошибок «CR Control» | 71 |
| ЗАКЛЮЧЕНИЕ | 84 |
| СПИСОК ЛИТЕРАТУРЫ | 86 |
| ПРИЛОЖЕНИЕ 1 | 92 |

ВВЕДЕНИЕ

В настоящее время серьезной проблемой является полное отсутствие или частичное использование систем отслеживания ошибок для осуществления полноценного процесса сопровождения программного обеспечения, написанного на устаревшем языке программирования COBOL. Это приводит к следующим негативным последствиям:

1) не осуществляется должным образом регистрация ошибок для пополнений базы знаний, необходимой для эффективного сопровождения;

2) используемые системы отслеживания ошибок не интегрированы непосредственно с приложениями на языке COBOL, что не позволяет им учитывать особенности данного языка.

Решением данных недостатков может послужить использование системы отслеживания ошибок, непосредственно взаимодействующей с самим языком COBOL и позволяющей отслеживать все изменения в программной коде без изменения документации сопровождаемого программного продукта.

Таким образом, **актуальность** темы исследования обусловлена тем, что программное обеспечение, разработанное на устаревшем языке программирования COBOL, которое по-прежнему используется некоторыми компаниями для поддержки сложных экономических задач, нуждается в разработке концепции по его сопровождению и модернизации.

Серьезный вклад в изучение процесса сопровождения программного продукта и программной инженерии внесли С. Орлик, В. Липаев, С. Орлов, В. Григорьев, Р. Гласс, В. Хан, Б. Уайт, Е. Палкин и др.

Проблематикой разработки приложений на языке COBOL занимались зарубежные специалисты Д.Маккракен, Р. Лемос, Н. Стерн, Н.Лоу, Р. Гилл, Т. Джексон и др.

Сложность и важность отмеченных проблем для теории и практики формирования эффективной политики компаний в области сопровождения программного обеспечения, а также поддержки программных продуктов,

написанных на устаревших языках программирования, требуют проведения серьезных методологических и методических исследований в области совершенствования концепций сопровождения специализированного программного обеспечения.

Целью диссертационной работы является разработка концепции сопровождения и модернизации программного обеспечения, созданного на языке COBOL.

Обозначенная цель определила объект и предмет исследования.

Объект исследования - процесс сопровождения и модернизации программного обеспечения, созданного на языке COBOL.

Предмет исследования – концепция сопровождения и модернизации программного обеспечения, написанного на языке COBOL.

Гипотеза: реализация предлагаемой концепции позволит создать и апробировать новую методику сопровождения и модернизации программного обеспечения, разработанного на языке COBOL.

Для достижения поставленной цели, необходимо решить следующие **задачи:**

1) исследовать используемые в настоящее время методы сопровождения и модернизации устаревшего программного обеспечения;

2) сформировать концепцию сопровождения и модернизации программного обеспечения, реализованного на языке программирования COBOL;

3) на основе предлагаемой концепции разработать и реализовать новую методику сопровождения и модернизации программного обеспечения, реализованного на языке программирования COBOL.

В работе использованы законодательные и нормативные акты, официальные документы различных государственных и правительственных органов, регулирующие корпоративные отношения на территории Российской Федерации.

Методы исследования: системный анализ, методы верификации и аттестации программного сопровождения, современные подходы к сопровождению и модернизации программного обеспечения.

Основные этапы исследования: исследование велось с 2014 по 2016 гг. в три этапа.

На *первом* – *констатирующем этапе* исследования (2014 г.) – определялась актуальность темы исследования, прорабатывалась литература по избранной теме, уточнялся аппарат и программа исследования, формулировалась гипотеза, определялись цели, задачи, предмет, объект и методы исследования, изучались проблемы использования существующих систем отслеживания ошибок в процессе сопровождения программного обеспечения на языке COBOL.

Второй этап (2014 - 2015 гг.) – *поисковый*. Здесь формировались концептуальные основы работы отдела технической поддержки, уточнялись цели, задачи и гипотеза исследования, осуществлялась апробация теоретических подходов в выступлениях и публикациях.

На *третьем* этапе (2015 - 2016 гг.) – осуществлялся *эксперимент*. Проводилась апробация разработанного модуля системы отслеживания ошибок, проверялась обоснованность и достоверность сформулированной гипотезы, обрабатывались и анализировались результаты эксперимента, обобщались итоги экспериментальной работы.

Апробация Основные положения диссертационной работы были апробированы в компании ООО «НетКрэкер», где проходила преддипломная практика с 28.03.2016 по 08.05.2016 г. После внедрения модуля была улучшена эффективность работы специалистов отдела технической поддержки.

В период исследования были опубликованы следующие статьи:

1. *Стрижов, А.Е. Оптимизация процесса сопровождения программного обеспечения, написанного на языке Cobol/ Студенческие дни науки в ТГУ. – Издательство ТГУ, 2015.*

2. *Стрижов, А.Е. Совмещение обязанностей ИТ-специалистов для осуществления полноценного сопровождения программного обеспечения // Научный альманах. – 2016. – С.175-178.*

Научная новизна исследования состоит в разработке новой концепции сопровождения и модернизации программного обеспечения, реализованного на языке COBOL.

Практическая значимость исследования заключается в улучшении эффективности работы отдела технической поддержки, занимающегося сопровождением и модернизацией программного обеспечения, реализованного на языке COBOL.

На защиту выносятся:

1. Концепция сопровождения и модернизации программного обеспечения, реализованного на языке COBOL.

2. Методика сопровождения и модернизации программного обеспечения, реализованного на языке COBOL, разработанная на основе предлагаемой концепции.

3. Информационно-технологическое обеспечение службы технической поддержки, обеспечивающее повышение эффективности сопровождения и модернизации программного обеспечения, реализованного на языке COBOL.

Работа представляет собой результат теоретической и практической деятельности в области сопровождения программного продукта, а также проектирования и реализации информационной системы отслеживания ошибок.

В структуру работы входят: введение, 4 главы, заключение, список литературы и приложение.

ГЛАВА 1. АНАЛИЗ СОВРЕМЕННОГО СОПРОВОЖДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ СИСТЕМ ОТСЛЕЖИВАНИЯ ОШИБОК

1.1 Применение языка программирования COBOL

COBOL (COmmon Business Oriented Language) был первым языком, в котором средства описания данных соответствуют процедурным возможностям, и первым языком, в котором введен тип данных «запись», являющийся основной структурой данных. К одной из примечательных особенностей COBOL относится рекурсивное описание данных. Другая его особенность в том, что программы на COBOL разбиваются на части, называемые разделами, причем каждая программа состоит из четырех разделов: идентификации, оборудования (среды), данных и процедур.

COBOL обеспечивает действительно эффективную работу с простыми файловыми структурами и операциями над крупными наборами данных. Точность цифровых операций его такова, что достичь ее с помощью других языков непросто до сих пор. Например, он поддерживает выполнение математических операций до 38-го знака. Кроме того, его легко изучить. Это один из самых простых языков для понимания программирования [59].

Согласно исследованию Gartner (исследовательская и консалтинговая компания) за 2008 год (для этого языка эти данные можно считать актуальными), 85% мирового объема бизнес-информации обрабатывалось на языке программирования COBOL, а в настоящее время, объем кода, написанного на этом языке программирования, только увеличивается [47]. Компания Micro Focus на данный момент активно занимающаяся развитием инструментов разработки и модернизации COBOL, сообщает о том, что около 70% мирового бизнеса активно использует этот язык [46]. Большинство существующих приложений будут использоваться еще долгое время, хотя часть их будет переписана на новые языки программирования, но общая бизнес-

логика меняться не будет, так как эти приложения безопасны и стабильны, и использующим их компаниям, нет смысла отказываться перестраиваться.

В качестве примера можно привести бизнес-процесс перевода денежных средств с одного банковского счета, на другой, потому что подобная процедура не изменилась с момента ее создания. Но взаимодействие клиентов со своими банковскими счетами и транзакциями, существенно, если не кардинально, изменилось. На сегодняшний день, большинство клиентов хочет иметь доступ к банковскому счету через персональный компьютер или смартфон. В свою очередь, компаниям, использующим приложения, написанные на COBOL, самым трудным моментом будет сохранение и передача накопленного опыта и знаний о существующих бизнес-процессах новым разработчикам. Те программисты, которые уже знают этот язык и понимают то, как он интегрируется в современные технологии будут выгодно выделяться на фоне своих коллег, даже несмотря на то, что в принципе это не самая сложная, для изучения, технология.

В настоящее время COBOL поддерживает 90% бизнес-систем, используемых корпорациями из списка Fortune 500 (рейтинг крупнейших мировых компаний, критерием составления которого служит выручка компании) и используется в 85% всех ежедневных финансовых (бизнес) транзакций [44,51]. В США по-прежнему пишутся программы на COBOL и даже есть ощутимый спрос на таких программистов (в основном это поддержание огромного количества уже написанного).

В свою очередь, для компаний, использующих приложения, написанные на COBOL, самым трудным моментом будет сохранение и передача накопленного опыта и знаний о существующих бизнес-процессах новым разработчикам. Те программисты, которые уже знают этот язык и понимают то, как он интегрируется в современные технологии будут выгодно выделяться на фоне своих коллег, даже несмотря на то, что в принципе это не самая сложная, для изучения, технология.

Компания Micro Focus провела существенную работу по адаптации этого языка программирования к современным компьютерным средам разработки, в связи с этим, он вполне может использоваться для современных проектов, которые имеют свои особенности.

1.2 Описание программного комплекса Wizard

В рамках данной научной работы рассматривается сопровождение программного обеспечения (ПО), а именно программного комплекса Wizard, который представляет собой комплексное, пре-интегрированное решение по CRM (управление отношениями с клиентами) и биллингу (процесс определения стоимости услуг) для ведущих кабельных и спутниковых компаний, предлагающих следующие услуги:

- 1) телевидение;
- 2) телефонная связь;
- 3) Интернет.

Сам комплекс Wizard написан непосредственно на языке программирования COBOL. Особенностью системы Wizard является отсутствие рейтинга, то есть все, что связано с использованием продукта, которым пользуется клиент. В дальнейшем продукт был интегрирован с Geneva – системой, поддерживающей рейтинговую систему, которая в настоящее время больше известна как RBM.

Wizard является закрытой системой и, как следствие, только авторизованные пользователи могут производить изменения конфигурации и менять значения системных параметров. Стоит отметить, что на начальных этапах не была реализована интеграция с другими системами и это обусловлено фактом того, что на тот момент такой необходимости не было. В связи с этим отсутствовал API (интерфейс создания приложений), обеспечивающий не только взаимодействие Wizard с другими системами, но и предоставляющий возможности для разработчиков создавать интерфейсы для новых интеграций.

В дальнейшем необходимость интеграции внешних систем с системой Wizard стала актуальной для ряда заказчиков, в результате чего в зависимости от потребностей каждого из них была инициирована разработка соответствующих API. На рис. 1.1 представлены функциональные части Wizard системы:



Рисунок 1.1 – Программный комплекс WIZARD

В настоящий момент в рамках Wizard-а реализована интеграция, в том числе на базе Java API, которая, в свою очередь, взаимодействует с продуктом на основе RMI интерфейса.

Детальное описание продуктовых компонентов WIZARD представлено ниже:

1) Системный компонент CRM используется для управления взаимодействием между Customer Service Representatives (CSRs) и заказчиком или потенциальным клиентом. Он предоставляет доступ к информации о заказчиках, потенциальных клиентах, биллинге, адресах и оборудовании, а так же возможность выполнять множество подобных функций.

2) Work Order – это набор системных действий для выполнения таких функций, как создание нового аккаунта, смена существующего оборудования или продукта клиента, решение вопросов, касательно жалоб клиента и отсоединение клиента

3) Системный компонент Accounts Receivable(AR) предоставляет доступ к финансовой информации, с возможностью ее анализа и управления. Системный компонент Accounts Receivable взаимодействует с компонентом CRM и предоставляет возможность отображать и выполнять задачи касательно биллинга и финансового пакета данных.

4) Системный компонент Codes and Parameters(C&P) предоставляет возможность определить многие аспекты функциональности WIZARD. Это возможно с помощью определения кодов, параметров и дает возможность установить то, как будет работать система.

5) Системный компонент Inventory Tracking позволяет выполнять действия, касающиеся инвенторного оборудования, такие как добавление, изменение, удаление индивидуальных сущностей или групп сущностей.

6) Системный компонент Addressability включает несколько процессов, которые происходят на Unix сервере. Эти транзакции по передаче данных с WIZARD происходят в соответствии с используемым Access Control Center (ACC)/CAS протоколом.

7) Системный компонент Telephone Number Management (TNM) позволяет администрировать все функции Telephone Number (TN). Системный компонент Telephone Number Management (TNM) предоставляет возможность создания, поддержки и обновления телефонных номеров в системе. Телефонные номера идентифицируются различными атрибутами, среди которых системный статус, тип номера и метод, с помощью которого номера сгруппированы.

8) Telephony Number (TN) Codes (Коды телефонных номеров) – это отдельное приложение установки, с помощью которого возможно добавлять

только коды, которые относятся к телефонии. Существует необходимость установить все необходимые коды перед использованием системного компонента Telephony Number Management (TNM).

9) Системный компонент Pay Per View позволяет абонентам кабельных и спутниковых компаний заказывать (оплачивать) индивидуальные события и программы, такие как фильмы, спортивные события, используя разнообразные методы заказа. Использование компонента PPV возможно с помощью применения доступного оборудования, которое позволяет продавать абонентам отдельные программы и фильмы единожды, на основании разового платежа (чаще за программу, чем за месяц).

10) Системный компонент Billing and Statements позволяет вам производить оценку, закрывать биллинговый месяц, а так же формировать и печатать отчеты.

11) Системный компонент Marketing and Sales предоставляет доступ к информации и возможность проводить управление и анализ информации по маркетингу и продажам.

12) Collections – это системный процесс решения невыполненных или просроченных платежей. После того, как функции WIZARD были определены, большинство из них запускаются автоматически и появляются в фоновом режиме без вмешательства оператора.

Wizard имеет клиент-серверную архитектуру. Пользователь подключается со своего персонального компьютера к серверу посредством Citrix XenApp.

Citrix XenApp обеспечивает удаленный доступ к приложениям и данным Windows по любой сети и на любом устройстве, а также позволяет сотрудникам получать доступ к виртуальным десктопам Linux одновременно с ресурсами Windows, что обеспечивает полноценную работу. Для обработки основного потока данных используются Unix-сервера. Для связи клиента с сервером используется SQL*Net протокол.

1.3 Обзор существующих методов сопровождения программного продукта с помощью систем отслеживания ошибок

Существует огромное количество программ и каждая из них имеет свои особенности, преимущества и недостатки, при этом даже хорошо оттестированная программа может иметь сбои.

Всё это происходит потому, что одна программа «живет» в окружении множества других программ, написанных совсем другими людьми, не учитывая совместимость с операционной системой и аппаратными средствами. Необходимо учитывать, что многие компании, занимающиеся разработкой программного обеспечения, зачастую уменьшают цикл производства, не заботясь о полноценном тестировании, ввиду этого, разработчикам приходится постоянно взаимодействовать со службой поддержки.

Работники службы поддержки принимают от пользователей множество вопросов, заявлений об ошибках и дефектах, регистрируют их в системах отслеживания ошибок и передают разработчикам. В свою очередь, в компаниях проводящих тщательное тестирование перед запуском программного продукта, разработчикам приходится рассматривать отчеты об ошибках и дефектах, непосредственно от команды тестировщиков.

Регистрация и обработка данных об ошибках и дефектах, может показаться на первый взгляд просто задачей, хотя таковой она не является. Все дело в том, что еще до выдачи сам программист должен провести полноценный тест своей программы [60].

Для контроля ошибок и дефектов был создан замечательный продукт - система отслеживания ошибок.

Система отслеживания ошибок BTS (англ. bug tracking system) - прикладная программа, позволяющая разработчикам ПО (программистам, тестировщикам и др.) учитывать и контролировать ошибки (баги), найденные в программах, а также следить за процессом устранения этих ошибок [52].

В настоящее время существует огромное количество компаний занимающихся разработкой подобных систем и их распространением, например: Netscape, JetBrains, Edgwall Software, ГРАН, Atlassian и др.

Значимость подобных систем в процессе сопровождения программ рассматривали такие авторы, как Г. Лиля, М. Фишер, Н.Серрано и др. Важность точности и корректности работы систем отслеживания ошибок для повышения эффективности сопровождения рассматривали такие авторы как Т. Зиммерманн, Р. Премраж, Дж. Силито и С. Брэу.

Для анализа таких систем были определены критерии, по которым системы будут оценены:

- язык: в системе много пользователей, из них некоторые заказчики, и в любом случае далеко не все знают английский, поэтому очень желательно, чтобы система имела перевод;
- интеграция с системой контроля версий (Git или SVN);
- удобство использования: желательно, чтобы было сразу понятно, как пользоваться системой, и у заказчиков и разработчиков не возникало сотни вопросов;
- настраиваемые поля;
- фильтрация: поиск и фильтрация должны быть удобными;
- перевод: если русский язык есть, то он должен быть качественным;
- права доступа: любые настройки ролей для пользователей и групп;
- локальная установка: установка системы на свой сервер;
- экспорт: форматы, в которых можно экспортировать данные;
- цена: стоимость системы.

В результате исследования специальной литературы и известных ИТ-решений для сравнения были выбраны рассмотренные ниже системы отслеживания ошибок.

Система Assembla. Assembla (рис. 1.2) представляет собой набор облачных задач и инструментов управления кодами для разработчиков ПО.

Продукт Assembla принадлежит компании Assembla, INC и был создан в 2005 г. Она поддерживает более 100 000 коммерческих проектов и ее используют более чем 800 000 пользователей в более чем 100 странах мира.

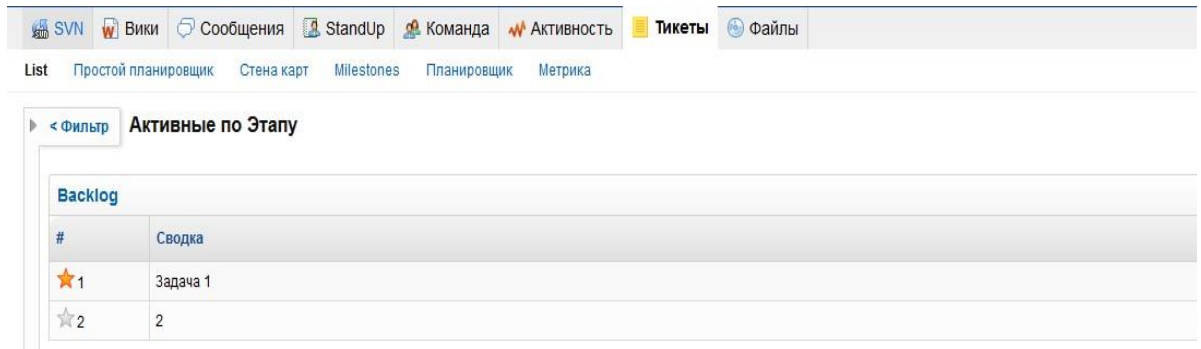


Рисунок 1.2 – Интерфейс системы Assembla

Плюсов у системы достаточно много. Очень хорошая служба поддержки: в 19:30 по Москве задан вопрос по возможностям программы, и ответ получен в течение пяти минут. Очень качественная статистика: можно проследивать любые действия разработчика и видеть, что и когда он делал. Все изменения статусов, открытие/закрытие задач и внесение изменений отображаются в статистике. Можно прямо в системе заполнять ежедневные отчеты. Хорошо реализован поиск.

Один из главных минусов программы — перевод. Он сделан некачественно, и переведено далеко не все. В программе нужно долго разбираться. Вряд ли получится понять что-то, зайдя в нее первый раз. Задачи нельзя переносить между проектами, установить баг-трекер на свой сервер нельзя, что неудобно для больших компаний. Стоимость системы составляет 1188 долларов в год за 50 человек.

Система Trello. Система отслеживания ошибок Trello (рис. 1.3) - это бесплатное веб-приложение для управления проектами небольших групп, разработанное Fog Creek Software. К 2014 года системой пользуются более миллиона пользователей.

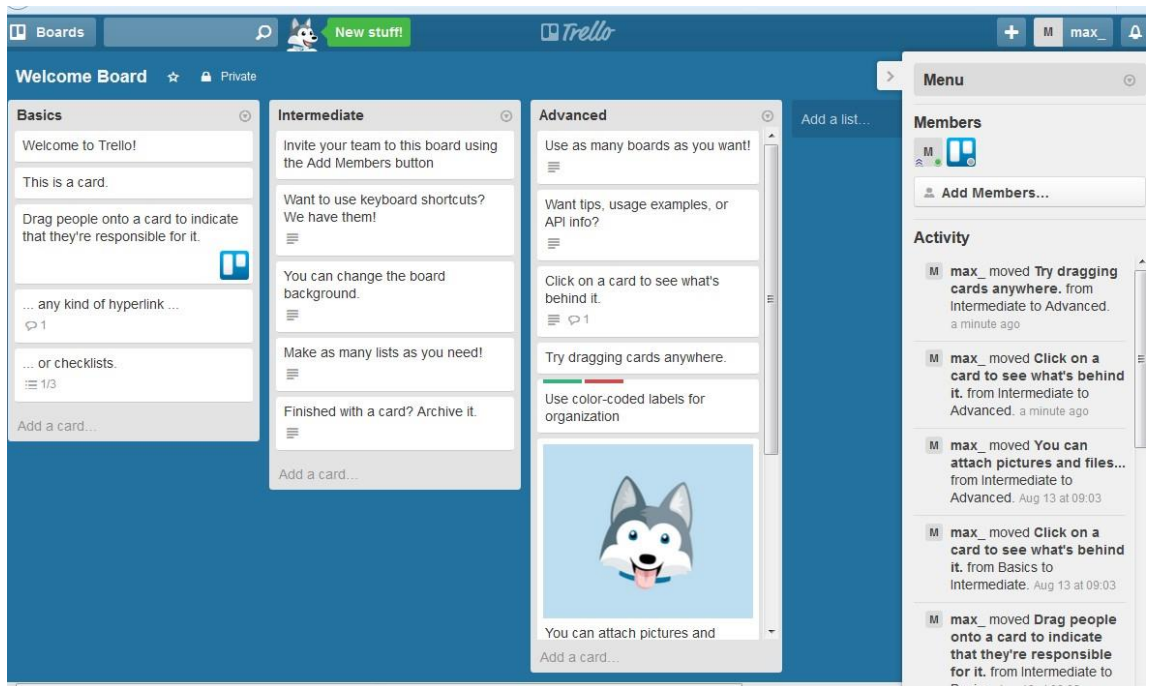


Рисунок 1.3 – Интерфейс системы Trello

Система удобна тем, что она вся построена на основе доски и все, что в ней есть, находится на одном экране: и задачи, и история изменений, и любые комментарии. Но это же и главный минус программы. Она слишком простая и не предназначена для больших команд. Задачи банально не имеют номера, не говоря уже о переводе или установке на свой сервер. Конечно, такой подход удобен для заказчиков, которые могут увидеть все задачи в одном месте, не делая сложных поисков и не разбираясь в системе, но, как только количество открытых задач приблизится к пятистам, это, даже для заказчиков, из плюса превратится в минус.

Баг-трекер недорогой и подойти может только небольшим командам, у которых мало задач. В таком случае система будет справляться со своими обязанностями и помогать разработчикам следить за выполнением задач. Стоимость системы составляет 600 долларов в год за 50 человек.

Система YouTrack. YouTrack - онлайн баг-трекер и система управления задачами и проектами. YouTrack пользуется популярностью среди разработчиков из-за развитого использования клавиатурных шорткатов (коротких путей), эффективных поисковых запросов и командного синтаксиса,

а также полностью настраиваемых рабочих процессов и атрибутов задач (рисунки 1.4,1.5).

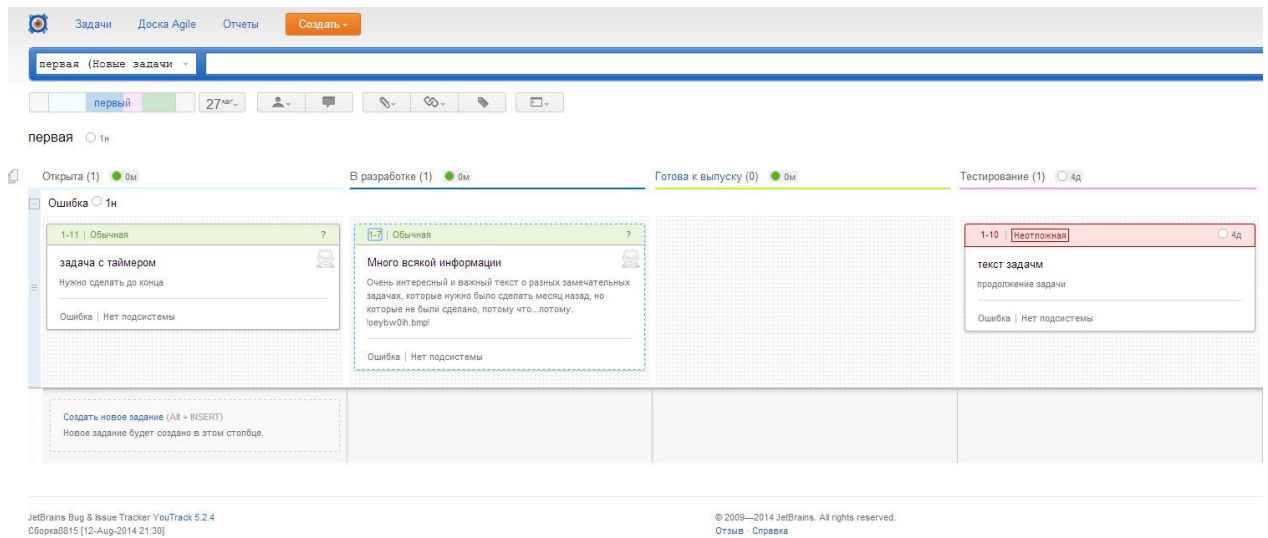


Рисунок 1.4 – Интерфейс системы YouTrack

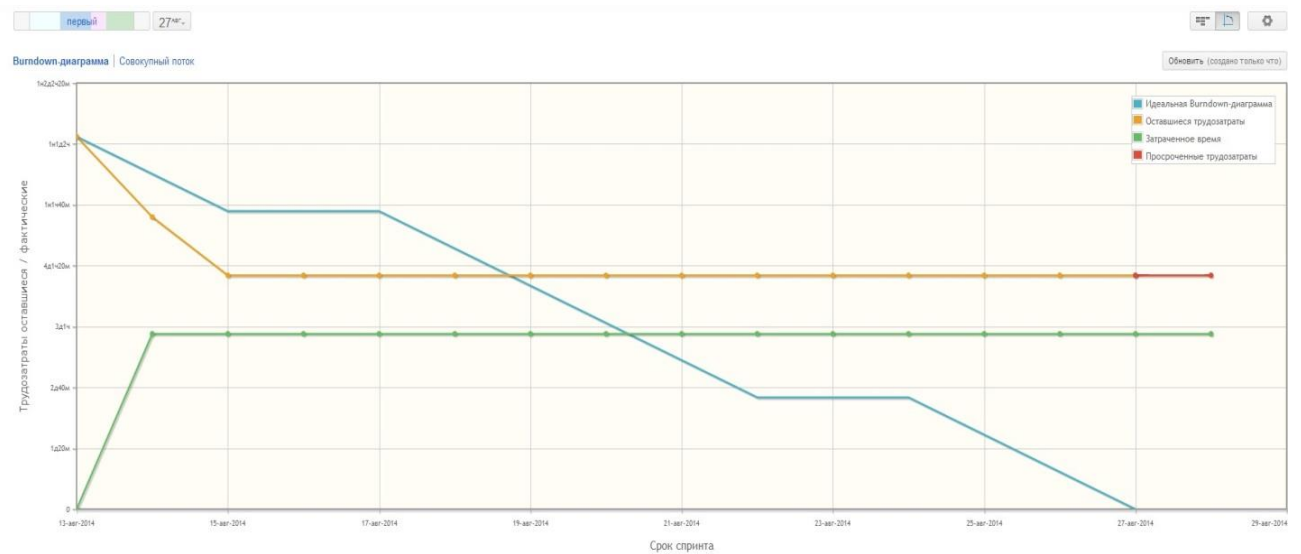


Рисунок 1.5 – Интерфейс системы YouTrack

Плюсов у YouTrack значительно больше, чем минусов. Не очень удобной кажется нумерация, так как номер задачи меняется при переносе ее в другой проект, поиск и подписка на обновления вообще требуют отдельной статьи — описать их в несколько строчек невозможно. Но с этими проблемами разобраться несложно, и сделать это должен только администратор баг-трекера, у пользователей этих проблем при правильной настройке не возникнет. Доска, как и диаграмма, очень удобные, статусы и приоритеты можно выделять так,

чтобы сразу было видно то, что нужно ярко выделить, все удобно настраивается. Что также очень полезно — есть скрипт для импорта задач из большинства баг-трекеров, что сильно облегчает работу менеджера при переходе (хоть импорт и имеет свои минусы). Также минусом является отсутствие техподдержки. Если программа стоит 750 долларов, то можно как-то выделить человека, который будет отвечать на вопросы, а из помощи существует только англоязычный форум и собственно YouTrack, в котором можно создавать задачи по системе и писать о багах. Стоимость системы составляет 750 долларов в год за 50 человек.

Система JIRA Agile. JIRA - это инструмент для планирования, управления, отслеживания и просмотра результатов работы команды. Agile — это концептуальный каркас, в рамках которого выполняется разработка ПО.

JIRA Software используется для управления agile-проектами более чем 3800 организациями в 80 странах мира. Это крупнейшие корпорации, правительственный, образовательный, финансовый и промышленный сектор.

Для тестирования была выбрана не обычная версия JIRA, а JIRA Agile, имеющая доску и диаграмму для Scrum. Эта версия программы стоит дороже обычной. Баг-трекер не переведен на русский язык (рисунки 1.6, 1.7).

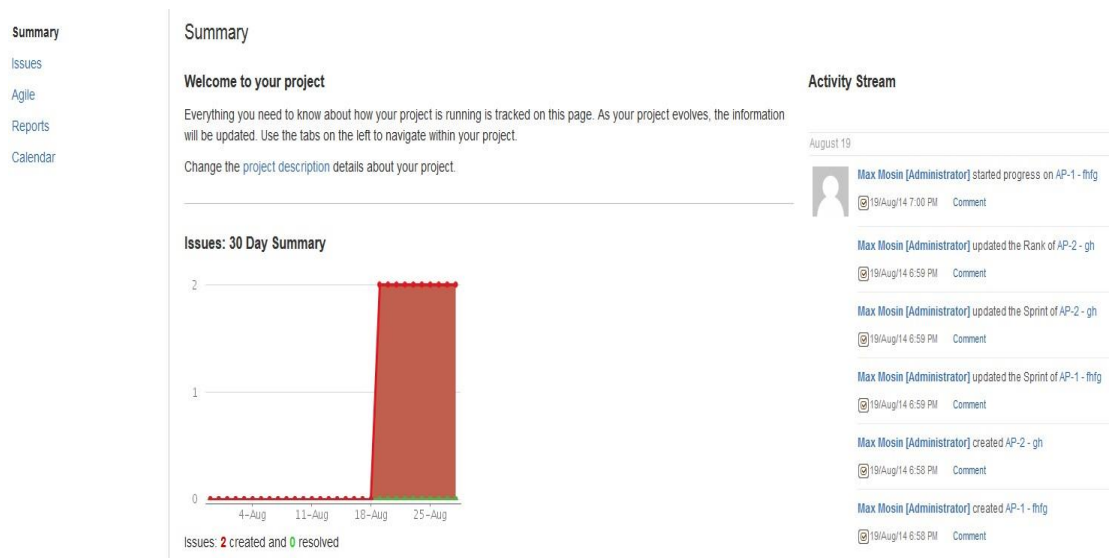


Рисунок 1.6 – Интерфейс системы JIRA

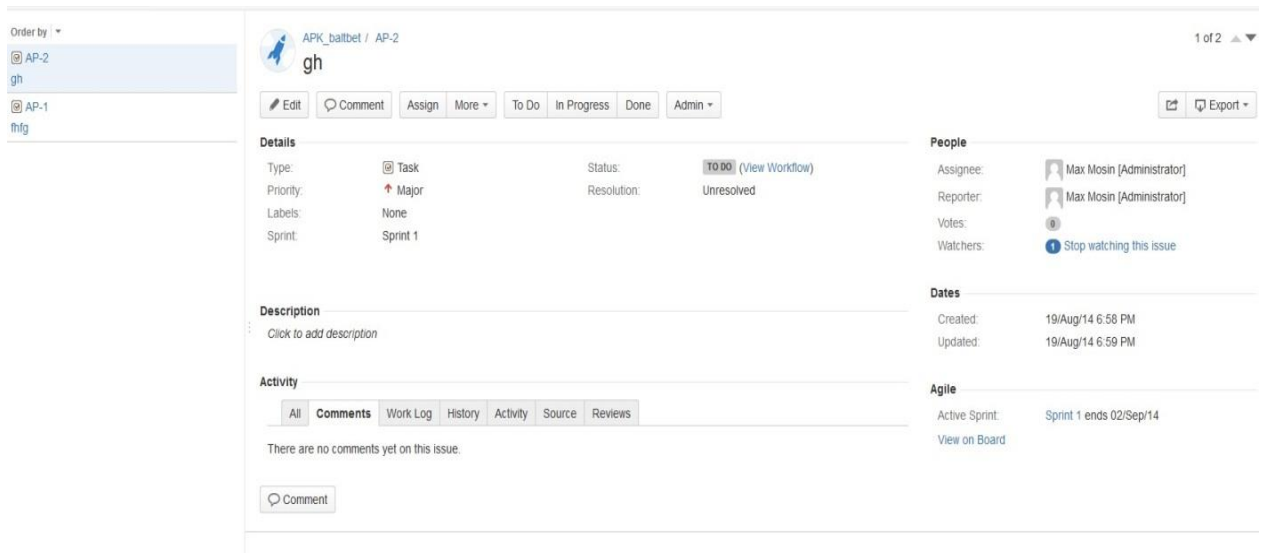


Рисунок 1.7 – Интерфейс системы JIRA

В системе очень много возможностей для настройки. Может быть, даже слишком много. По каждому проекту можно посмотреть подробную статистику, доска хорошая, но кажется менее удобной, чем в YouTrack. Что действительно удобно - сами задачи и список задач находятся в одном окне, то есть для переключения между задачами достаточно одного клика. В Assembla и YouTrack это реализовано хуже.

Можно сделать вывод: система сделана программистами для программистов, что имеет свои плюсы и минусы. К примеру, заказчик будет разбираться с системой долго, и вопросов возникнет очень много. Стоимость системы составляет 1100 долларов в год за 50 человек.

Система Targetprocess. Targetprocess – это онлайн баг трекер, который можно использовать для планирования и отслеживания всех процессов, в том числе с помощью досок, списков и сроков. TargetProcess используется как небольшими командами, так и крупными предприятиями (рис. 1.8)

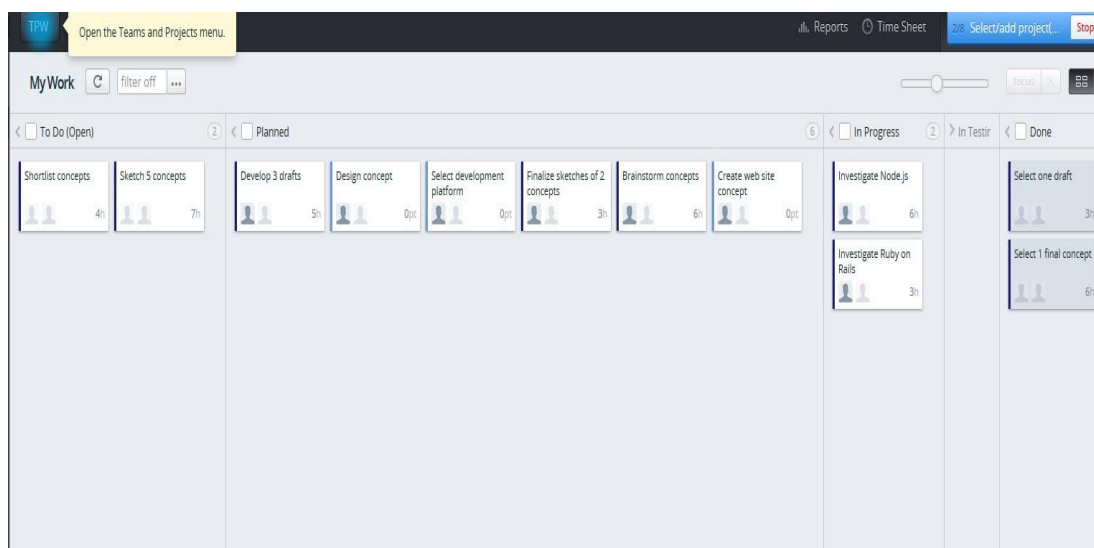


Рисунок 1.8 – Интерфейс системы Targetprocess

Система очень дорогая, и тяжело понять, почему она так дорого стоит. Доска удобная, а размещение задач сделано по такому же типу, как в Trello, только более приспособлено к большим командам и большому количеству задач. Диаграмма тоже хорошая, ничего лишнего, и все понятно. Возможности по настройке полей хуже, чем у других систем. Видно, что баг-трекер хотели сделать как можно более простым и удобным, но вместо того, чтобы сделать всю функциональность как можно более простой, похоже, решили просто ее обрезать. Конечно, удобно заходить и видеть все задачи, легко переключаться между досками проектов и менять статусы, но, как только требуется сделать какие-то более сложные действия, возникают проблемы. На русский язык Targetprocess также не переведен. Стоимость системы составляет 7000 долларов в год за 50 человек.

Система CR Control. CR Control представляет собой полноценную систему баг-трекинга со своими особенностями. Данная система была разработана компанией Convergys более 15 лет назад и продолжает использоваться на сегодняшний день в компании «НетКрэкер», в качестве системы отслеживания ошибок для осуществления сопровождения программного комплекса Wizard, написанного на языке программирования COBOL. В мае 2012 года компания «НетКрэкер», приобрела подразделение

Information Management компании Convergy. Богатый опыт Convergy в области разработки и внедрения бизнес-решений значительно укрепили позиции на рынке ПО для управления клиентами и доходами и пополнил ряд имеющегося ПО, включая Wizard и CR Control.

Система CR Control реализована как web-приложение, имеющая интерфейс полностью на английском языке, т.к. предназначена для поддержки программного продукта Wizard, который в свою очередь используют в разных странах мира, таких как Польша, Венгрия, Израиль. Система работает во внутренней сети компании «НетКрэкер» и, помимо, стандартной авторизации, системе реализована настройка прав доступа пользователям, т.е. конкретному пользователю может быть доступна только определенная часть функционала системы.

В системе настроен прямой доступ к базе данных Wizard из web-интерфейса, также осуществляется контроль версий файлов программы на COBOL, что сравнимо с наличием у других систем Git и SVN, присутствуют различные сборы статистик за промежутки времени (рис. 1.9)

The screenshot displays the 'Create New CR' form in the CR Control system. At the top, there is a navigation bar with links: 'Create CR', 'Update CR', 'Show CR', 'Reports', and 'Maintenance'. Below this, the page title is 'WIZARD 7-9' and a 'Save' button is visible in the top right corner. The main form area is titled 'Create New CR' and contains several sections:

- Title:** A text input field.
- Topic:** A dropdown menu.
- CR Type:** A dropdown menu.
- Development Stage:** A dropdown menu.
- Severity:** A dropdown menu.
- Customer Priority:** A dropdown menu.
- Estimated Days To Accomplish:** A text input field.
- Effort:** A text input field.
- Hours:** A text input field.
- Problem Type:** A group of checkboxes including 'Code', 'Basic Customization', 'Database', 'Help', 'User Interface', 'New Parameter', 'Performance', 'New Privilege', and 'Script'. The 'Code' checkbox is checked.
- Project:** A dropdown menu with the value 'WIZARD76'.
- Release:** A dropdown menu with the value '7.6.8'.
- Customer/Site:** A dropdown menu with the value 'hot'.
- From Customer:** Radio buttons for 'Yes' and 'No', with 'No' selected.
- Reported by:** A text input field.
- Site Ref#:** A text input field.
- Original release/version/item:** A text input field.
- Patch Number:** A text input field.
- Rework:** Radio buttons for 'Yes' and 'No', with 'No' selected.
- Code Change:** Radio buttons for 'Yes' and 'No', with 'Yes' selected.
- Core/Non Core:** Radio buttons for 'Core' and 'Non Core', with 'Core' selected.
- Scenario/Setup:** A large text area with a vertical scrollbar.
- Problem Description:** A large text area with a vertical scrollbar.
- Solution/Recommendation:** A large text area with a vertical scrollbar.
- Booklet:** A dropdown menu.
- SA#:** A dropdown menu.
- Release Notes:** Radio buttons for 'Y' and 'N', with 'Y' selected.
- On Site Env:** A text input field.

Рисунок 1.9 – Интерфейс системы CR Control

Главной особенностью CR Control является интеграция с языком программирования COBOL, под которой подразумевается простейшая проверка программного кода на наличие структурных ошибок, если файл не был скомпилирован в среде разработки, также проверяется использование процедур и функций, что позволяет избежать путаницы в последующей работе над файлом другими разработчиками.

Для полного представления о том, какая должна быть система отслеживания ошибок, необходимо изучить программный продукт, сопровождением которого необходимо заниматься.

В рамках исследования был рассмотрен программный продукт Wizard, разработанный компанией Convergys в 1992 году, всё ядро данного продукта написано на языке программирования COBOL. Конкретно, для сопровождения данного продукта и происходит анализ существующих систем баг-трекинга. Для полноценного сопровождения Wizard, необходимо учитывать все его особенности и, в частности, особенности языка COBOL.

Важнейшим показателем приведенных систем является интеграция с системой контроля версий, которые позволяют вести контроль версий файлов программного продукта. В случае с программой на COBOL, это является важнейшим составляющим, т.к. подобные программы имеют сотни тысяч строк программного кода, за которыми нужен постоянный контроль.

Таблица 1.1 - Сравнительный анализ систем отслеживания ошибок

| Критерий сравнения | Assembla | Trello | YouTrack | JIRA Agile | Target process | CR Control |
|---------------------------------------|----------|--------|----------|------------|----------------|------------|
| Интеграция с системой контроля версий | + | - | + | + | - | + |
| Удобство | + | + | + | - | - | + |
| Настраиваемые поля | + | + | + | + | + | + |

Продолжение таблицы 1.1

| Критерий сравнения | Assembla | Trello | YouTrack | JIRA Agile | Target process | CR Control |
|-----------------------------|----------|--------|----------|------------|----------------|------------|
| Фильтрация | + | - | + | + | - | + |
| Настройка прав доступа | + | - | + | + | + | + |
| Наличие экспорта (CSV, XML) | + | - | + | + | + | + |
| Web-приложение | + | + | + | + | + | + |
| Английский язык | - | + | - | + | + | + |
| Отчеты | + | + | + | + | + | - |
| Цена | - | + | + | - | - | + |
| Итого | 8 | 6 | 9 | 8 | 6 | 9 |

В ходе выбора баг-трекера было изучено шесть программ. Но учитывая, что система CR Control в данный момент используется для сопровождения программного комплекса написанного на COBOL, выбор будет проходить среди оставшихся пяти систем, для выявления наиболее актуального конкурента для CR Control.

Система Trello оказалась непригодной для больших компаний и большого количества задач, система красиво выглядит и может подойти небольшим командам, но не более того. Четвертое место занял Targetprocess: система похожа на Trello, но более приспособлена для работы с большими объемами задач. Доска сделана качественно и просто, но, когда дело доходит до более тонкой настройки, появляется много сложностей и деталей, с которыми Targetprocess не справляется. Широких возможностей по настройке система не дает, а стоимость больше, чем остальных систем, что кажется необоснованным.

Популярная система JIRA с плагином Agile заняла третье место. В ней очень много возможностей для настройки, хорошая статистика по проектам, удобно смотреть задачи и неплохо реализована доска. Но разобраться в системе достаточно сложно, заказчикам изучение этого баг-трекера будет стоить немало труда, и администратору придется постоянно отвечать на возникающие вопросы. Если баг-трекером будут пользоваться только программисты, ее можно рассматривать как реальный вариант, но в компании, где половину пользователей составят заказчики, система должна быть такой, чтобы при первом входе было хотя бы приблизительно понятно, что нажимать, чтобы найти нужные задачи или создать новую.

Также, выделяются две системы: Assembla и YouTrack. Они очень отличаются друг от друга, что сильно усложнило выбор. Assembla ведет прекрасную статистику каждого пользователя, по ней можно изучать работу программистов и оценивать ее, видно все коммиты и к каким задачам они относятся. В YouTrack такого нет. Время, потраченное разработчиками на задачи, отслеживать можно, но для этого нужно заставить их писать это время в каждой задаче, что осложняет работу. Также Assembla не требует сторонних программ для интеграции с системой контроля версий. YouTrack здесь отстает несильно, для него есть бесплатное приложение TeamCity, которое предоставляют разработчики, но с ним нужно будет дополнительно разбираться. Также в Assembla очень удобно следить за отчетами разработчиков, чего вообще нет в YouTrack.

Выбирая между этими двумя системами, нужно решить, что важнее - возможность контролировать разработчиков и вести статистику их работы или, настроив систему самостоятельно и потратив на нее немало времени, получить простой в использовании и наглядный баг-трекер. Был выбран второй вариант, и первое место в обзоре из пяти систем управления проектами занял YouTrack.

Исходя из таблицы 1.1 видно, что лидирует система YouTrack наравне с CR Control. Сравнивая YouTrack с CR Control, не трудно заметить, что в нем

больше всевозможных статистик и способов отслеживания работы программистов. Но в тоже время он является достаточно сложным в использовании и требует больших финансовых вложений, учитывая, что в сопровождении Wizard задействовано более 50 человек, в отличие от системы CR Control, которая уже используется компанией и не требует дополнительных затрат на каждого сотрудника. Также, в YouTrack, как и в большинстве баг-трекеров, отсутствует какая-либо интеграция с программным кодом, в отличие от CR Control.

Из выше изложенного можно сделать вывод, что для осуществления сопровождения нового проекта наиболее подойдет YouTrack, в котором реализованы все необходимые функции системы баг-трекинга. Но в конкретно рассматриваемом случае, где на первый план выходит интеграция с языком программирования и фактор внедрения системы, целесообразно использование системы отслеживания ошибок CR Control.

Вывод по главе

1) По данным некоторых зарубежных аналитических и консалтинговых компаний приложения, написанные на языке COBOL, по-прежнему пользуются популярностью в сфере поддержки задач обработки бизнес-информации.

2) Сравнительный анализ систем отслеживания ошибок показал, что для осуществления сопровождения новых проектов на языке COBOL наиболее подходит система CR Control.

ГЛАВА 2. ОБОСНОВАНИЕ ВЫБРАННЫХ МЕТОДОВ СОПРОВОЖДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Теоретические основы организации процесса сопровождения программного обеспечения с использованием системы отслеживания ошибок

Сопровождение ПО, разработанного на языке COBOL (далее – сопровождение ПО), определяется как вся совокупность деятельности, необходимой для обеспечения эффективной (с точки зрения затрат) поддержки программных систем, разработанных на этом языке. Эти работы выполняются как перед вводом системы в эксплуатацию, так и после этого.

Предварительные работы включают планирование деятельности по сопровождению системы, а также организацию перехода к ее полнофункциональному использованию [1].

Если новая система должна заменить старую систему, предназначенную для решения тех же задач, просто на новом уровне эффективности, стоимости использования, новых функциональных возможностей, в этом случае важно обеспечить плавный переход со старой системы на новую, максимально естественный для пользователей. С этим связано не только планирование, например, переноса информации, хранимой в соответствующих базах данных, но и обучение пользователей, подготовка, настройка и проверка “боевой” конфигурации, определение последовательности операций, организация и обучение службы поддержки [35].

Область знаний «Сопровождение ПО» связана с другими аспектами программной инженерии. По сути, описание этой области знаний непосредственно пересекается со всеми другими дисциплинами.

На рис. 2.1 представлена структурная схема области знаний «Сопровождение ПО».

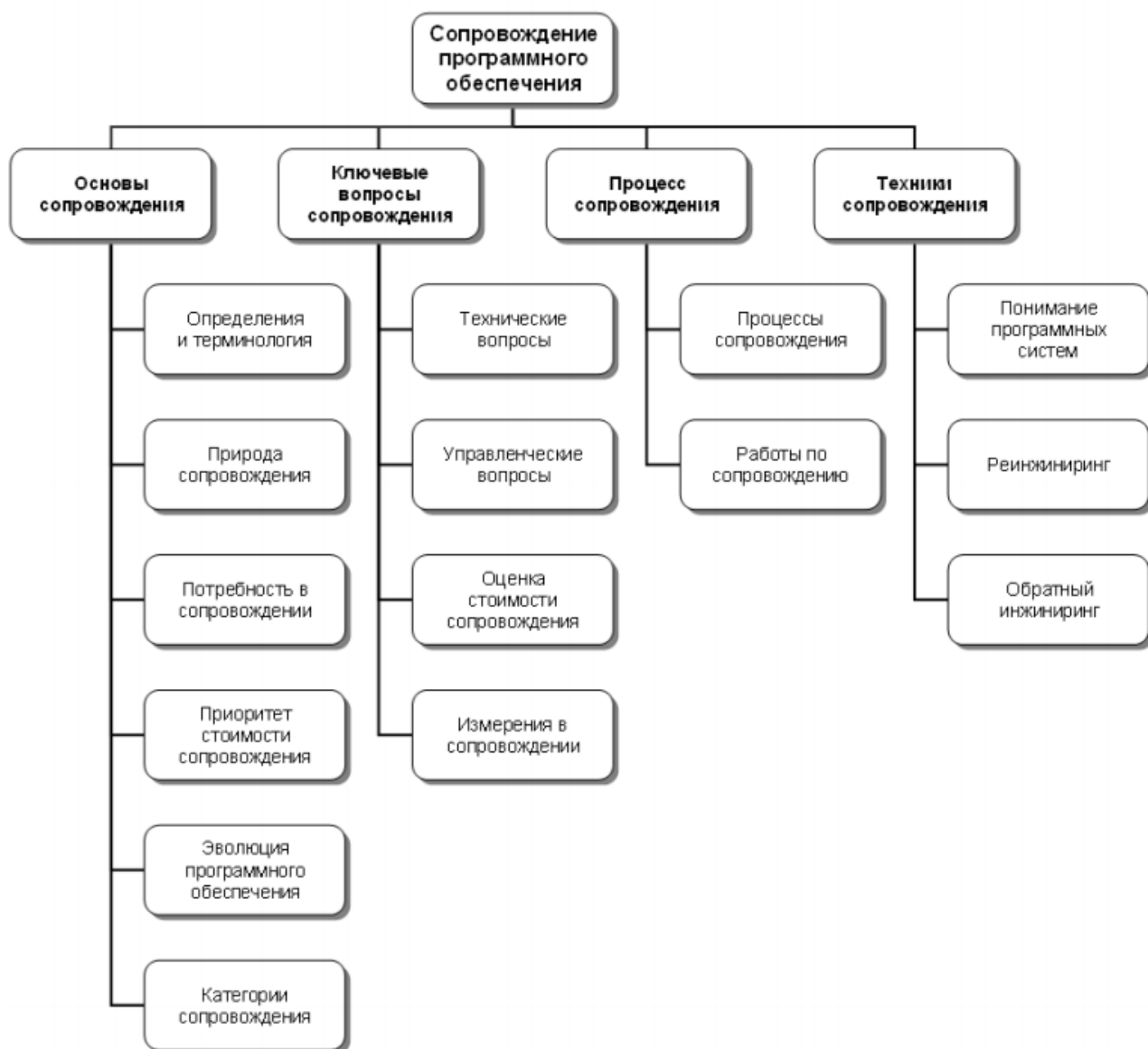


Рисунок 2.1 - Область знаний «Сопровождение ПО»

Основы сопровождения. Эта секция включает концепции и терминологию, формирующие основы понимания роли и содержания работ по сопровождению программных систем. Темы данной секции предоставляют соответствующие определения и описывают, почему именно существует потребность в сопровождении. Категории сопровождения критически важны для понимания сути обсуждаемых вопросов.

Ключевые вопросы сопровождения. Для обеспечения эффективного сопровождения программных систем необходимо решать целый комплекс вопросов и проблем, связанных с соответствующими работами. Необходимо

понимать, что процесс сопровождения предъявляет уникальные технические и управленческие требования к персоналу, занимающемуся сопровождением и, в первую очередь, специалистам-инженерам. Попытка найти дефект в продукте, содержащем 500 тысяч строк кода, написанных другими инженерами – яркий пример сложностей, с которыми приходится сталкиваться инженерам по сопровождению. Другой пример, уже организационный, постоянная борьба за ресурсы с разработчиками (с моей точки зрения, это чаще всего проявляется в вопросах отвлечения разработчиков от текущей работы для помощи в решении проблем сопровождения, а также в конкуренции за приоритеты финансирования разработки новой системы или сопровождения существующей). Одновременное планирование перспективной версии системы, реализация следующей версии и подготовка критических патчей для текущей версии – еще один классический пример проблем, с которыми приходится сталкиваться в процессе эксплуатации ПО. Данная секция представляет некоторые технические и управленческие вопросы, связанные с сопровождением программных систем

Процесс сопровождения. Вопросы организации процесса сопровождения напрямую связаны с соответствующими стандартами и de facto практиками реализации такого процесса. Тема “Работы по сопровождению” (Maintenance Activities) различает вопросы сопровождения и разработки и показывает взаимосвязь с другими аспектами деятельности программной инженерии.

Типичные и распространенные потребности в процессах программной инженерии подробно описаны и документированы в различных источниках. Одна из наиболее детально проработанных и распространенных (на уровне стандарта de facto) процессных моделей, изначально созданных с ориентацией на ПО – CMMI (Capability Maturity Model Integration – интегрированная модель зрелости), разработанная в Институте программной инженерии университета Карнеги-Меллон (SEI CMU). CMMI, в частности, уделяет специальное

внимание процессам сопровождения. Существуют и другие, менее распространенные, но, тем не менее, развивающиеся модели (рисунки 2.2, 2.3).

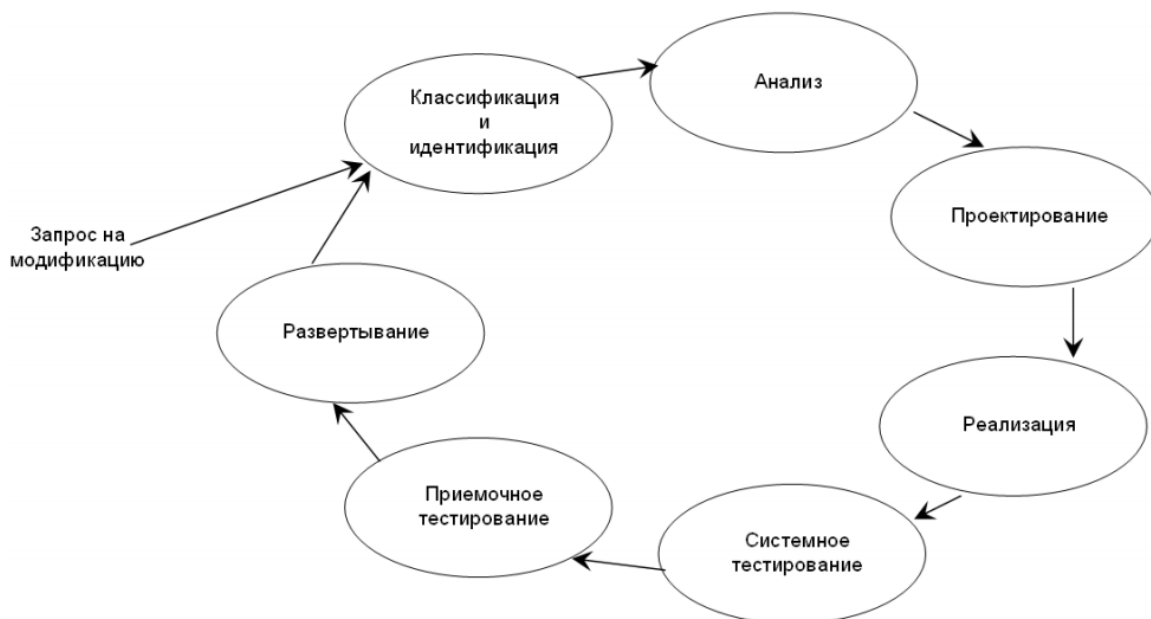


Рисунок 2.2 - Работы в процессе сопровождения по стандарту ГОСТ Р ИСО/МЭК 14764

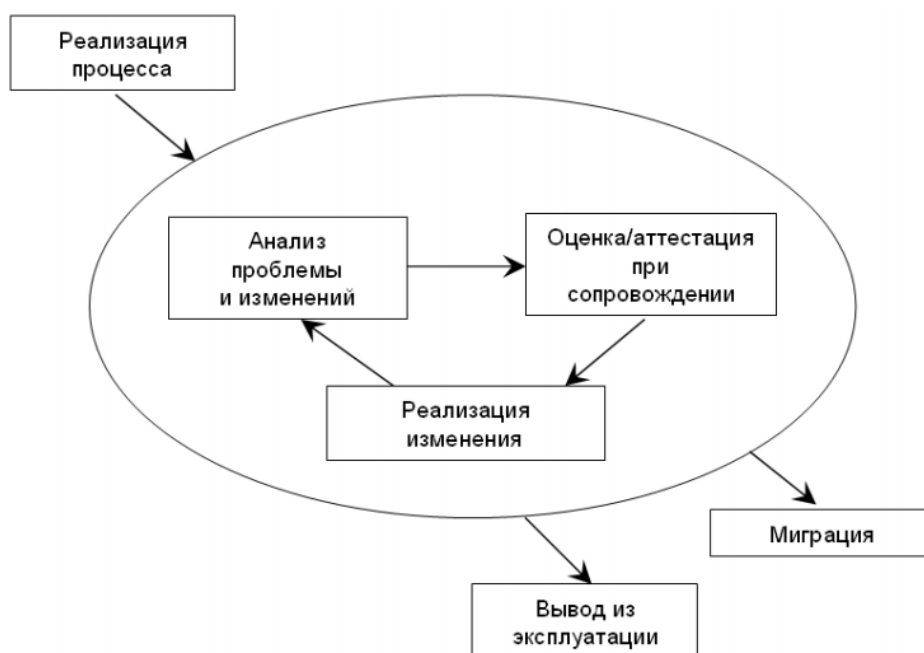


Рисунок 2.3 - Процесс сопровождения по стандарту ГОСТ Р ИСО/МЭК 14764

Многие работы по сопровождению похожи на аспекты деятельности по разработке. В обоих случаях необходимо проводить анализ, проектирование, кодирование, тестирование и документирование. Специалисты по

сопровождению должны отслеживать требования так же, как и инженеры-разработчики и обновлять документацию по мере разработки и/или выпуска обновленных или новых релизов продукта. Стандарт ГОСТ Р ИСО/МЭК 14764 рекомендует, чтобы персонал или организации, отвечающие за сопровождение, в случае использования элементов процессов разработки в своей деятельности, адаптировали эти процессы в соответствии со своими потребностями [2]. В то же время, работы по сопровождению программных средств обладают определенными особенностями, что влечет за собой необходимость использования специализированных процессов.

Техники сопровождения. Данная секция вводит некоторые общепринятые техники, используемые в процессе сопровождения программных систем.

2.2 Основные положения концепции сопровождения программного обеспечения

Описание концепции сопровождения ПО должно быть первым шагом при разработке политики сопровождения программного средства. Концепция сопровождения должна быть разработана сразу же при первом выпуске исходного программного продукта [1].

Концепция сопровождения должна отражать:

- область сопровождения программного средства;
- практическое применение (адаптацию) данного процесса;
- определение организаций (лиц), ответственных за сопровождение;
- оценку стоимости сопровождения.

Область сопровождения должна отражать обязанности сопроводителя, определять, какую поддержку программного средства обязан обеспечить сопроводитель. Область сопровождения зачастую определяется наличием соответствующих бюджетных ограничений. Область сопровождения должна охватывать:

- типы выполняемого сопровождения;
- сопровождаемый уровень документов;
- реакцию на сопровождение;
- обеспечиваемый уровень обучения персонала;
- обеспечение поставки продукта;
- организацию справочной службы.

Основной целью концепции сопровождения является отражение задач сопровождения после выдачи программного продукта. В то же время, различные организации могут выполнять различные задачи в процессе сопровождения. Непосредственно, подобные организации следует попытаться определить в самой концепции сопровождения. Используемый процесс сопровождения, обязательно должен быть отражен в концепции сопровождения.

Определение ответственных специалистов за сопровождение, отвечающих за сопровождение продукта, является важной частью концепции сопровождения. Это в равной степени справедливо и в случае внутреннего сопровождения в самой организации. При выполнении сопровождения по соглашению с третьей стороной (аутсорсинг) это должно быть отмечено в концепции сопровождения. В основных процессах заказа и поставки по ГОСТ Р ИСО/МЭК 12207 детально описаны услуги по заказу и поставке.

Выбор сопроводителя обосновывается следующими факторами:

- срок службы программного средства;
- размер долгосрочных затрат;
- размер первоначальных затрат;
- наличие соответствующего места;
- квалификацию персонала;
- работоспособность программного продукта;
- программу (график) сопровождения;
- знание предметной области применения программного продукта.

Стоимость сопровождения должна зависеть от области сопровождения.

Дополнительными факторами, подлежащими учету, являются стоимость:

- проезда до места расположения пользователя;
- обучение, как сопровождающих, так и пользователей;
- персонала (зарплаты и премии).

При разработке концепции сопровождения стоимость оценивают на основе ограниченных исходных данных. В дальнейшем эти оценки должны быть уточнены. В качестве исходных данных при оценке стоимости сопровождения могут быть использованы соответствующие архивные данные по аналогичным проектам [2].

2.3 Описание процесса сопровождения

Для достижения поставленной цели, необходимо проанализировать текущее состояние использования системы отслеживания ошибок CR Control в процессе сопровождения ПО, написанного на языке COBOL. Наиболее удобным способом будет построение IDEF0 диаграммы процессов [4]. Для этого была выбрана кроссплатформенная система моделирования и анализа бизнес-процессов – Ramus, которая позволяет:

- разрабатывать графические модели бизнес-процессов (поддерживаются нотации IDEF0 и DFD);
- разрабатывать системы классификации и кодирования (с привязкой к моделям процессов);
- формировать отчетность по моделям и системе классификации (в виде регламентов бизнес-процессов, должностных инструкций и т.п.).

С точки зрения специалиста службы поддержки основным является процесс сопровождения ПО.

Там, где на вход поступает сообщение о проблеме от клиента, служба поддержки должна осуществить полный процесс обработки этого сообщения и

последующее устранения данной проблемы. В результате чего на выходе будет отчет о проведенных изменениях и модифицированное ПО.

На рис. 2.4 представлена контекстная диаграмма «AS-IS» («как есть») в нотации IDEF0 процесса «Сопровождение ПО».

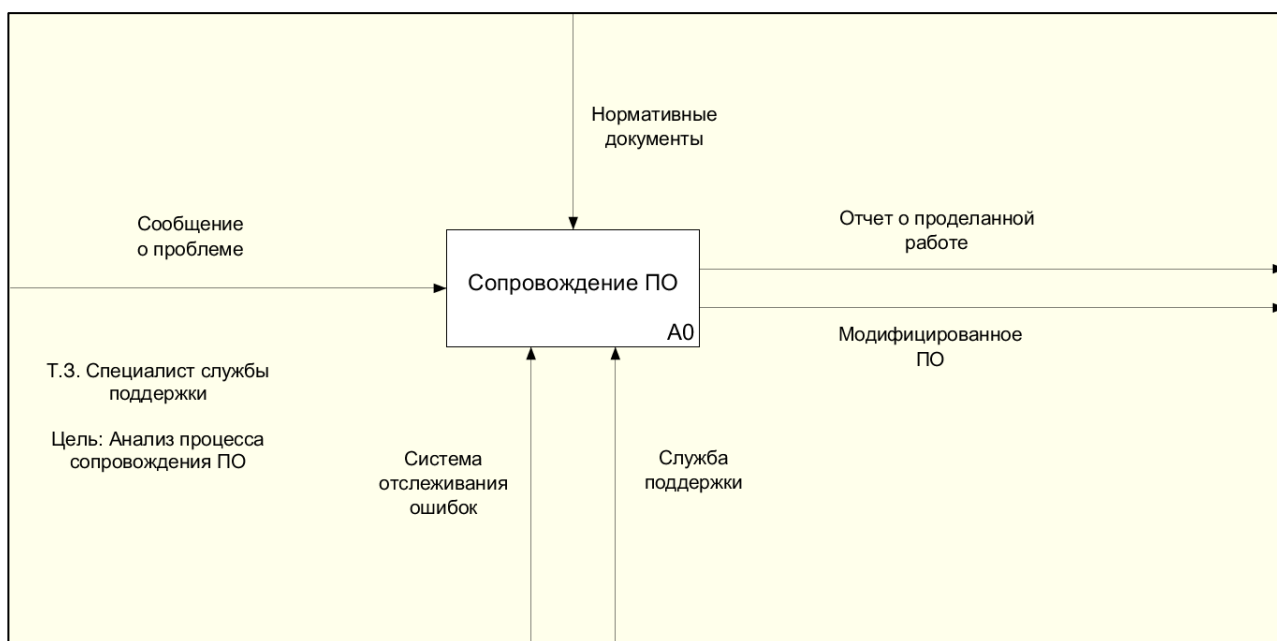


Рисунок 2.4 – Контекстная диаграмма «AS-IS» процесса «Сопровождение ПО»

На рис. 2.5 представлена декомпозиция процесса сопровождения, в который входят следующие операции:

- 1) классифицировать проблему;
- 2) провести анализ проблемы;
- 3) подготовить и внести изменения;
- 4) протестировать и выдать изменения;
- 5) формирование отчета.

Следует отметить, что в каждой операции участвует такой механизм, как система отслеживания ошибок, с помощью которой и происходит автоматизация данного процесса.

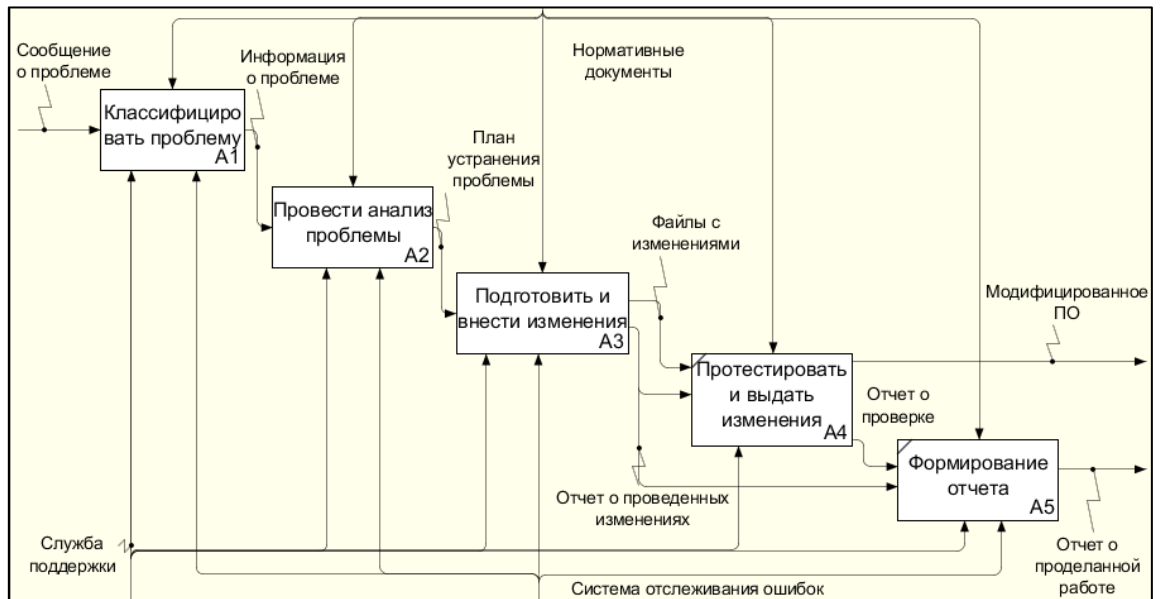


Рисунок 2.5 – Декомпозиция контекстной диаграммы «AS-IS» процесса «Сопровождение ПО»

Данная диаграмма отражает основные операции во время осуществления процесса сопровождения, некоторые будут рассмотрены более подробно.

На рис. 2.6 представлена декомпозиция процесса «Классифицировать проблему».

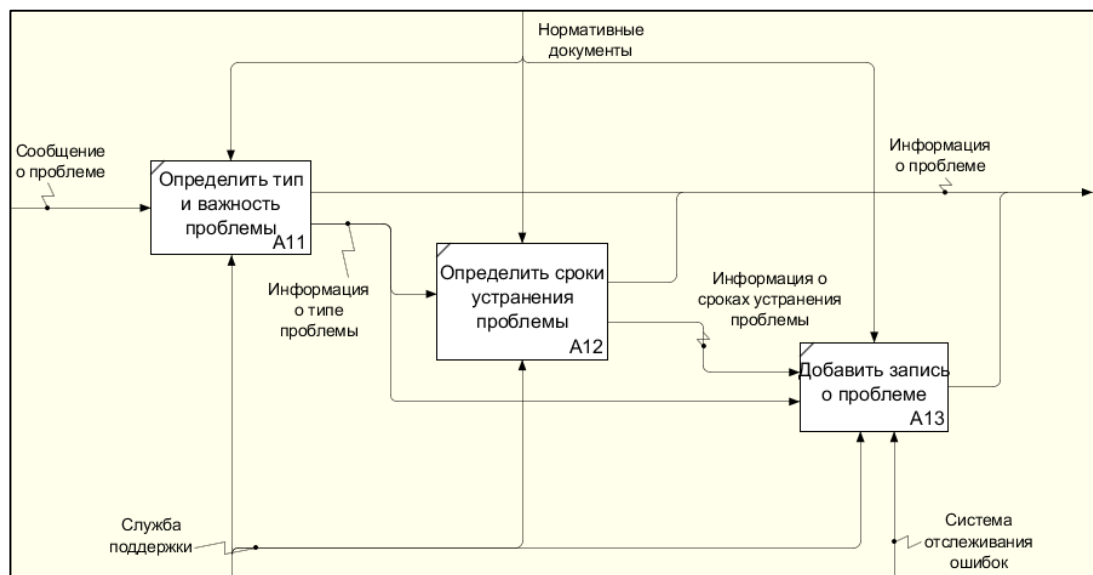


Рисунок 2.6 – Декомпозиция процесса «Классифицировать проблему» («AS-IS»)

На данной диаграмме можно увидеть, что при начально поступлении сообщения о проблеме определяется тип и степень ее важности, после чего

устанавливаются сроки устранения проблемы, на которые будет ориентироваться уже конкретный специалист, устраняющий проблему.

На рис. 2.7 представлена декомпозиция процесса «Провести анализ проблемы».

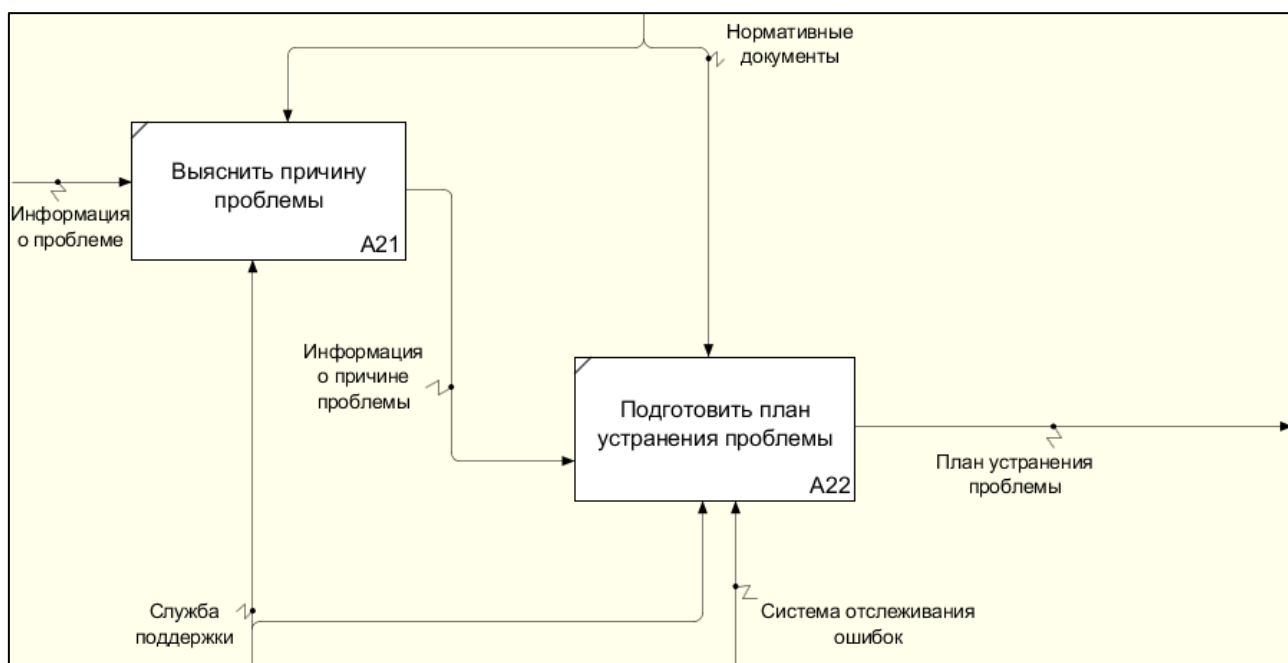


Рисунок 2.7 – Декомпозиция процесса «Провести анализ проблемы» («AS-IS»)

Приведенная диаграмма отражает основные операции, проводимые специалистом службы поддержки во время анализа проблемы, которые включают в себя выяснения причин возникновения проблемы и подготовка плана устранения проблемы. Процесс выяснения причин проблемы полностью осуществляется сотрудниками отдела технической поддержки, в то время как в составлении плана устранения найденной проблемы принимает участия система отслеживания ошибок.

На рис. 2.8 представлена декомпозиция процесса «Подготовить и внести изменения».

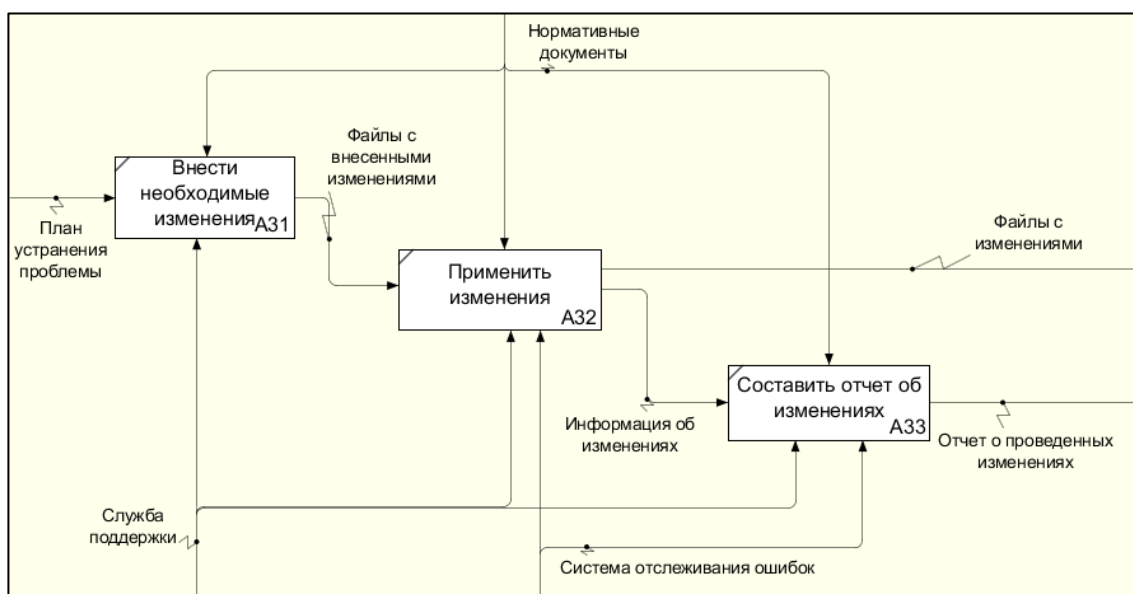


Рисунок 2.8 – Декомпозиция процесса «Подготовить и внести изменения» («AS-IS»)

Можно заметить, что согласно плану устранения проблемы в программу вносятся необходимые изменения, после чего составляется отчет о проведенных изменениях.

Анализ построенной структурно-функциональную модель показал, что для более эффективного классифицирования и устранения проблемы необходимо вести собственную базу знаний сотрудников, в которой будут храниться аналитические отчеты.

2.4 Концептуальное моделирование процесса сопровождения

Общепринятая технология проектирования информационной системы (ИС) подразумевает сначала создание модели AS-IS, затем на основе ее анализа определяются направления улучшения процессов, и строится модель TO-BE («как должно быть»). На основе разработанной модели TO-BE далее происходит построение модели данных, прототипа и окончательного варианта ИС.

Целью создания модуля системы является автоматизация процесса сопровождения, что подразумевает построение единого информационного пространства, которое должно включать в себя все прежние процессы, но их

выполнение должно обеспечиваться совместно специалистом службы поддержки и информационной системой. На рис. 2.9 представлена модель процесса сопровождения ПО после автоматизации в нотации IDEF0 («ТО-ВЕ» - «как должно быть»).

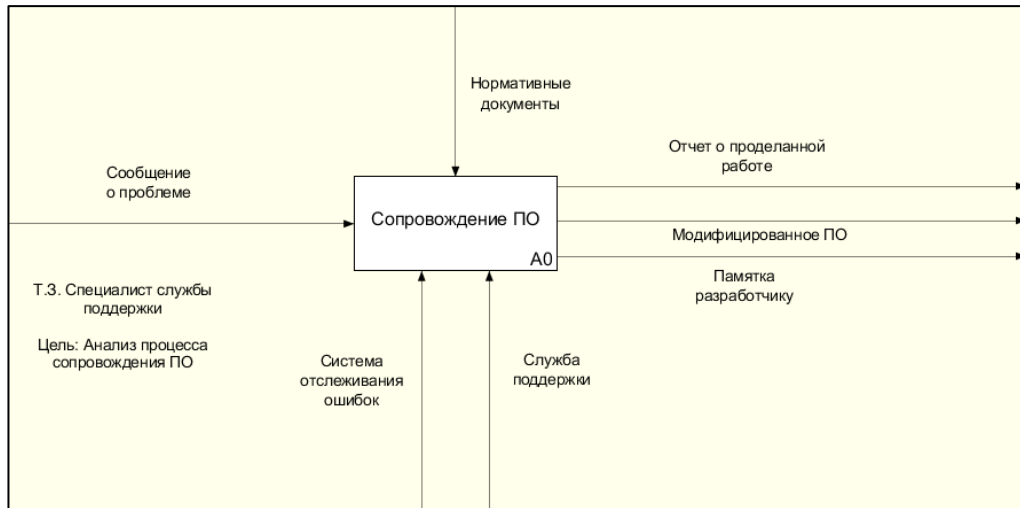


Рисунок 2.9 – Контекстная диаграмма «ТО-ВЕ» процесса «Сопровождение ПО»

Если сравнить контекстную диаграмму ТО-ВЕ с диаграммой AS-IS, то можно заметить, что на выходе добавлена памятка разработчику.

На рис. 2.10 представлена декомпозиция процесса сопровождения ПО после автоматизации в нотации IDEF0 («ТО-ВЕ» - «как должно быть»).

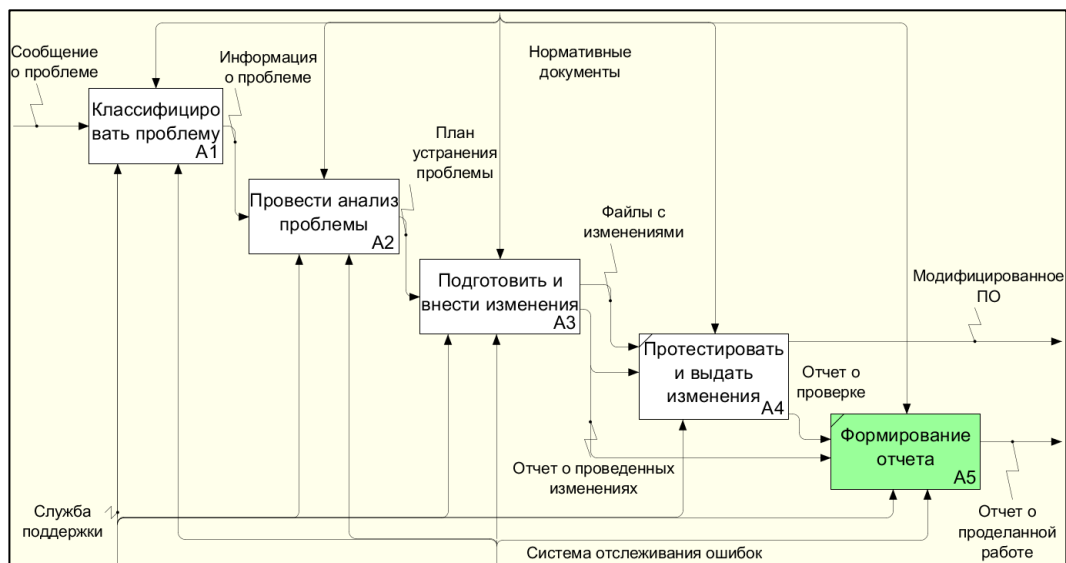


Рисунок 2.10 – Декомпозиция контекстной диаграммы «ТО-ВЕ» процесса «Сопровождение ПО»

На рис. 2.11 представлена декомпозиция процесса «Провести анализ проблемы».

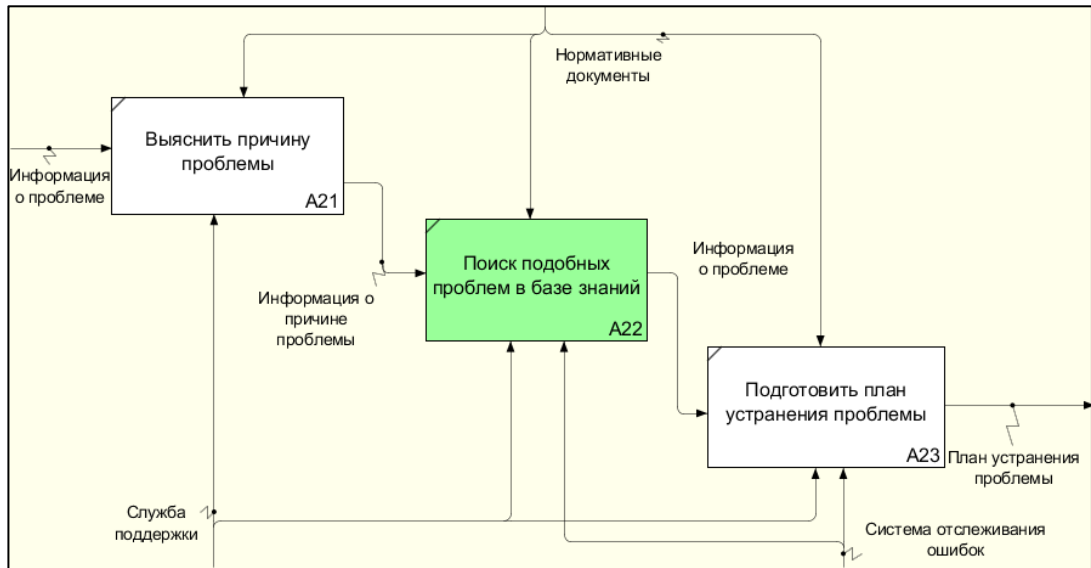


Рисунок 2.11 – Декомпозиция процесса «Провести анализ проблемы» («ТО-ВЕ»)

Можно заметить, что появился новый процесс «Поиск проблем в базе знаний», помогающий разработчику облегчить решение проблемы путем использования опыта в решении более ранних проблем.

На рис. 2.12 представлена декомпозиция процесса «Подготовить и внести изменения».

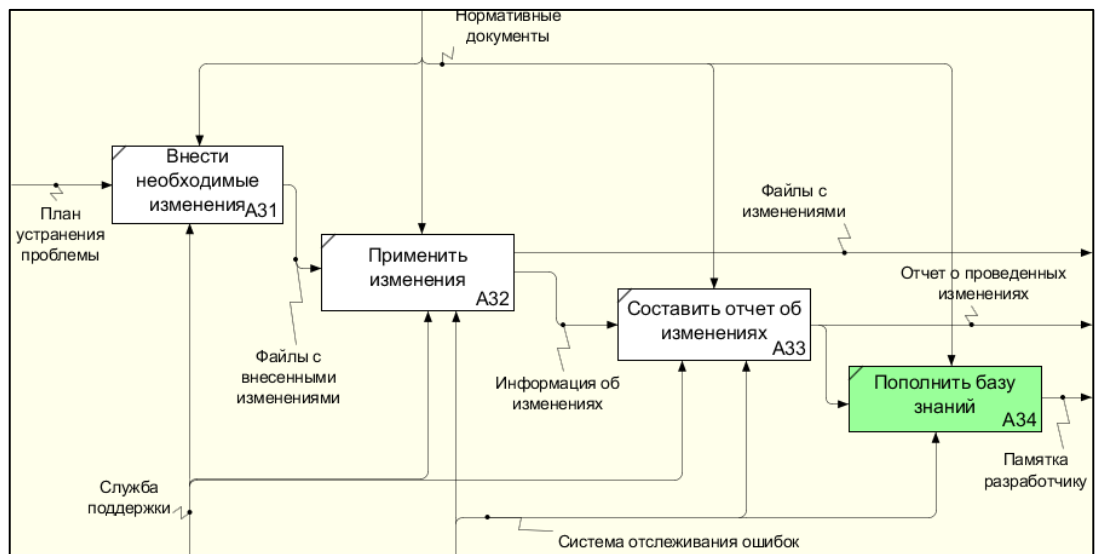


Рисунок 2.12 – Декомпозиция процесса «Подготовить и внести изменения» («ТО-ВЕ»)

На диаграмме можно увидеть, что специалист службы поддержки должен вносить данные по проблеме в базу знаний системы отслеживания ошибок.

На основе выводов, сделанных при анализе диаграммы ТО-ВЕ, получаем, что внедрение модуля системы отслеживания ошибок позволит автоматизировать некоторые функции специалиста службы поддержки и существенно облегчит его работу.

На диаграмме представлен весь процесс сопровождения ПО «как должно быть». Основные функции специалистов службы поддержки существенно не изменились, но несколько процессов будут автоматизированы при помощи системы и добавлены несколько процессов.

Таким образом, система отслеживания ошибок будет справляться с поставленными ранее задачами и существенно облегчит деятельность службы поддержки.

Непосредственно действия, выполняемые во время обнаружения дефекта и до конечного его устранения и выдачи результатов клиенту с использованием потоков данных, отражены с помощью DFD-диаграммы.

Диаграммы потоков данных (Data Flow Diagrams — DFD) представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления — продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Диаграмма потоков данных являлась одним из основных инструментов структурного анализа и проектирования ИС, до появления унифицированного графического языка моделирования UML [26].

При построении диаграммы были выделены две основные внешние сущности – клиент и специалист службы поддержки. Сама диаграмма DFD представлена на рис. 2.13.

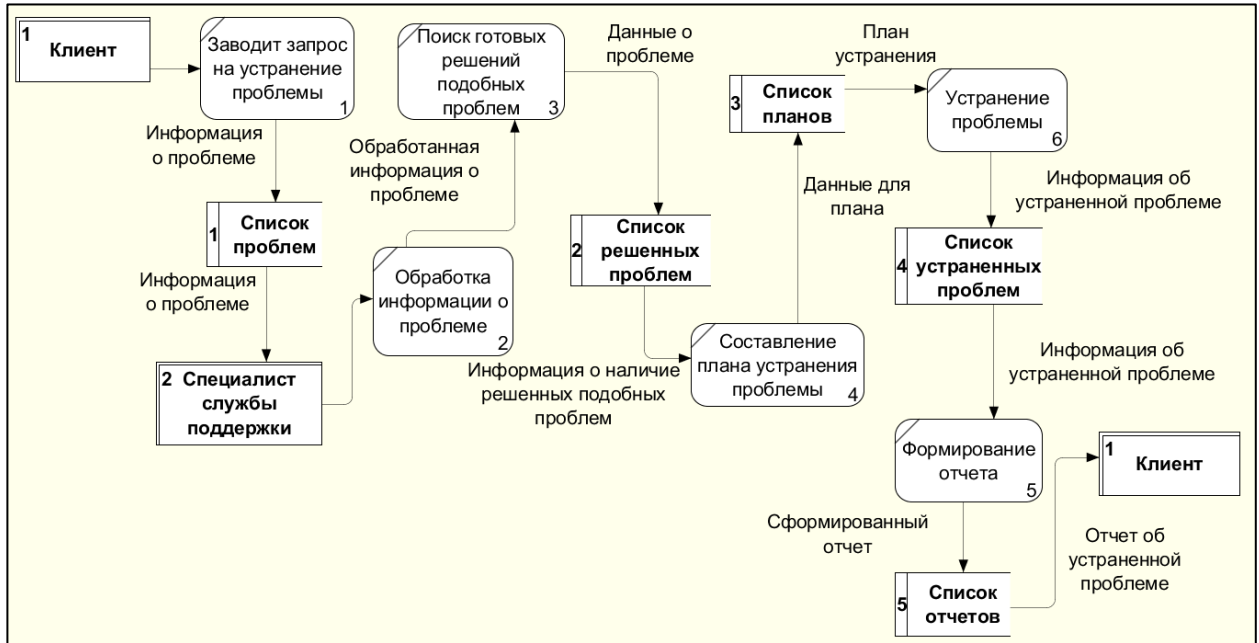


Рисунок 2.13 – Диаграмма DFD сопровождения ПО в нотации Гейна-Сарсона

При построении диаграммы были выделены две основные внешние сущности – клиент и специалист службы поддержки. Клиент регистрирует проблему в системе отслеживания ошибок, которая вносится в базу данных, после чего специалист службы поддержки обрабатывает зарегистрированную проблему, проверяет наличие готовых решений для подобной проблемы. Далее на основе полученных данных составляется план устранения проблемы, который также вносится в базу.

На основе указанного плана при помощи системы происходит устранение проблемы и формируется отчет о проделанной работе.

Вывод по главе

1) Анализ структурно-функциональной модели процесса сопровождения ПО «как есть» показал, что для более эффективного классифицирования и устранения проблемы необходимо вести собственную базу знаний сотрудников, в которой будут храниться аналитические отчеты.

2) Для улучшения процесса сопровождения ПО его необходимо автоматизировать, что подразумевает построение единого информационного пространства, которое должно включать в себя все прежние процессы, но их выполнение должно обеспечиваться совместно специалистом службы поддержки и информационной системой.

3) Концептуальная модель системы сопровождения ПО представляет собой развернутую модель «как должно быть», построенную в нотации методологии DFD.

ГЛАВА 3. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА КОМПОНЕНТОВ СИСТЕМЫ ОТСЛЕЖИВАНИЯ ОШИБОК «CR CONTROL»

3.1 Формализованное описание процесса сопровождения программного обеспечения в компании ООО «НетКрэкер»

На стадии разработки формализованного описания процесса сопровождения ПО применена методология объектно-ориентированного анализа на основе языка UML.

Объектно-ориентированный подход использует объектную декомпозицию, при этом статическая структура системы описывается в терминах объектов и связей между ними, а поведение системы - в терминах обмена сообщениями между объектами. Главными достоинствами визуальной модели являются ясность представления выбранного решения и относительная простота программной реализации.

Язык визуального моделирования UML охватывает все основные стадии объектно-ориентированного анализа и дизайна и предназначен для построения набора моделей, описывающих проектируемую систему.

Модель UML состоит из элементов. Элементы - это классы, объекты, компоненты, атрибуты и другие понятия объектной модели, между которыми устанавливаются отношения.

В соответствии с нотацией UML, каждый элемент имеет свое графическое представление. Средства автоматизации моделирования, построенные на основе UML, естественно, включают средства графического визуального представления моделей. Благодаря высокому авторитету авторов (и умелой маркетинговой политике), язык UML стал для средств визуального моделирования стандартом де-факто.

В методологии языка UML модель представляет собой совокупность диаграмм, описывающих различные аспекты структуры и поведения исследуемой системы. Однако при всем многообразии их типов при логическом моделировании ИС используются, как правило, диаграммы,

составляющие ядро языка (в частности, диаграмма вариантов использования, диаграмма классов и диаграмма взаимодействия).

В качестве разрабатываемого программного обеспечения был выбран модуль системы отслеживания ошибок.

Для более наглядного представления обработки данных подобной системой была составлена диаграмма деятельности UML, описывающая жизненный цикл обнаруженного программного дефекта от момента обнаружения, до момента устранения (рис. 3.1).

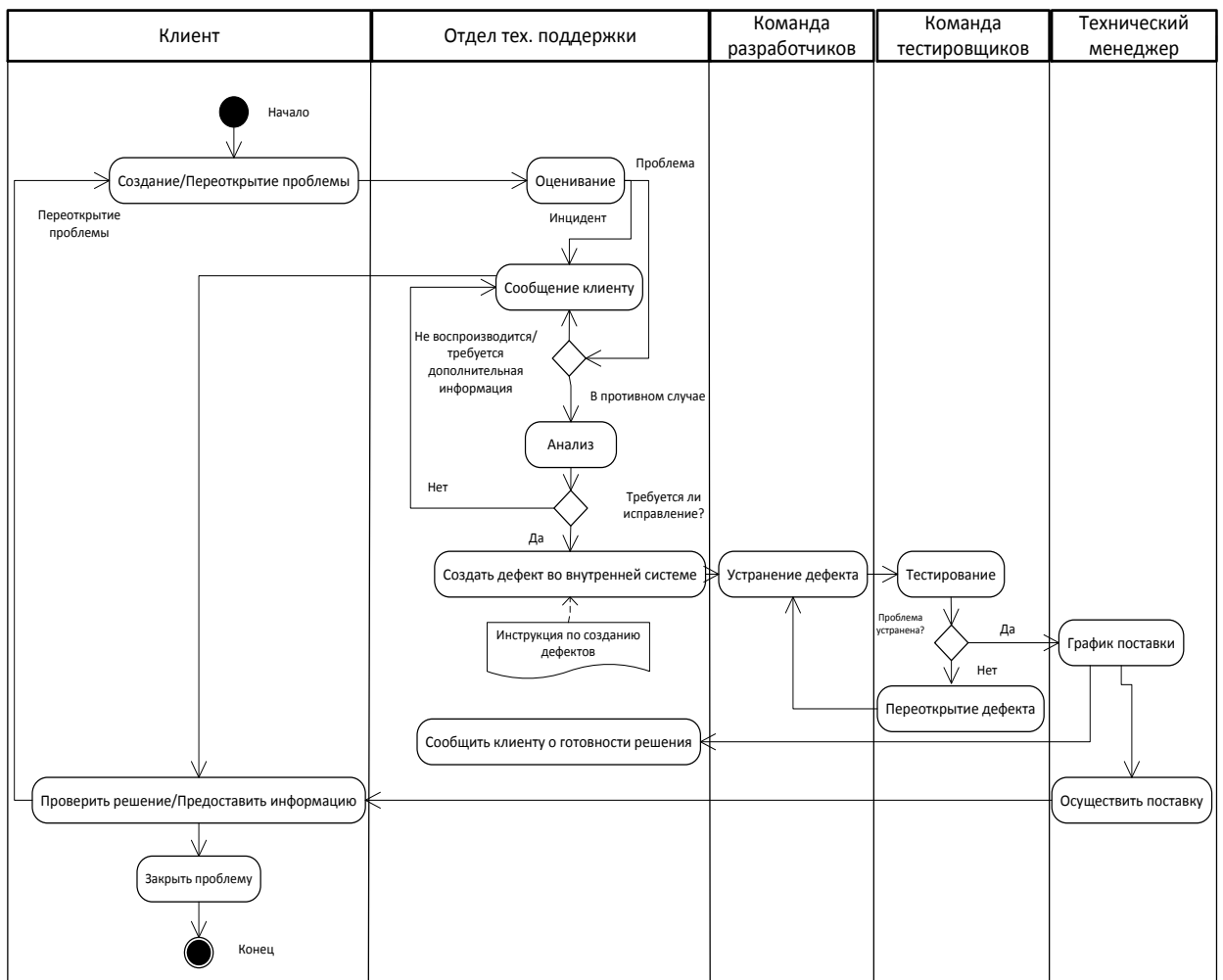


Рисунок 3.1 – Диаграмма деятельности жизненного цикла дефекта программного обеспечения

На рисунке представлены этапы по процессу устранения (жизненный цикл) дефекта в системе Wizard. Заказчик после обнаружения какой-либо неисправности в функциональности программной системы (ПС) оповещает об

ошибке специалиста технической поддержки ИТ-компании и предоставляет ему всю имеющуюся информацию по данной проблеме, на основании которой последний дает рекомендации заказчику по ее решению.

Если сведений об обнаруженной ошибке недостаточно для выработки решения по ее устранению, то об этом сообщается заказчику, после чего производится более детальный анализ проблемы, а выявленный дефект вносится в перечень отслеживаемых ошибок.

Далее вся информация поступает в команду разработчиков, которые занимаются устранением обнаруженного дефекта системы на уровне программного кода. После устранения дефекта выполняется дальнейшее тестирование функциональности ПС. Если дефект устранен и не обнаружены новые ошибки, составляется график выдачи модифицированного ПО заказчику.

В противном случае ПС дорабатывается разработчиками до полного устранения дефектов.

В качестве примера более детально рассмотрим порядок выполняемых работ разработчиком для устранения обнаруженного дефекта ПС.

На рис. 3.2 изображена диаграмма деятельности процесса сопровождения ПО (устранение неисправностей или модернизация функциональности) выглядит с точки зрения разработчика отдела поддержки.

Из представленной диаграммы следует, что после получения задания разработчик проводит собственный анализ, если есть вопросы, то задает их, после чего проверяет текущий функционал и, если всё понятно, приступает к написанию плана устранения проблемы. Затем происходит внедрение изменений разработчика и их проверка, и, если всё прошло хорошо, то изменения выдаются на проверку уровню выше, а оттуда, отправляются заказчику или возвращаются разработчику.

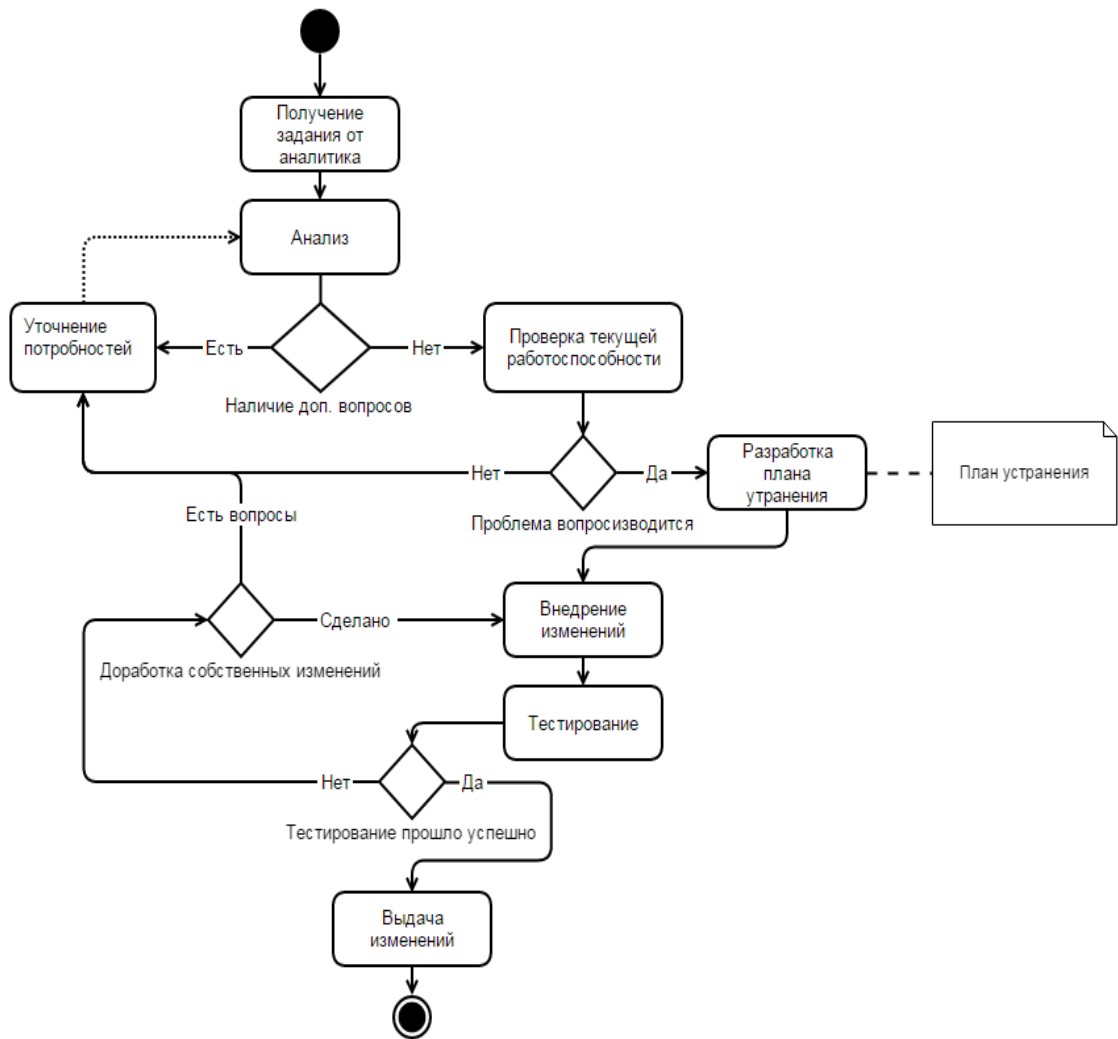


Рисунок 3.2 – Диаграмма деятельности процесса устранения неисправности разработчиком

При выполнении анализа был рассмотрен весь процесс устранения обнаруженного дефекта в системе Wizard, написанной на языке программирования COBOL.

3.2 Архитектура системы отслеживания ошибок CR Control

В качестве языка программирования использовался язык программирования Perl, с помощью которого разрабатывались модули к системе CR Control.

Perl относится к категории интерпретируемых языков. Он используется для обработки больших массивов текстовых данных. Синтаксически Perl

организован по аналогии языком С. Отсутствие ограничений на глубину рекурсий позволяет значительно уменьшать тело разработанной на Perl программы.

Perl позволяет использовать для поиска и управления данными регулярные выражения и создавать объекты с определенными свойствами, что существенно упрощает разработку процедур управления базами данных, в том числе спроектированных в среде Oracle.

Для работы серверной части системы необходим веб-сервер с поддержкой Perl 5.x и системы управления базами данных (СУБД) Oracle. В качестве клиента используется любой доступный браузер.

3.3 Проектирование базы данных системы CR Control

Логическое проектирование - создание схемы базы данных на основе конкретной, например, реляционной модели данных. Для реляционной модели данных логическая модель - набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Концептуальная модель преобразуется в логическую по формальным правилам. Автоматизация этого процесса осуществляется путем использования комплексных программных средств.

Логический уровень – это абстрактный взгляд на данные, на нем данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире. Логическая модель данных является универсальной и никак не связана с конкретной реализацией баз данных.

Логическая модель данных является начальным прототипом будущей базы данных. От того, насколько хорошо разработана логическая структура базы данных, зависит качество конечных – физических, моделей, и, в конечном счете, эффективность работы базы данных (которая непосредственно завязана на эффективность работы всей ИС в целом). Поэтому при разработке

логической модели необходимо уделить особое внимание качеству проектирования.

Главное преимущество реляционной модели данных, используемое в настоящей разработке - это практическая возможность объединения на логическом уровне проблемно-ориентированного программного кода и оперативных данных системы в рамках единой базы данных.

Реляционное моделирование данных рекомендуется производить с помощью метода IDEF1X и CASE-средства ERWin. Методология IDEF1X относится к типу методологий «сущность - связь» и специально разработана для построения реляционных баз данных.

Сущность в IDEF1X описывает собой совокупность или набор экземпляров похожих по свойствам, но однозначно отличаемых друг от друга по одному или нескольким признакам. Каждый экземпляр является реализацией сущности.

Методология IDEF1X подразделяется на логический и физический уровни, соответствующие проектируемой модели данных системы.

Логический уровень моделирования данных позволяет создавать концептуальную модель данных, базирующиеся на понятии ER- диаграммы (Entity Relationship Diagram, диаграммы «сущность-связь»), представляющей модель данных в виде сущностей, их атрибутов и отношений между ними.

В свою очередь физическая модель данных в методологии IDEF1X содержит специфические объекты реальной базы данных, такие как таблицы, поля и их типы данных и обычно создается на основе логической модели данных.

Применение метода IDEF1X наиболее целесообразно для построения логической модели данных лишь после того, как все бизнес-процесс проанализированы и решение о внедрении реляционной базы данных, как части ИС, было принято окончательно.

Процесс разработки логической модели данных по методу IDEF1X можно разбить на следующие этапы:

- выделение сущностей с помощью ERD-метода путем сопоставления с диаграммой классов, которая позволяет импортировать объектную модель Rational Rose в модель данных ERwin или AllFusion ERwin Data Modeler);
- определение зависимостей между сущностями (ассоциации и агрегации преобразуем в неидентифицирующие отношения);
- задание первичных и альтернативных ключей;
- определение неключевых атрибутов сущностей;
- построение IDEF1X –диаграммы с помощью доступных CASE- средств;
- генерация SQL- скриптов реализуемой реляционной базы данных.

Этап наиболее эффективного размещения данных, то есть отображении логической структуры базы данных (БД) в структуру хранения, учитывая специфику конкретной модели данных, и не учитывая специфики конкретной СУБД.

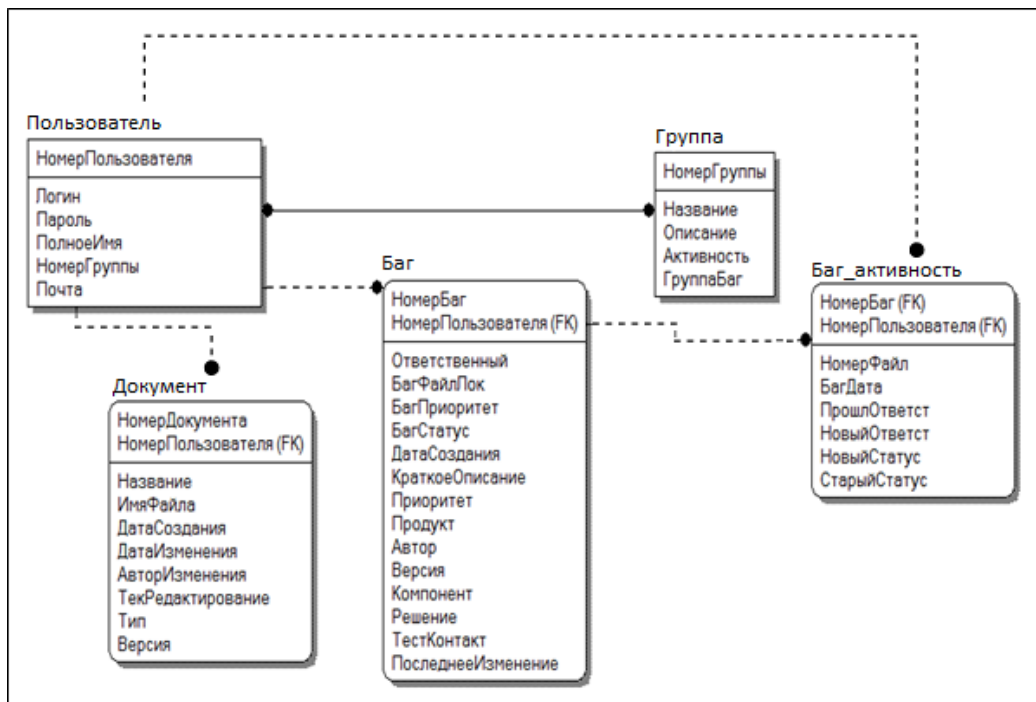


Рисунок 3.3 – Логическая модель данных системы отслеживания ошибок «CR Control»

На рис. 3.3 представлено логическое представление части БД, которая непосредственно использовалась во время разработки. Была создана новая сущность «Документ», для последующей ее реализации с помощью модуля системы «Знания».

Физическая модель базы данных определяет способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне. Результатом физического моделирования является генерация программного кода базы данных на соответствующем выбранной СУБД диалекте структурированного языка запросов SQL.

На рис. 3.4 представлена физическая модель данных модулей системы CR Control, построенная на основе логической модели данных с учетом особенностей типизации данных в СУБД Oracle.

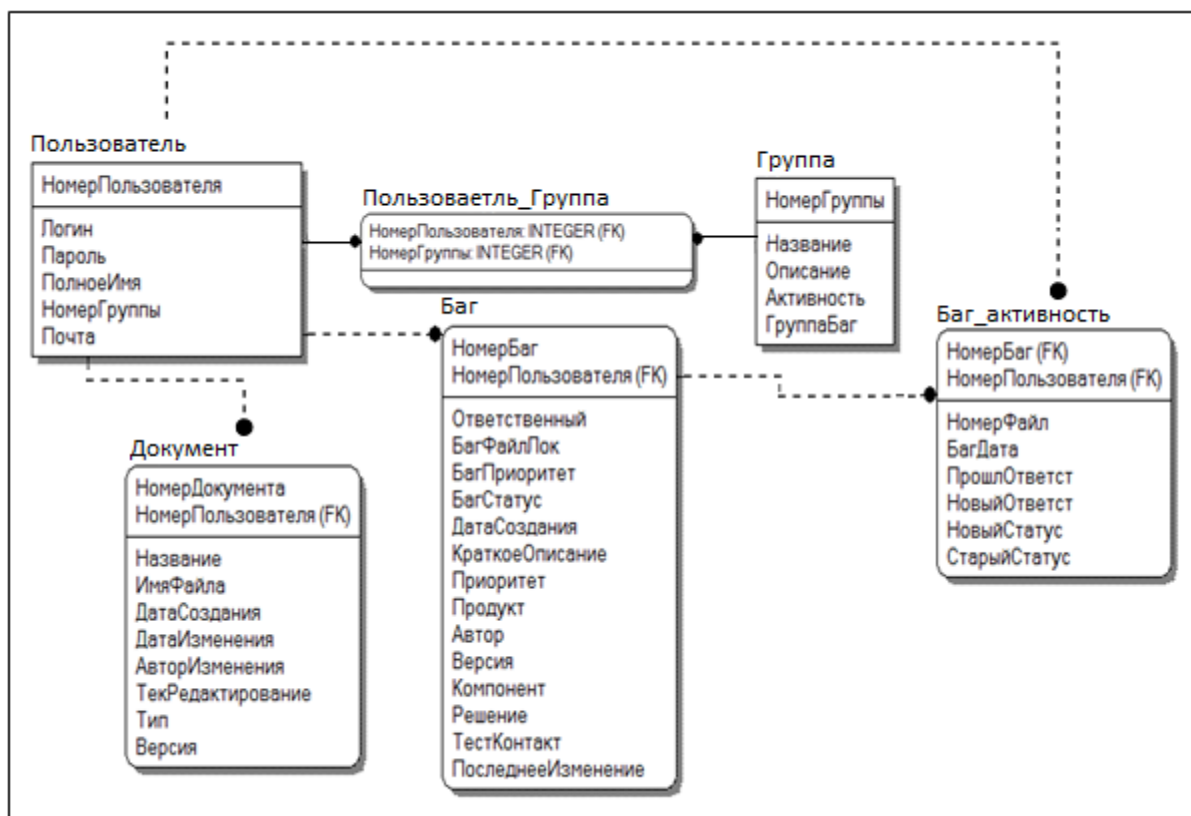


Рисунок 3.4 – Физическая модель данных системы отслеживания ошибок «CR Control»

При сравнении физической и логической моделей данных можно заметить, что структура база данных не изменилась, но добавились типы

данных полей, используемые в СУБД Oracle. Также, была создана дополнительная сущность Пользовател_Группа, позволяющая устранить связь «много-ко-многим» между сущностями Пользователь и Группа.

При разработке логической и физической модели данных использовалось CASE - средство ERWin.

Рассматриваемая структура представляет собой фрагмент БД всей системы отслеживания ошибок.

Интегрированная таблица «Документ» хранит информацию обо всех документах, загруженных на сервер при помощи модуля «Знания». К таким документам могут относиться файлы с программным кодом, нормативные документы или отчеты программистов об устранённых дефектах. Кроме того, в указанной таблице хранится информация об обучающих документах (табл. 3.1).

Таблица 3.1 – Структура сущности «Документ»

| Название поля | Ключ | Тип поля | Null значения |
|-------------------|---------------------------------|--------------|---------------|
| НомерДокумента | Primary key (Первичный ключ) | Integer | NOT NULL |
| Автор | Foreign key (Внешний ключ) | Integer | NOT NULL |
| Название | - | Varchar2(45) | NOT NULL |
| ИмяФайла | - | Varchar2(45) | NOT NULL |
| ДатаСоздания | - | Date | NOT NULL |
| ДатаИзменения | - | Date | NULL |
| АвторИзменя | Foreign key (Внешний ключ) | Integer | NULL |
| ТекРедактирование | - | Integer | NULL |
| Тип | - | Varchar2(45) | NOT NULL |
| Версия | - | Integer | NOT NULL |

Ниже приведено более детально описание полей сущности:

- НомерДокумента – идентификатор документа, загруженного на сервер;
- Автор – идентификатор пользователя, который загрузил файл на сервер;
- Название – название документа, отображаемое на странице;
- ИмяФайла – название самого файла, который был загружен;
- ДатаСоздания – дата и время загрузки файла на сервер;
- ДатаИзменения – дата и время изменения файла на сервере;
- ДатаИзменения – идентификатор пользователя, который изменял файл на сервере;
- ТекРедактирование – отображение блокировки документа пользователем;
- Тип – тип документа;
- Версия – версия документа на сервере, после каждого изменения версия увеличивается.

Описанные поля сущности удовлетворяют всем основным требованиям, предъявляемым разрабатываемому функционалу системы отслеживания ошибок CR Control.

Описанная модель БД предоставляет возможность реализовать все необходимые модули и функции системы отслеживания ошибок.

3.4 Описание функционала системы отслеживания ошибок «CR Control»

Система CR Control (далее – система) была разработана компанией Convergys более 15 лет назад и используется по настоящее время в компании ООО «НетКрэкер» в качестве системы отслеживания ошибок для осуществления сопровождения программного комплекса Wizard.

В ее функциональной части реализованы основные возможности подобных систем, например такие, как регистрация дефектов, отражения их жизненных циклов, контроль версий программных файлов, составление отчетов и др.

Система имеет широкие функциональные возможности, к которым можно отнести полнотекстовый поиск, позволяющий находить нужные задачи по конкретным полям, таким, как тема, описание, комментарии, прикрепленные файлы, или текст в формате кода. Кроме того, система поддерживает поиск по тексту на нескольких языках, включая английский, русский и иврит.

Есть возможность массово изменять поля задач, отмечать задачи, назначать ответственных, зависимости и многое другое. Можно указать собственные состояния, типы или приоритеты, установить значения по умолчанию, подсвечивать значения поля, допускать назначение нескольких ответственных лиц, настроить приватность полей. Разумеется, все настраиваемые поля поддерживаются в поисковых запросах, командах и отчетах. Помимо этого, присутствует доска задач пользователей, большой выбор отчетов, ведение истории и много другое (рис. 3.5).

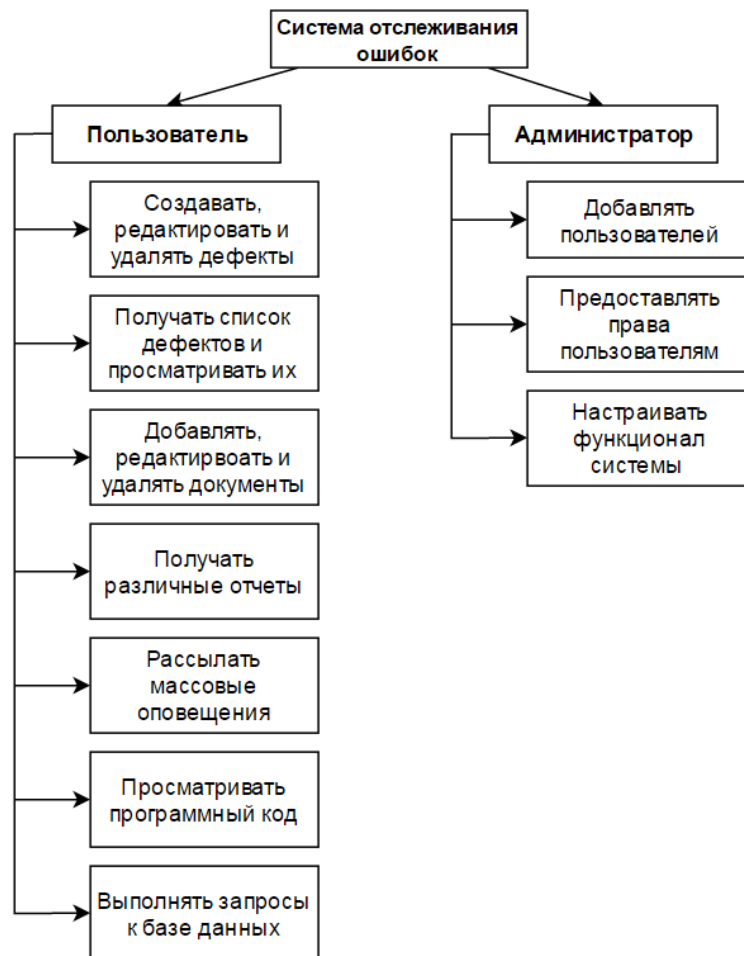


Рисунок 3.5 – Функциональная схема системы CR Control

Во время исследования дорабатывался функционал некоторых модулей, а именно: редактирование дефектов; добавление, редактирование и удаление документов; формирование отчетов.

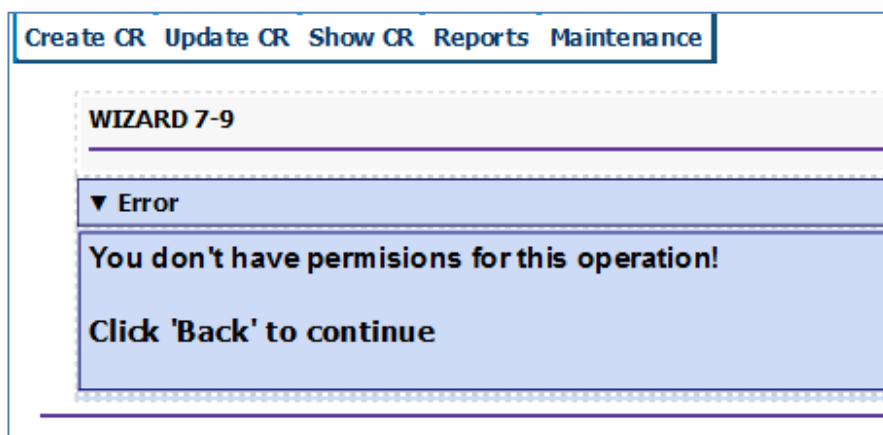
Авторизация в системе (рис. 3.6) организована с помощью формы авторизации, где пользователь указывает свои логин и пароль, в результате чего, ему предоставляется работа с доступным функционалом.



The screenshot shows a web browser window with the title 'WIZARD 7-9'. Below the title is a 'Login' section with the heading 'Log in with your personal ID and password:'. There are two input fields: 'User ID:' containing the text 'ans0714' and 'Password:' with masked characters. A 'Login' button is positioned below the password field. At the bottom of the page, there is a footer with the text: 'Learning Center | Home | The Container | ERP', 'www.NetCracker.com', and '©Version 2.4 Page updated 04/26/2016'.

Рисунок 3.6 – Форма авторизации пользователя

Если пользователю доступны некоторые функции с правами «только на чтение», а он пытается выполнить какие-либо иные действия, он получит на экране сообщение о том, что ему не хватает прав доступа и ему необходимо запросить доступ, если он необходим и обоснован (рис. 3.7).



The screenshot shows a web browser window with a navigation bar at the top containing links: 'Create CR', 'Update CR', 'Show CR', 'Reports', and 'Maintenance'. Below the navigation bar is a section titled 'WIZARD 7-9'. Underneath, there is an 'Error' section with the message: 'You don't have permissions for this operation!' and the instruction: 'Click 'Back' to continue'.

Рисунок 3.7 – Форма сообщения об отсутствии прав доступа

Одной из основных функций любой системы отслеживания ошибок является регистрирование дефектов в системе с предоставлением всей имеющейся о нем информации, для более быстрого анализа и его устранения

специалистами службы технической поддержки. На рис. 3.8 представлена форма регистрации нового дефекта в системе.

Рисунок 3.8 – Форма регистрации дефекта

Как видно из представленной формы, для создания в системе нового дефекта имеется большое количество полей для заполнения, такие, как название, тип дефекта, уровень значимости, версия программы, сценарий получения, описание и т.п.

Общий вид оформленного дефекта в системе отслеживания ошибок, представлен на рисунках 3.9, 3.10 по частям.

| CR#71635 : 212224 - Invalid Mdu Subd Code on API Reconnect work order | | | | | |
|---|----------------------|-------------------|----------------------|-------------------------------|---------------|
| Create Date | 08 MAR 2015 14:17 | Status Date | 08 MAR 2015 14:29 | Close Date | |
| Author | Dan Braudo (dbraudo) | Changed By | Dan Braudo (dbraudo) | Closed By | |
| Project | wizupc2 | Release | 7.6.5.2 | Site | Poland |
| Status | QA | Approved To | | Reported By | Biskup, Jacek |
| From Customer | Y | Site reference | 212224 | Original release/version/item | |
| Topic | API | Effort | | Estimated Days to accomplish | |
| Customer Priority | Medium | Severity | 2 | Problem Types | Code,Code... |
| CR Type | Bug/TR | Core/Non Core | Y | Code Change | Y |
| Development Stage | Post production | Solution Test | N | Release Notes | N |
| On Site Env | | S&P | | Entered From Customer | Y |
| Rework | N | Finalize | N | Patch | |
| Cancel/Defer Reason | | Development Item | | Discovered stage | |
| Stage Responsible | | Stage Reviewer | | Sub Reason | |
| Undiscovered Reason | | Imp In Production | N | Imp In Production Date | |
| Site Status | | | | | |

Рисунок 3.9 – Форма представления дефекта в системе (часть 1)

Вверху представлено меню навигации по системе, ниже меню навигации по странице и основные поля: дата создания, автор, проект, статус, тип и т.д.

Все действия по смене ответственных лиц и статусов сохраняются в истории и отображаются в соответствующей таблице.

The screenshot shows a defect form with the following sections:

- Scenario:** On CRM -> Connect a new account on an address that has a non-blank MDU-subdivision-code. Visit the CR WO. From WS or API: Send a CR WO request for the account on the same address.
- Problem Description:** The request is rejected with "Invalid Mdu Sub Code" message.
- Reply CR# 71635:** A table showing the history of replies.

| Reply# | Reply Date | Author | Problems | Reply Type | Reply Title | Status |
|--------|-------------------|----------------------|----------|------------|-----------------------------------|--------|
| 1 | 08 MAR 2015 14:17 | Dan Braudo (dbraudo) | 1 | Other | CR Open by dbraudo | |
| 2 | 08 MAR 2015 14:18 | Dan Braudo (dbraudo) | 1 | Other | CR Approved by dbraudo to dbraudo | |
| 3 | 08 MAR 2015 14:29 | Dan Braudo (dbraudo) | 1 | Other | CR QA by dbraudo | |
- Modules:** A table showing modules not applied to release.

| Module | Version | Date | Owner |
|----------------------------|---------|-------------------|---------|
| wiz_fc_upkai_reconnect cob | 7652.6 | 08 MAR 2015 14:26 | dbraudo |
| wiz_upkai_wg_maint cob | 7652.97 | 08 MAR 2015 14:22 | dbraudo |

Рисунок 3.10 – Форма представления дефекта в системе (часть 2)

Здесь отображается история жизненного цикла, отмечено, кто и когда его зарегистрировал, кто выполнял. Ниже представлен список файлов, в которые были внесены изменения во время устранения данного дефекта, их версия, дата изменения и кем они были изменены.

На рисунке 3.11 приведена форма выполнения SQL-запросов, с помощью которой пользователь напрямую может обратиться к БД программного комплекса Wizard.

The screenshot shows the SQL execution interface with the following details:

- Wizard 7.9** header.
- SQL:** A text box containing the query: `SELECT * from all_tables`.
- Run** button.
- Results:** A table with columns: OWNER, TABLE_NAME, TABLESPACE_NAME, CLUSTER_NAME, BY_NAME, STATISTICS, PREPCT, INDEX, BRANCH, MAX, TRANSITIONAL, EXTINDEX, EXTINDEX, EXTINDEX, EXTINDEX, INCREASE, PRELIMIN, PRELIM, GROUPS, LOCKED, UNDO, ROWS, BLOCKS. The table lists various system tables and their properties.

Рисунок 3.11 – Форма выполнения SQL-запросов

Данная функция является полезной при первичном анализе проблемы для того, чтобы получить данные, необходимые для ее описания, не запуская само приложение или его модули.

3.5 Реализация компонентов системы отслеживания ошибок

Во время разработки компонентов системы были реализованы следующие задачи:

- 1) расширение функции смены статуса дефекта;
- 2) формирование отчета о количестве дефектов за промежуток времени.
- 3) создание хранилище документов.

На рис. 3.12 представлена форма смены статуса, которая позволяет передавать задание от одного разработчика другому, или переводить его на тестировщиков, менять статусы, например: «не воспроизводится», «устранено» и т.п.

The screenshot shows a web form titled "WIZARD 7-9" with a sub-header "Change CR Status CR# 71636". The main content area displays "CR# : 71636 Status: QA". Below this, there are several input fields: "Approve To:" with a dropdown menu showing "Anatolii Strizhov"; "Reply type:" with a dropdown menu showing "Bug exist"; "Reason:" with a dropdown menu showing "Other"; and "Other Reason:" with a text input field. At the bottom of the form, there is a "Send email comment:" label followed by a large text area. At the very bottom of the form, there are three buttons: "Approve", "Quick Close", and "Release".

Рисунок 3.12 – Форма смены статуса

Первое поле отвечает за ответственного специалиста, второе за текущий статус, третье и четвертое - за дополнительную информацию по дефекту.

Также было добавлено поле для рассылки сообщений о смене статуса ответственным лицам по электронной почте, позволяющее оповещать

специалистов и проводимых действиях и возможной необходимости их вмешательства в процесс устранения дефекта.

Важной опцией, добавленной в систему, является возможность создания отчета о количестве дефектов и их статусах за определенный период.

Для формирования отчета необходимо перейти на вкладку «Reports» и выбрать «System test statistic report», после чего откроется форма заполнения критериев для отчета, представленная на рис. 3.13.

The screenshot shows a web interface for generating a report. At the top, there is a navigation menu with the following items: 'Create CR', 'Update CR', 'Show CR', 'Reports', and 'Maintenance'. Below this is a section titled 'WIZARD 7-9'. The main content area is titled 'System test statistics' and contains several input fields:

- Topic:** A dropdown menu with 'All' selected.
- Project:** A dropdown menu with 'wizhot76' selected.
- Release:** A dropdown menu with '7.6.8' selected.
- Customer/Site:** A dropdown menu with 'HOT' selected.
- From Date:** Three dropdown menus for day (01), month (Jan), and year (2014).
- To Date:** Three dropdown menus for day (01), month (May), and year (2016).
- System Test:** Three radio buttons labeled 'All', 'Y', and 'N'. The 'N' option is selected.

Рисунок 3.13 – Форма заполнения критериев для отчета

Поле Topic отвечает за выбор компонента программного комплекса Wizard, а в поле Project выбирается проект, для которого будет сформирован отчет, там же выбирается его версия. После чего необходимо ввести промежуток времени, для которого сформировать отчет и нажать кнопку ОК, представленную на форме справа. В результате формируется и выводится на экран отчет о дефектах, представленный на рис. 3.14.

| System test statistics report | | | | | | | | | |
|--------------------------------------|----------|------|----------|-----|----------|--------|----------|-------|--------------|
| status/priority | Canceled | Open | Deferred | QA | Released | Closed | Approved | Total | Actual Total |
| Low | 1 | | | 1 | | 1 | | 3 | 1 |
| Medium | 199 | 7 | 2 | 565 | 1 | 11 | 190 | 975 | 765 |
| Total | 200 | 7 | 2 | 566 | 1 | 12 | 190 | 978 | 766 |

Learning Center | Home | The Container | ERP

www.NetCracker.com

©Version 2.4
Page updated 04/26/2016

Рисунок 3.14 – Отчет по устраненным дефектам

Представленные дефекты в отчете делятся по приоритетам. Можно увидеть статусы дефектов и их текущее количество.

Для получения подробного отчета со списком дефектов и датой их создания реализован специальный тип отчета, форма заполнения которого представлена на рис. 3.15.

[Create CR](#)
[Update CR](#)
[Show CR](#)
[Reports](#)
[Maintenance](#)

WIZARD 7-9

▼ TWR report

Topic

Project Release Customer/Site

From Date To Date

CR type

Release Notes Y N

System Test All Y N

Ascending/Descending Desc Asc

Рисунок 3.15 – Форма заполнения критериев для подробного отчета по дефектам

На рис. 3.16 представлен подробный отчет по дефектам за промежуток времени для определенного проекта.

| CR Number | Create Date | Project | Site | Release | CR Type | Problem | Release Notes | System Test | Scenario |
|-----------|-------------|----------|------|---------|---------|---------|---------------|-------------|--------------------------------|
| 72408 | 27-OCT-15 | dkswb | ONO | 100 | bfp | code | N | N | HOT.HOT_Wizard_IRB_Main - Defe |
| 71736 | 13-APR-15 | dthcons | cns | 211 | bug | cust | N | N | We have performance problem w |
| 71637 | 08-MAR-15 | dthdev | cvg | 531 | cds | data | N | N | Account#352776 recieved HCE pr |
| 71507 | 15-FEB-15 | dthslweb | czh | 710 | cfg | help | N | N | ליקוח 41447463 לליקוח עוד כ כר |
| 71408 | 21-JAN-15 | dthsuweb | dkf | 750 | fip | intf | N | N | add task with jobs- > click ac |
| 71316 | 04-JAN-15 | dthweb | dkm | 751 | gdd | parm | N | N | lock an account and run the WI |

Learning Center | Home | The Container | ERP
www.NetCracker.com

Рисунок 3.16 – Подробный отчет по дефектам

Данный отчет отображает все зарегистрированные дефекты в системе, основную информацию по ним и ссылку на страницу самого дефекта, позволяющую получить дополнительно информации, если требуется

В систему отслеживания ошибок был встроен модуль «Knowledge» - обучение, в котором хранятся основные обучающие документы по биллинговой системе Wizard, по языку программирования COBOL. Также ведется запись причин однообразных дефектов и возможные способы их устранения, множество дополнительной информации, создаваемой для помощи другим специалистам.

Специалисты могут добавлять свои документы, при наличие на это соответствующих прав доступа, также просматривать и другие. Переход на модуль обучения представлен на рис. 3.17.

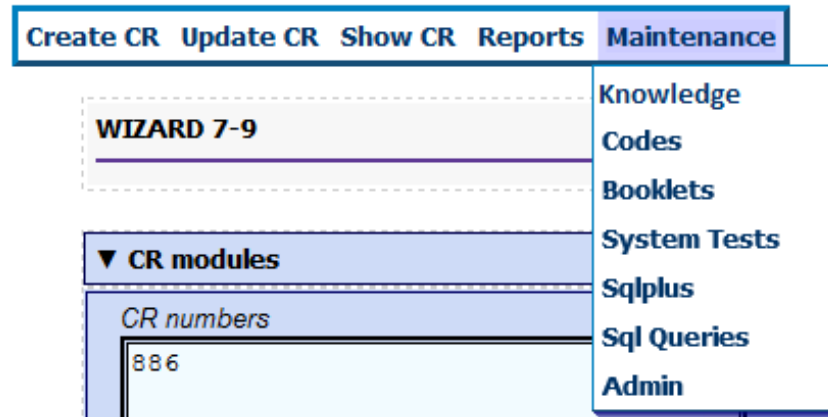


Рисунок 3.17 – Навигация на модуль «Знания»

На рис. 3.18 можно увидеть основную иерархию модуля обучения. В зависимости от типа создаются отдельные папки, в которые непосредственно помещаются сами файлы. Здесь могут храниться документы различных форматов: от всем знакомых MS Office файлов до специализированных файлов с расширением *c*, *cob*, *cbl*. Для более удобного хранения, сделана иерархия папок.

| Type | Name | Date | Author |
|--------|--|--------------------|--------------------|
| Folder | 0. Meeting minutes | 4/25/2013 4:33 PM | Andrey Funtov |
| Folder | 1. Fulfillment | 4/29/2013 5:20 PM | Natalia Simonova |
| Folder | 2. CRM | 4/29/2013 8:52 PM | Natalia Simonova |
| Folder | 3. RBM | 5/7/2013 1:40 PM | Natalia Simonova |
| Folder | 4. Assurance | 4/29/2013 5:20 PM | Natalia Simonova |
| Folder | 5. Resource Management | 5/8/2013 10:57 AM | Elena Markina |
| Folder | 6. Integration | 5/6/2013 12:14 PM | Elena Markina |
| Folder | 7. Activation | 5/13/2013 1:08 PM | Elena Markina |
| Folder | 8. Data Migration | 5/15/2013 5:08 PM | Elena Markina |
| Folder | 9. Product Catalog | 5/16/2013 5:56 PM | Elena Markina |
| File | 13106-DSG.DF | 11/30/2015 5:20 PM | Elena Markina |
| File | 13106-TRK-GRL-Usefull Documentation | 5/24/2013 2:36 PM | Elena Markina |
| File | 13106-TRM,GEN-Traceability Matrix | 8/15/2013 7:15 PM | Elena Markina |
| File | AT SPS Update 20151117 | 11/30/2015 2:41 AM | Elena Markina |
| File | CRM Design Drops | 9/25/2014 9:38 PM | Vladimir Chernov |
| File | D1 Design Drops (luckily saved) | 6/2/2015 4:35 PM | Vladimir Chernov |
| File | D1 Design Drops | 2/2/2015 7:14 PM | Vladimir Chernov |
| File | Design Documents - to be updated | 10/7/2014 11:10 AM | Irina Khorokhorina |
| File | Design Phase Deliverable Tracking | 7/16/2013 11:59 AM | Irina Khorokhorina |
| File | FF-PC Design Drops | 2/20/2015 4:12 PM | Irina Khorokhorina |
| File | 13106-TRN,INV-MPLS Exercises | 2/7/2014 9:42 AM | Irina Khorokhorina |
| File | 13106-TRN,INV-MPLS Lecture | 2/7/2014 9:35 AM | Irina Khorokhorina |
| File | 13106-TRN,INV-NC Environment Exercises | 1/30/2014 3:37 PM | Irina Khorokhorina |
| File | 13106-TRN,INV-NC Environment Lecture | 1/29/2014 1:37 PM | Irina Khorokhorina |
| File | 13106-TRN,INV-Resource Inventory Lecture | 1/29/2014 5:26 PM | Irina Khorokhorina |

Рисунок 3.18 – Визуально представление модуля «Знания»

Данный компонент представляет своего рода базу знаний службы поддержки, которая поможет в обучении новых специалистов, еще не знакомых

с языком программирования COBOL и системой Wizard. Для обучения у каждого новичка есть руководитель, который может предоставить доступ сотруднику к соответствующей документации, основываясь на его уровне деятельности в текущем проекте.

Занятые в проекте специалисты должны вносить информацию об устраненных ими дефектах и их решения для последующего облегчения процесса сопровождения и ускорения устранения проблем.

Одной из главных функций данного модуля является загрузка допустимых файлов на сервер. Загрузка осуществляется при помощи функции, программный код которой представлен ниже:

```
sub download {
    my ($filename, $count);
    open (DF, "<$datafile") || die "Cannot open $datafile: $!";
    flock (DF, 2);
    my @datalines = <DF>;
    flock (DF, 8);
    close (DF);
    chomp @datalines;
    foreach(@datalines) {
        if ( $q->param('id') == (split(/\|/))[0] ) {
            ($filename, $count) = (split(/\|/))[1,3];
            last; } }
    my ($ext) = $filename =~ /\.(w+)$/;
    $ext = lc($ext);
    my $file = $filedir.$filename;
    my $filesize = -s $file;
    print "Content-length: $filesize\n";
    print "Content-Disposition: attachment; filename=$filename\n";
    print "Content-type: $mimetypes{$ext}\n\n";
    open (F, "<$file")||die "Cannot open $filename: $!";
    binmode F;
    {
        local $/ = \4096; # sending in 4k blocks
        print while <F>;
    }
    close (F);
    $count++;
    open (DF, ">$datafile") || die "Cannot open $datafile: $!";
    flock (DF, 2);
    foreach (@datalines) {
        if ( $q->param('id') == (split(/\|/))[0] ) {
            my ($id,$filename,$desc) = (split(/\|/))[0..2];
            print DF "$id\|$filename\|$desc\|$count\n";
        } else {print DF "$_\n";}
    }
}
```

```

flock (DF, 8);
close (DF);
exit;
}

```

Функциональная блок-схема процесса загрузки файлов на сервер представлена на рис. 3.19.

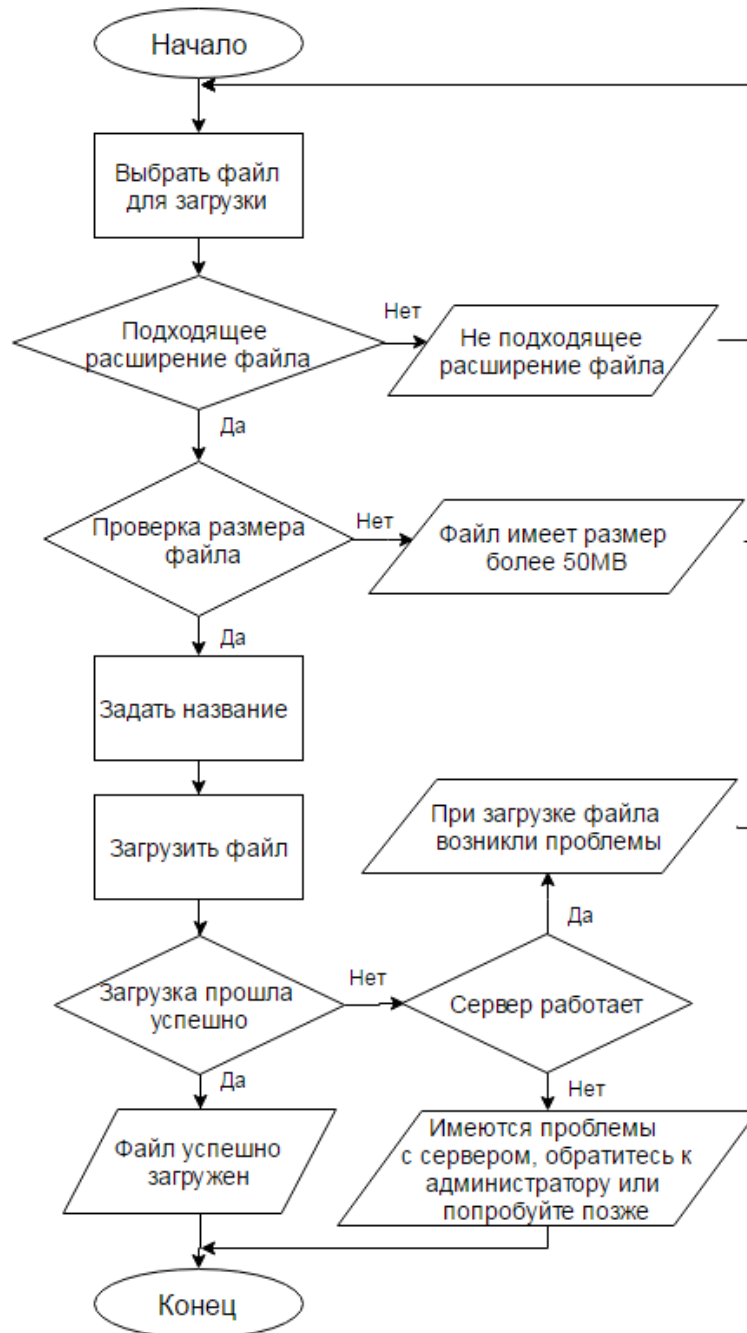


Рисунок 3.19 – Блок-схема процесса загрузки файлов на сервер

Пользователю необходимо нажать на кнопку «Загрузить файл», после чего появится дополнительная форма, в которой необходимо выбрать необходимый файл на рабочем компьютере.

После выбора файла осуществляется проверка расширения файла.

Файл может быть информативным или программным, система поддерживает большинство программных расширений, таких как с, cbl, sob, sql и т.д. Также, возможна загрузка файлов txt, doc, docx, xls, csv и т.д. Далее следует проверка размера файла, ограничение которого установлено в 50МБ.

Затем пользователь дает название загружаемому файлу, которое будет отображаться непосредственно на экране и нажимает «Загрузить файл».

Система проверяет возможность загрузки текущего файла и, если преград для этого нет, загружает файл в определенную директорию.

На рисунке 3.20 представлена страница модуля «Обучение», которой является частью модуля «Знания».

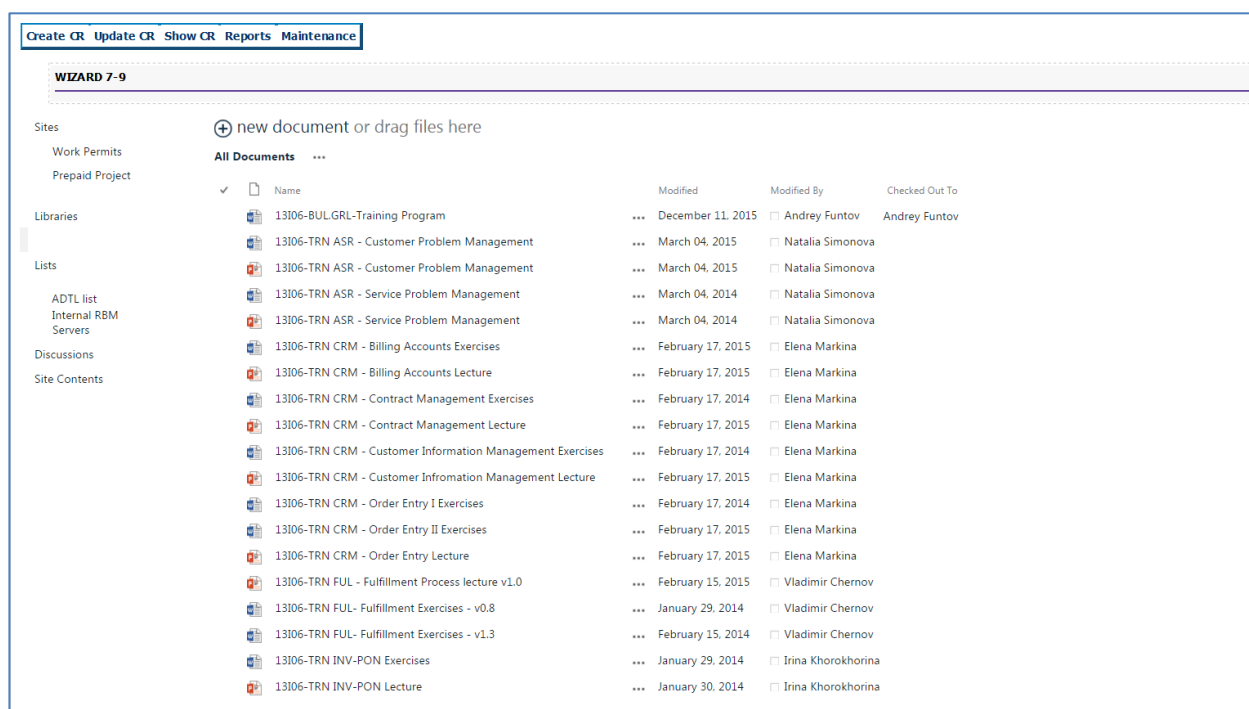


Рисунок 3.20 – Визуальное представление модуля «Обучение»

Модуль «Обучение» представляет собой своего рода хранилище документов, необходимых для обучения сотрудников. С помощью его функционала можно создавать документ, в котором отражается автор

документа и дата его создания. В модуле реализована система контроля, автоматически регистрирующая имя пользователя, редактирующего документ, в поле Checked Out To. Это означает блокировку документа до завершения редактирования, что помогает сохранить его целостность.

Функциональная блок-схема процесса проверки файла на текущее редактирование пользователем представлена на рис. 3.21.

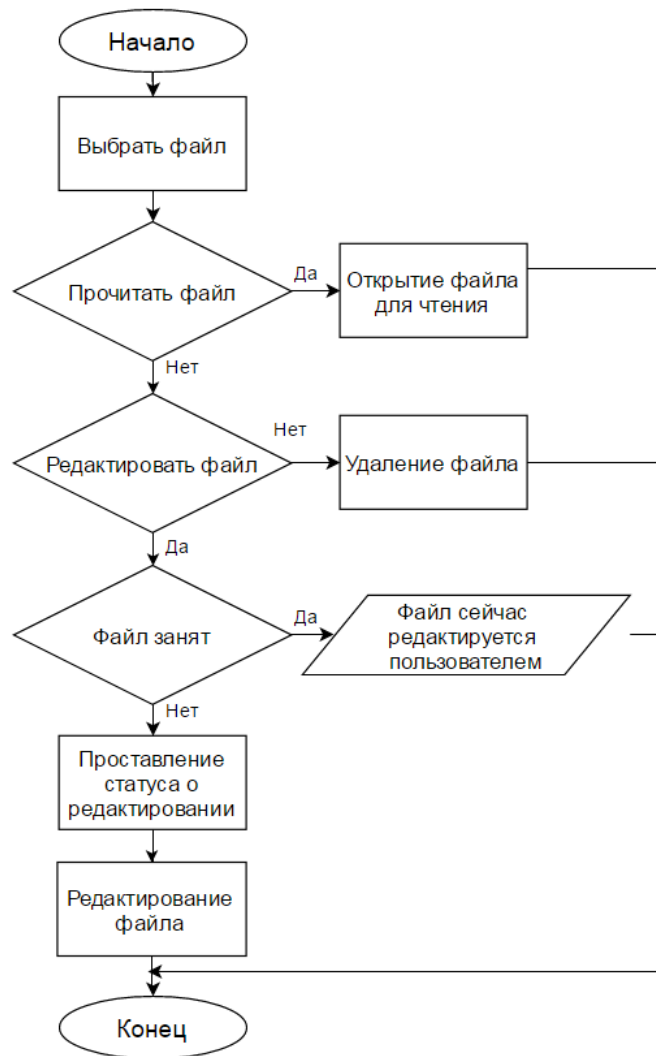


Рисунок 3.21 – Блок-схема процесса проверки файла на изменение

Проверка файла на текущее редактирование осуществляется с помощью функции, программный код которой представлен ниже:

```

sub checked_used {
    $leng = $ENV{'CONTENT_LENGTH'};
  
```

```

if ($leng > $cgi_lib'maxdata) { #
    &CgiDie("cgi-lib.pl: File used: $leng bytes\n");
}
$types = $ENV{'CONTENT_TYPE'};
$met = $ENV{'REQUEST_METHOD'};
if (!defined $met || $met eq " || $met eq 'GET' ||
    $met eq 'HEAD' ||
    $types eq 'applications') {
local ($keys, $value, $i);

# Read in text
if (!defined $met || $met eq ") {
    $inq = $ENV{'QUERY_STRING'};
    $comdflag = 1;
} elsif($met eq 'GET' || $met eq 'HEAD') {
    $inq = $ENV{'QUERY_STRING'};
} elsif ($met eq 'POST') {
    if (($get = read(STDIN, $inq, $leng) != $leng))
        {$errflags="Short Read: wanted $leng, got $get\n"};
} else {
    &CgiDie("cgi-lib.pl: Unknowan method of request: $met\n");
}
@inq = split(/[&;]/,$in);
push(@inq, @ARGV) if $comdflag;}

```

Таким образом, были разработаны компоненты для основной деятельности системы отслеживания ошибок CR Control и добавлен модуль знаний, позволяющий более эффективно организовывать обмен информацией между специалистами.

3.6 Тестирование и испытание «CR Control»

Тестирование - это процесс, который заключается в проверке соответствия информационной системы заявленным характеристикам и требованиям. Требования предъявляются к функциональным возможностям информационной системы, к функционированию информационной системы при определенных нагрузках, к удобству использования системы пользователем, к защищенности информационной системы от внешних, несанкционированных воздействий [32].

Целью тестирования является выявление ситуаций, при которых поведение программы является неправильным или не соответствующим спецификации.

Для категоризации видов тестов могут использоваться различные методологии, широко применяемые при тестировании веб-ориентированных приложений.

Тестирование статичного контента. Самый простой вид тестирования – «контентное». Это простой тест, который проверяет существование статичного, т. е. не изменяющегося элемента пользовательского интерфейса.

Тестировать контент нужно не всегда. Например, если содержание страницы вряд ли будет меняться со временем, то эффективнее проверять его вручную. Если же приложение, например, предполагает перемещение файлов, то контентное тестирование может быть оправдано.

Тестирование ссылок. Некорректные ссылки являются частым источником ошибок на веб сайтах. Тестирование подразумевает переход по каждой ссылке и проверку того, что выдана ожидаемая страница. Если статичные ссылки редко меняются, то достаточно будет ручного тестирования.

Тестирование функциональности. Эти тесты направлены на проверку определенного функционала приложения путем передачи ему пользовательского ввода и получения некоторого результата. Очень часто такой тест может состоять из переходов по нескольким страницам, ввода

данных в различные формы и поля, операций подтверждения и отмены действий и проверки результатов. Пользовательский ввод может осуществляться с помощью текстовых полей, выпадающих списков, чек-боксов и других форм ввода, которые поддерживает браузер.

Тестирование функциональности - чаще всего самый сложный вид тестирования, который вам придется автоматизировать, но он же и самый важный. Обычно такими тестами выполняется проверка процессов регистрации на сайте, входа в систему, операций над учетной записью пользователя, изменения настроек, сложных операций по получению данных, и тому подобного. В большинстве случаев функциональный тест копирует сценарии использования, определяющие функционал и устройство приложения.

Тестирование динамических элементов. Зачастую веб-элемент обладает уникальным идентификатором, который дает возможность найти этот элемент на странице. Впрочем, конкретная реализация, безусловно, зависит от работы веб-приложения.

Тестирование модулей системы отслеживания ошибок «CR Control» проводилось.

Система имеет архитектуру «клиент-сервер». Все данные хранятся на сервере, а также все вычисления, обработка поступающей и исходящей информации. Так как все действия происходит на сервере, основное тестирование проводилось на доступность сервера в то, или иное время, в тех или иных местах, пропускную способность сетевого канала для определенного количества пользователей. Максимальное количество одновременно работающих в системе пользователей было определено в 100 человек.

Тестирование с нескольких рабочих компьютеров, находящихся в удаленных друг от друга местах проблем с доступностью не выявило. Тестирование доступности сервера в разное время было проведено еще на стадии разработки АИС, проблем не было выявлено.

Также, было проведено тестирование, с использованием автотестов с использованием приложения Selenium, выполняющих одновременные действия по 8 потокам. Selenium - это инструмент для автоматизированного управления браузерами [48]. Наиболее популярной областью применения Selenium является автоматизация тестирования веб-приложений. Данное тестирование недостатков функционала не выявило.

Нагрузочное тестирование было проведено при помощи сервиса онлайн нагрузки Load Impact (рис. 3.22)

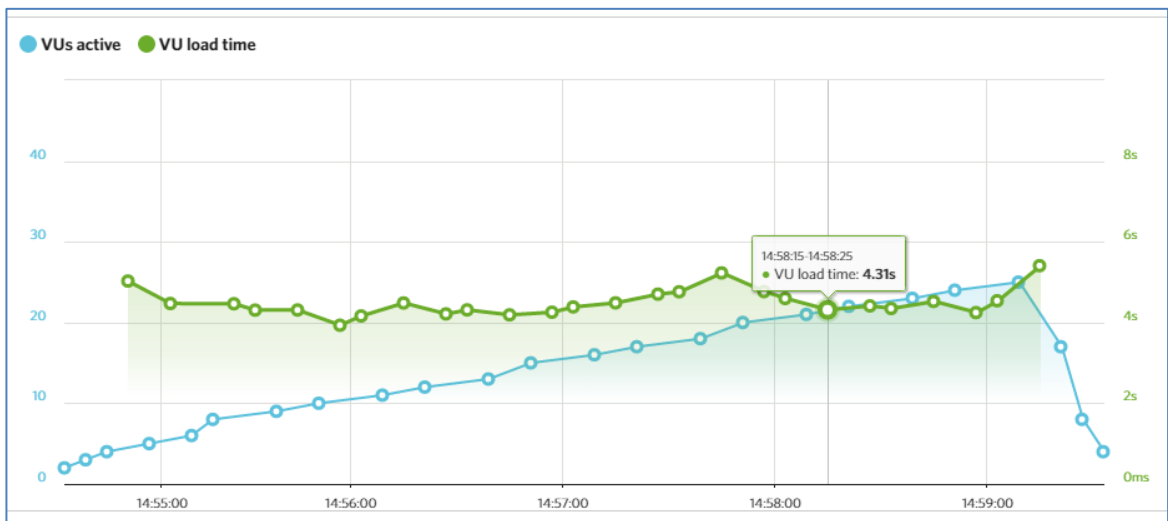


Рисунок 3.22 – Результат нагрузочного тестирования

Load Impact является ведущим сервисом онлайн нагрузки и тестирование производительности, который позволяет проверить веб-сайт, веб-приложение, мобильное приложение или API через Интернет [28].

Данный сервис предоставляет возможность имитировать одновременную деятельность 25 человек на ресурсе и выводит среднее время отклика системы пользователю.

Данное тестирование показало, что при максимально нагрузке, среднее время отклика составляет 4,5 секунды, что является нормой, а максимальное не превышает 6 секунд.

Вывод по главе

1) Система CR Control позволяет отправлять оповещения на электронную почту, добавлять, редактировать и хранить документацию, необходимую для осуществления полноценного сопровождения, составлять отчет о статусах дефектов за промежуток времени.

2) Разностороннее тестирование системы и отдельных ее модулей не выявило существенных недостатков, что подтверждает обеспечения функциональности системы на требуемом уровне.

ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНАЯ АПРОБАЦИЯ И ВНЕДРЕНИЕ КОНЦЕПЦИИ СОПРОВОЖДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

4.1 Внедрение модулей в систему CR Control

В рамках преддипломной практики в компании ООО «НетКрэкер», которая проходила с 28.03.2016 по 08.05.2016 г., были внедрены модули в систему отслеживания ошибок CR Control, в результате чего была улучшена эффективность работы специалистов отдела технической поддержки данной компании.

Основным критерием оценки являлось среднее время, затрачиваемое специалистом на устранение дефекта в программной комплексе Wizard, до внедрения модулей системы отслеживания ошибок и после.

Назначением внедренных модулей, является автоматизация процесса сопровождения усовершенствованием текущего функционала, добавление дополнительной отчетности и возможность системного хранения обучающих и регламентирующих документов. Последний модуль позволит сотрудникам отдела технической поддержки вести подробный учет собственных решений по проблемам, также пользоваться решениями других специалистов, и, главное, позволит ускорить процесс обучения вновь принятых на данное направление сотрудников.

4.2 Экспериментальная оценка качества сопровождения с использованием новых модулей системы отслеживания ошибок «CR Control»

Для оценки эффективности использования в процессе сопровождения разрабатываемых модулей системы отслеживания необходимо было провести эксперимент.

Целью эксперимента является апробация разработанных модулей системы для оценки эффективности их использования в процессе сопровождения.

На основании цели эксперимента были выделены следующие задачи:

- 1) выбор экспериментальных групп;
- 2) осуществление практического этапа внедрения и реализации;
- 3) анализ результатов работы с использованием разработанного информационного обеспечения.

Перед любым отделом технической поддержки ставится задача повышения качества сопровождения ПО с целью улучшения самого функционала сопровождаемых систем и уменьшение времени затрачиваемого на устранение дефектов.

В качестве критерия оценки улучшения процесса сопровождения было выбрано среднее время, затрачиваемого специалистом компании, на устранение дефекта.

Система отслеживания ошибок фиксирует время начала и время окончания работы над зарегистрированным дефектом, потраченное специалистом отдела технической поддержки. В связи с этим был проведен анализ на то, как влияет добавленный обучающий модуль на ускорение процесса устранения дефектов.

В качестве эксперимента была определена группа испытуемых из 12 человек, в которую входили как опытные специалисты, так и недавно принятые в компанию, для изучения влияния на обе группы лиц.

Полученные в результате эксперимента данные о сотрудниках отдела технической поддержки служат основанием для определения генеральной совокупности. В любом случае экспериментальная оценка полученной совокупности будет иметь какого-либо рода погрешность, в силу этого полученный результат может носить лишь рекомендательный характер, а не как

отверждение. Вышеизложенные предположения о признаках и свойствах совокупностей имеют название статистических гипотез.

Как указывает Г.В. Суходольский, «под статистической гипотезой обычно понимают формальное предположение о том, что сходство (или различие) некоторых параметрических или функциональных характеристик случайно или, наоборот, неслучайно» [6].

Основой для проверки статистической гипотезы является установка согласования статистических данных, полученных в результате эксперимента и выдвинутой гипотезы. Таким образом, статистическая гипотеза это научная гипотеза, допускающая статистическую проверку, а математическая статистика это научная дисциплина задачей которой является научно обоснованная проверка статистических гипотез.

При проверке статистических гипотез используются два понятия: так называемая нулевая гипотеза H_0 (или гипотеза о сходстве) и альтернативная гипотеза H_1 (или гипотеза о наличии различий). Таким образом, принятие альтернативной гипотезы H_1 говорит о наличии различий, а гипотезы H_0 об их отсутствии.

Уровнем значимости называется вероятность ошибочного отклонения нулевой гипотезы. Другими словами, уровень значимости - это вероятность ошибки первого рода при принятии решения. Исторически сложилось так, что в прикладных науках, использующих статистику, считается, что низшим уровнем статистической значимости является уровень $P = 0,05$; достаточным - уровень $P = 0,01$, а высшим - уровень $P = 0,001$. Поэтому в статистических таблицах, которые приводятся в приложении к учебникам по статистике, обычно даются табличные значения для уровней $P = 0,05$, $P = 0,01$ и $P = 0,001$.

Уровнем значимости принято называть вероятность ошибочного отклонения нулевой гипотезы, по-другому, уровень значимости - это вероятность ошибки первого рода при принятии решения. Исторически сложилось так, что в прикладных науках, использующих статистику, считается,

что низшим уровнем статистической значимости является уровень $P = 0,05$; достаточным - уровень $P = 0,01$, а высшим - уровень $P = 0,001$. В связи с этим, в статистических таблицах, которые обычно прилагаются к учебникам по статистике, чаще всего даются табличные значения для этих уровней.

Значения $0,05$ и $0,01$ называют стандартными уровнями статистической значимости. Во время статистического анализа, проводится эксперимент, для которого должны быть обозначены необходимые уровни значимости, исходя из задач и гипотез исследования. Можно заметить, что максимальное значение уровня статистической значимости, равняется $0,05$ — это означает, что допускается пять ошибок в выборке из ста элементов (случаев, испытуемых). Предполагается отсутствие большего числа ошибок. Значимость подобных ошибок очень велика. Цена таких ошибок будет слишком велика. На основании уровня значимости строится его графическое представление — ось значимости [6].

Предполагается, что выдвинутая гипотеза может быть правильной или неправильной, поэтому возникает необходимость ее проверки. Ввиду того, что сама проверка осуществляется статистическими методами, ее принято называть статистической. Неправильное решение во время проверки статистической гипотезы может быть принято всего в двух случаях, т. е. могут быть допущены ошибки двух родов. Сущность ошибки первого рода заключается в том, что будет отвергнута правильная гипотеза, ошибки второго рода - будет принята неправильная гипотеза.

Статистическим критерием (или просто критерием) называют случайную величину, которая служит для проверки нулевой гипотезы. Для проверки гипотезы по данным выборок вычисляют частные значения входящих в критерий величин, и таким образом получают частное (наблюдаемое) значение критерия. Статистический критерий представляет собой решающее правило, обеспечивающее математически обоснованное принятие истинной и отклонение ложной гипотезы. Статистические критерии определяют в

практической деятельности метод расчета определенного числа, которое обозначается как эмпирическое значение критерия, например, $t_{эмп}$ для t -критерия Стьюдента [6].

Основанием для подтверждения или опровержения гипотезы является соотношение эмпирического и критического значений критерия.

Количество степеней свободы — это количество значений в итоговом вычислении статистики, способных варьироваться. Иными словами, количество степеней свободы показывает размерность вектора из случайных величин, количество «свободных» величин, необходимых для того, чтобы полностью определить вектор [6].

Критерии различий принято условно разделять на две группы: параметрические и непараметрические критерии. Критерий различия называют параметрическим, если он основан на конкретном типе распределения генеральной совокупности (как правило, нормальном) или использует параметры этой совокупности (средние, дисперсии и т.д.). Критерий различия называют непараметрическим, если он не базируется на предположении о типе распределения генеральной совокупности и не использует параметры этой совокупности. Поэтому для непараметрических критериев предлагается также использовать такой термин как критерий, свободный от распределения. Для нахождения непараметрического критерия, полученные результаты эксперимента должны быть в интервальной или ранговой шкале.

Параметрические критерии имеют большую мощность в отличие от непараметрических, при нормальном распределении совокупности. Другими словами, их использование позволяет с большей достоверностью отвергать нулевую гипотезу, если она неверна. Исходя из этого, для выборок, взятых из нормально распределённых генеральных совокупностей, необходимо использовать параметрические критерии.

Как правило, в исследованиях чаще всего применяются такие параметрические критерии как: t -критерий Стьюдента, позволяющий

оценивать различия средних в двух выборках и F – критерий Фишера, который оценивает различия между двумя дисперсиями [6]. Они позволяют получить наиболее наглядное представление анализируемых совокупностей.

В данном исследовании, в качестве эксперимента, был проведен анализ статистических данных отдела технической поддержки, двенадцать специалистов которого представляли экспериментальную группу исследования.

В качестве критериев оценивания были выбраны следующие показатели:

- 1) среднее время, затраченное программистом на устранение дефекта, до внедрения нового модуля системы CR Control;
- 2) среднее время, затраченное программистом на устранение дефекта, после внедрения нового модуля системы CR Control.

В первом случае используются статистические данные отдела технической поддержки за последние четыре месяца 2015 года, во втором - за первые четыре месяца 2016 года.

Полученные данные отражены в таблице 4.1.

Таблица 4.1 Данные о среднем времени устранения дефектов

| № испытуемого | Среднее время до внедрения | Среднее время после внедрения |
|---------------|----------------------------|-------------------------------|
| 1 | 6,8 | 6,7 |
| 2 | 6,7 | 6,7 |
| 3 | 7,6 | 7,4 |
| 4 | 6,9 | 6,8 |
| 5 | 7,3 | 7,2 |
| 6 | 7,9 | 8,1 |
| 7 | 12,1 | 11,8 |
| 8 | 13,2 | 12,7 |
| 9 | 10,9 | 10,1 |
| 10 | 10,3 | 10,2 |
| 11 | 10,1 | 9,2 |
| 12 | 9,9 | 9,8 |

Приведенные в таблице данные отображены в виде гистограммы на рис. 4.1.

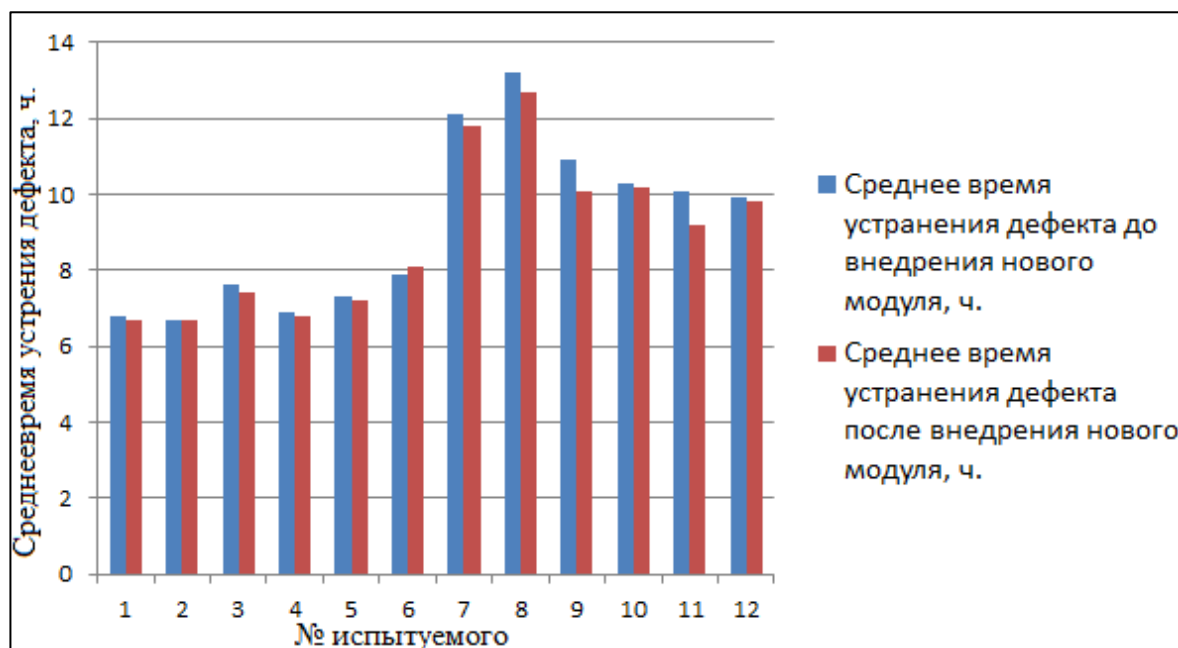


Рисунок 4.1 – Диаграмма представления статистических данных о среднем времени устранения дефекта

На приведенной диаграмме можно заметить, что для большинства испытуемых среднее время устранения дефекта меньше после внедрения нового модуля системы отслеживания ошибок CR Control, чем время до его внедрения. Исходя из этого, уже можно сделать вывод об эффективности внедрения нового модуля системы, так как основной целью внедрения подобного модуля, являлось сокращение времени устранения дефекта программистом отдела технической поддержки.

Для подтверждения эффективности работы отдела технической поддержки необходимо провести анализ полученных статистических данных. В качестве инструмента проведения подобного анализа была выбрана надстройка «Пакет анализа», используемая при проведении сложного статистического или инженерного анализа, входящая в программу для работы с электронными таблицами MS Excel. В качестве метода анализа, был использован двухвыборочный t-тест.

Наиболее частые случаи применения t -теста связаны с проверкой равенства средних значений в двух выборках.

По своей сути t -тест, представляет собой t -критерий Стьюдента. Критерий t Стьюдента направлен на оценку различий величин средних X и Y двух выборок X и Y . Главным достоинством критерия является широта его применения. Он может быть использован для сопоставления средних у связанных и несвязанных выборок, причем выборки могут быть не равны по величине.

Для приведенных средств, значение t вычисляется и отображается как « t -статистика» в выводимой таблице.

t -статистика обычно отображается по следующему общему принципу: в числителе случайная величина с нулевым математическим ожиданием (при выполнении нулевой гипотезы), а в знаменателе — выборочное стандартное отклонение этой случайной величины, получаемое как квадратный корень из несмещенной оценки дисперсии.

Двухвыборочный t -тест осуществляет проверку на равенство средних значений генеральной совокупности в каждой выборке [33]. Существует три вида подобного теста, характеризующиеся следующими условиями:

- дисперсии распределения равны;
- дисперсии совокупности не равны;
- представление выборок до и после эксперимента по одному и тому же признаку.

Парный двухвыборочный t -тест для средних. Парный тест используется, когда используется наглядная парность эксперимента в выборках, например, когда генеральная совокупность проверяется до и после эксперимента. Этот инструмент анализа применяется для проверки гипотезы о различии средних для двух выборок данных. Равенство дисперсий генеральных совокупностей, из которых были выбраны данные для эксперимента, в данном методе не предполагается [33].

Одним из вычисляемых результатов парного двухвыборочного t-теста для средних является общая дисперсия (объединенное измерение распределения данных вокруг среднего), определяемая по следующей формуле:

$$S^2 = \frac{n_1 * S_1^2 + n_2 * S_2^2}{n_1 + n_2 * 2} \quad (1)$$

Парный двухвыборочный t-тест для средних используется для проверки гипотезы о различии средних для двух выборок данных. Если не использовать «Пакет анализа», то алгоритм расчета включает следующие этапы:

- 1) найти разности парных вариантов;
- 2) вычислить среднюю разность;
- 3) вычислить дисперсию;
- 4) вычислить ошибку средней;
- 5) вычислить t статистику;
- 6) вычислить число степеней свободы;
- 7) определить достоверность различий.

Двухвыборочный t-тест для дисперсии. В основе данного инструмента лежит двухвыборочный t-тест Стьюдента, который используется для проверки гипотезы о равенстве средних для двух выборок. Подобная форма t-теста предполагает совпадение значений дисперсии генеральных совокупностей и называется гомоскедастическим t-тестом, где гомоскедастичность - свойство корреляции, состоящее в постоянстве условной дисперсии.

Двухвыборочный t-тест с различными дисперсиями. Данный инструмент анализа осуществляет двухвыборочный t-тест Стьюдента, который используется для проверки гипотезы о равенстве средних для двух выборок данных из разных генеральных совокупностей. Подобная форма t-теста подразумевает несовпадение дисперсий генеральных совокупностей и обычно называется гетероскедастическим t-тестом. Гетероскедастичность - неоднородность дисперсии [33].

Для определения тестовой величины t используется следующая формула.

$$r' = \frac{\bar{x} - \bar{y} - \Delta_0}{\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}}} \quad (2)$$

Если тестируется одна и та же генеральная совокупность, необходимо использовать парный тест.

Анализ t-тестов показал, что наиболее подходящим для данного исследования является «Парный двухвыборочный t-тест для средних», так как используется одна совокупности и имеются данные до и после эксперимента.

Используя процедуру «Парный двухвыборочный t-тест для средних», проверяем на уровень значимости 0.05 гипотезу H_1 : среднее время устранения дефекта программистом до внедрения новых модулей системы отслеживания ошибок CR Control, больше среднего времени, затрачиваемого на устранение дефекта после их внедрения. Основная гипотеза H_0 : среднее время устранения дефекта программистом до внедрения новых модулей системы отслеживания ошибок и после данного внедрения существенно не изменилось.

Данный инструмент находится в пакете «Анализ данных» программного комплекса MS Excel на вкладке «Данные».

Визуально представление пакета «Анализ данных» изображено на рис. 4.2.

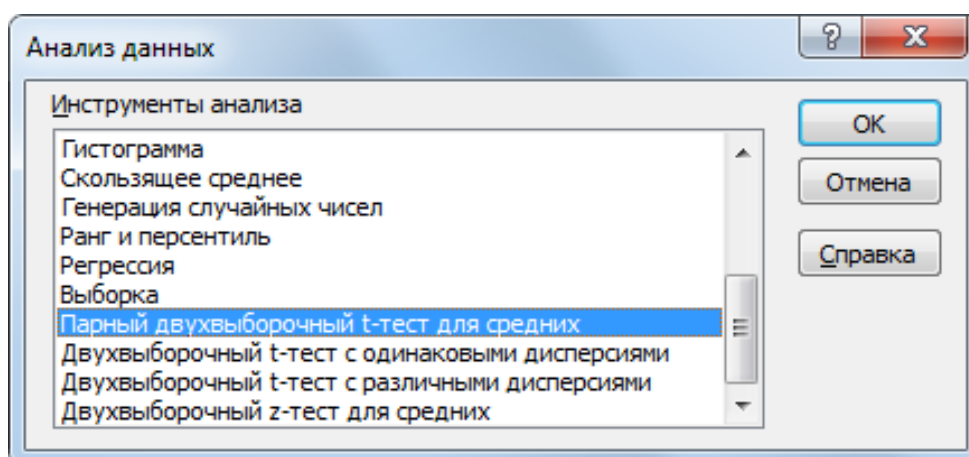


Рисунок 4.2 – Диалоговое окно «Анализ данных»

После нажатия кнопки «ОК» на диалоговом окне «Анализ данных», открывается диалоговое окно «Парный двухвыборочный t-тест для средних», в

котором необходимо указать ссылки на генеральные совокупности данных и выходной интервал, где будут выведены результаты расчётов (рис. 4.3).

| | A | B |
|----|------------------------|---------------------------|
| 1 | Время до внедрения, ч. | Время после внедрения, ч. |
| 2 | 6,8 | 6,7 |
| 3 | 6,7 | 6,7 |
| 4 | 7,6 | 7,4 |
| 5 | 6,9 | 6,8 |
| 6 | 7,3 | 7,2 |
| 7 | 7,9 | 8,1 |
| 8 | 12,1 | 11,8 |
| 9 | 13,2 | 12,7 |
| 10 | 10,9 | 10,1 |
| 11 | 10,3 | 10,2 |
| 12 | 10,1 | 9,2 |
| 13 | 9,9 | 9,8 |

Рисунок 4.3 – Заполнение входных параметров для расчета

На рис. 4.4 изображены результаты расчета парного двухвыборочного t-теста для средних.

| Парный двухвыборочный t-тест для средних | | |
|---|---------------------|---------------------|
| | <i>Переменная 1</i> | <i>Переменная 2</i> |
| Среднее | 9,141666667 | 8,891666667 |
| Дисперсия | 4,993560606 | 4,249924242 |
| Наблюдения | 12 | 12 |
| Корреляция Пирсона | 0,991707582 | |
| Гипотетическая разность средних | 0 | |
| df | 11 | |
| t-статистика | 2,655425023 | |
| P(T<=t) одностороннее | 0,011184341 | |
| t критическое одностороннее | 1,795884819 | |
| P(T<=t) двухстороннее | 0,022368681 | |
| t критическое двухстороннее | 2,20098516 | |

Рисунок 4.4 – Результат расчета парного двухвыборочного t-теста для средних

Основными значениями для анализа парного двухвыборочного t-теста для средних являются:

- 1) Среднее – среднее значение величин первой и второй совокупности.
- 2) Дисперсия – дисперсия величин первой и второй совокупности.
- 3) Наблюдения – количество значений в совокупности (количество испытуемых).
- 4) Корреляция Пирсона – коэффициент характеризующий существование линейной зависимости между двумя величинами.
- 5) Гипотетическая средняя разность – число, равное предполагаемой разности средних.
- 6) df – число степеней свободы.
- 7) t-статистика – расчетное значение t .
- 8) $P(T \leq t)$ одностороннее - вероятность того, что наблюдаемое значение t-статистики будет отрицательным, если предположить, что средние генеральной совокупности равны, при $t < 0$.

При $t \geq 0$ « $P(T \leq t)$ одностороннее» делает возможным наблюдение значения t-статистики, которое будет положительным.

9) « t критическое одностороннее» дает пороговое значение, так что вероятность наблюдения значения t-статистики большего или равного « t критическое одностороннее».

10) « $P(T \leq t)$ двустороннее» дает вероятность наблюдения значения t-статистики, по абсолютному значению большего, чем t .

11) « t критическое двустороннее» выдает пороговое значение, так что значение вероятности наблюдения значения t-статистики, по абсолютному значению большего, чем « t критическое двустороннее».

Согласно приведенным расчетам, среднее первой совокупности больше среднего второй, что подтверждает общее сокращение времени, затрачиваемого программистом на устранения дефекта.

Коэффициент корреляции Пирсона равен 0,99, что говорит о прямо пропорциональной и сильной связи между совокупностями. Но ввиду ограниченности выборки (12 человек) данный результат не может считаться объективным.

Также были определены минимальный и максимальный пороги допустимого значения t-статистики, которые равны 1,8 и 2,2 соответственно.

Значение 2,2 является критическим значением коэффициента t-Стьюдента для вероятности 0,05 согласно таблице критических значений данного критерия.

Полученное значение t-статистики 2,66 превышает максимальный порог в 2,2. Это позволяет сделать вывод о том, что среднее время устранения дефекта программистом до внедрения новых модулей существенно меньше этого времени после внедрения, и эти модули оказывают положительное влияние на эффективность работы сотрудников отдела технической поддержки.

Таким образом, по результатам t – теста нулевая гипотеза отклоняется и принимается альтернативная гипотеза.

Вывод по главе

1) В результате эксперимента с использованием системы отслеживания ошибок были собраны статистические данные за год, которые позволяют оценить эффективность деятельности сотрудников отдела технической поддержки до и после внедрения разрабатываемых модулей системы.

2) Анализ эффективности работы отдела технической поддержки при помощи парного двухвыборочного t-теста для средних подтвердил увеличение эффективности работы программистов при решении задач сопровождения благодаря использованию разработанных модулей системы отслеживания ошибок CR Control.

ЗАКЛЮЧЕНИЕ

В настоящее время серьезной проблемой является полное отсутствие или частичное использование систем отслеживания ошибок для осуществления полноценного процесса сопровождения программного обеспечения, написанного на устаревшем языке программирования COBOL.

Актуальность темы исследования обусловлена необходимостью разработки концепции по его сопровождению и модернизации.

В результате проведенного исследования достигнуты следующие результаты:

1. Разработана концепция сопровождения и модернизации ПО, реализованного на языке COBOL.

На основе анализа специальной литературы были выявлены недостатки известных концепций и методик сопровождения и модернизации ПО, разработанного на устаревших языках программирования, и предложена новая концепция, основанная на использовании в качестве средства, обеспечивающего высокую эффективность сопровождения и модернизации ПО, системы отслеживания ошибок.

2. Предложена методика сопровождения и модернизации программного обеспечения, реализованного на языке COBOL, разработанная на основе предлагаемой концепции.

Предлагаемая методика основана на построении единого информационного пространства, обеспечивающего совместное выполнение процесса отслеживания ошибок специалистом службы поддержки и информационной системой.

В качестве базовой системы отслеживания ошибок предлагается использовать систему CR Control.

3. Разработано информационно-технологическое обеспечение службы технической поддержки, обеспечивающее повышение эффективности

сопровождения и модернизации программного обеспечения, реализованного на языке COBOL.

Разработаны новые модули системы CR Control, обеспечивающие поддержку следующих задач:

- расширение функции смены статуса дефекта;
- формирование отчета о количестве дефектов за промежуток времени.
- создание хранилище документов.

Основные положения диссертационной работы были апробированы в компании ООО «НетКрэкер».

Анализ эффективности работы отдела технической поддержки был выполнен с помощью парного двухвыборочного t-теста для средних.

Тестирование и опытная эксплуатации модернизированной системы подтвердили, что использование внедренных модулей приводит к улучшению показателей качества и эффективности сопровождения ПО специалистами отдела технической поддержки благодаря использованию разработанных модулей системы отслеживания ошибок CR Control.

Теоретические результаты исследования опубликованы в двух статьях в журналах, индексируемых РИНЦ.

Практические результаты исследования подтверждены актом о внедрении модулей системы отслеживании ошибок CR Control в ООО «НетКрэкер».

Результаты исследования могут быть рекомендованы для использования компаниями-вендорами прикладного ПО для автоматизации процессов его сопровождения и модернизации.

СПИСОК ЛИТЕРАТУРЫ

Нормативно-правовые акты

1. ГОСТ Р. ИСО/МЭК 12207-99 //Информационная технология. Процессы жизненного цикла программных средств. – 1999.
2. ГОСТ Р. ИСО/МЭК 14764-2002 //Информационная технология. Сопровождение программных средств. – 2002.
3. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание документа.
4. Методология функционального моделирования IDEF0. Руководящий документ. Издание официальное. — М.: ИПК Издательство стандартов, 2000. — 75 с.

Учебники и учебные пособия

5. Дудина И. П. Рекомендации по выполнению выпускной квалификационной работы магистра по направлению подготовки "Прикладная информатика" : учеб.-метод. пособие / И. П. Дудина, О. М. Гущина, С. В. Мкртычев ; ТГУ ; Ин-т математики, физики и информационных технологий ; каф. "Информатика и вычислительная техника". - ТГУ. - Тольятти : ТГУ, 2013. - 51 с. - Библиогр.: с. 43-45. - Прил.: с. 46-50. - 24-04.
6. Ермолаев, О.Ю. Математическая статистика для психологов / О.Ю. Ермолаев. - М.: МПСИ: Флинта. - 2002. – С. 207-215.
7. Кричевец А. Н. Математическая статистика для психологов:[учебник для студ. вузов, обуч. по направл." Психология"] //Кричевец А.Н., Корнеев А. А., Рассказова Е. И.–М.: Академия. – 2012. – С. 50-52.
8. Липаев, В. В. Программная инженерия. Методологические основы [Текст]: Учеб. / В. В. Липаев; Гос. ун-т — Высшая школа экономики. — М.: ТЕИС, 2009. — 608 с. — 1000 экз. — ISBN 5-7598-0424-3 (в пер.).
9. Федосеев С. В. Современные проблемы прикладной информатики [Электронный ресурс]: учебное пособие/ Федосеев С.В.— М.: Евразийский открытый институт, 2011.— 272 с.

Периодические издания

10. Авдошин, С. М. Информатизация бизнеса. Управление рисками // Информационные технологии. – 2012. – №. 5. – С. 58-62.
11. Андреева М. М., Волков В. Р. Корреляционный анализ в социологических исследованиях // Вестник Казанского технологического университета. – 2013. – Т. 16. – №. 7. – С. 72-76.
12. Бажанов В. Ю., Кравченко Ю. А. Исследование статистических данных методами корреляционно-регрессионного анализа // Вестник Московского финансово-юридического университета МФЮА. – 2012. – №. 2. – С. 108-117.
13. Баженов Р. И., Корнилков А. П., Лопатин Д. К. Проектирование web-ориентированной информационной системы университета на основе клиент-серверных технологий // Актуальные проблемы гуманитарных и естественных наук. 2014. №4. С. 68-71.
14. Баженов Р. И., Лопатин Д. К. О применении современных технологий в разработке интеллектуальных систем // Журнал научных публикаций аспирантов и докторантов. 2014. № 3 (93). С. 263-264.
15. Брагина, Т. И., Табунщик Г. В. Сравнительный анализ итеративных моделей разработки программного обеспечения // Радиотехника. Информатика. – 2010. – №. 2. – С. 130-139.
16. Братищенко, В. В. Проектирование информационных систем. – Иркутск: Изд-во БГУЭП, 2004.
17. Брауде, Дж. Э. Технология разработки программного обеспечения. СПб.: Питер. – 2004.
18. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем. – М. : Финансы и статистика, 2012.
19. Гагарина, Л. Г. Технология разработки программного обеспечения. - М.: ИД ФОРУМ: НИЦ Инфра-М. – 2013.

20. Готман А. Ш. Теория вероятностей и математическая статистика // Международный журнал прикладных и фундаментальных исследований. – 2011. – №. 7. – С. 31-35.
21. Гусятников, В. Н., Безруков А. И. Стандартизация и разработка программных систем: учеб. пособие–М //М.: Финансы и статистика. – 2013.
22. Марьин, С. В. Интеллектуальная платформа управления композитными приложениями в распределенных вычислительных средах : дис. – СПб: СПбГУ ИТМО, 2010.
23. Мкртычев С.В. Объектно-структурное моделирование страховых информационных систем // [Вектор науки Тольяттинского государственного университета](#). – 2013. - №1(23). – С. 59-63.
24. Мосиенко, М. А. Автоматизированный перевод устаревших приложений на новые языки и платформы : диссертация кандидата физико-математических наук : Санкт-Петербург, 2006. - 122 с.
25. Орлов, С. А. Технологии разработки программного обеспечения: современный курс по программной инженерии:[по специальности" Программное обеспечение вычислительной техники и автоматизированных систем"]. – Издательский дом" Питер", 2012.
26. Поморова, О. В., Иванов А. В. Система нечеткого вывода для анализа, мониторинга и оценки рисков при разработке программного обеспечения. СПб.: Питер. – 2013.
27. Пустовалова Н. В., Авдеенко Т. В. Построение согласованной модели требований для процесса программной инженерии //Труды СПИИРАН. – 2015. – Т. 1. – №. 44. – С. 31-49.
28. Стрижов, А.Е. Оптимизация процесса сопровождения программного обеспечения, написанного на языке Cobol/ Студенческие дни науки в ТГУ. – Издательство ТГУ, 2015.

29. Стрижов, А.Е. Совмещение обязанностей ИТ-специалистов для осуществления полноценного сопровождения программного обеспечения // Научный альманах. – 2016. – С.175-178
30. Хан, В. Л. Сопровождение систем автоматизации программного обеспечения [Текст] / В. Л. Хан, А. В. Кизим // Молодой ученый. — 2011. — №5. Т.1. — С. 110-112.

Электронные ресурсы

31. Бодров О. А. Предметно-ориентированные экономические информационные системы [Электронный ресурс]: учебник/ Бодров О.А., Медведев Р.Е. — М.: Горячая линия - Телеком, 2013.— 244 с.
32. Галахов И. В., Лапыгин Д.В., Позин Б.А., Шкляева Н.А. Реализация стандарта ГОСТ Р ИСО/МЭК 14764-2002 «Сопровождение программных средств» на основе технологии RUP. [Электронный ресурс]. – Режим доступа: <http://www.citforum.ru/SE/project/gostiso/> (дата обращения: 11.12.2014).
33. Использование пакета анализа [Электронный ресурс]. – Режим доступа: <https://support.office.com> (дата обращения: 15.05.2016).
34. Организация научно-исследовательской работы студентов (магистров) [Электронный ресурс]: Учебное пособие / В.В. Кукушкина. - М.: НИЦ ИНФРА-М, 2014. - 265 с. – ISBN 978-5-16-004167-4.
35. Орлик С., Булуй Ю. Введение в программную инженерию и управление жизненным циклом ПО. Программная инженерия. Программные требования //Режим доступа: http://www.sorlik.ru/swebok/3-1-software_engineering_requirements.pdf. – 2005.
36. Система нагрузочного тестирования Load Impact [Электронный ресурс]. – Режим доступа: <https://loadimpact.com/> (дата обращения: 10.03.2016).
37. Система отслеживания ошибок BUGS - the Bug Genie [Электронный ресурс]. – Режим доступа: www.thebuggenie.com (дата обращения: 15.05.2015).
38. Система отслеживания ошибок Bugzilla [Электронный ресурс]. – Режим доступа: www.bugzilla.org (дата обращения: 15.05.2015).

39. Система отслеживания ошибок JIRA [Электронный ресурс]. – Режим доступа: www.atlassian.com/software/jira/ (дата обращения: 15.05.2015).
40. Система отслеживания ошибок Trac [Электронный ресурс]. – Режим доступа: trac.edgewall.org (дата обращения: 16.05.2015).
41. Система отслеживания ошибок Track Studio [Электронный ресурс]. – Режим доступа: www.trackstudio.ru (дата обращения: 16.05.2015).
42. Стандарт DFD [Электронный ресурс]. – Режим доступа: http://wiki.mvtom.ru/index.php/Стандарт_DFD (дата обращения: 4.05.2015).
43. Черников, Б. В. Управление качеством программного обеспечения //Качество. Инновации. Образование. – 2011. – №. 3. – С. 32-35.
44. Язык программирования КОБОЛ [Электронный ресурс]. – Режим доступа: <http://notes.sochi.org.ru/1333/> (дата обращения: 19.11.2014).
45. Обзор систем отслеживания ошибок [Электронный ресурс]. – Точка доступа: <http://www.teamlead.ru/pages/viewpage.action?pageId=15794279> (дата обращения: 21.05.2015).
46. Micro Focus [Электронный ресурс]. – Режим доступа: <http://www.microfocus.com/> (дата обращения: 19.11.2014).

Литература на иностранном языке

47. Avram H. D., Droz J. R. MARC II and COBOL //Information Technology and Libraries. – 2012. – Т. 1. – №. 4. – С. 261-272.
48. Bowers A., Bell J. Automated testing with Selenium and Cucumber //Haettu. – 2013. – Т. 14. – С. 2015.
49. Dijkstra, E. W. Trip Report EW Dijkstra, ECI-Conference 9–12 August 1976, Amsterdam //Selected Writings on Computing: A personal Perspective. – Springer New York, 1982. – С. 225-229.
50. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: Institute of Electrical and Electronics Engineers, 1990.

51. Lancaster, G. T. Programming in COBOL: Library of Computer Education. – Elsevier, 2014.
52. Leela G. Bug Tracking System //International Journal of Computer (IJC). – 2015. – Т. 18. – №. 1.
53. Miller, E. F. A service concept for software auditing //Proc. Workshop Computer Auditing, San Francisco (January 1976). – 1976.
54. Nickerson, R. C. Fundamentals of structured COBOL. – Addison-Wesley Longman Publishing Co., Inc., 1997.
55. Rodriguez, J. M. et al. Bottom-Up and Top-Down Cobol System Migration to Web Services //Internet Computing, IEEE. – 2013. – Т. 17. – №. 2. – С. 44-51.
56. Rouvoy, R. et al. Music: Middleware support for self-adaptation in ubiquitous and service-oriented environments //Software engineering for self-adaptive systems. – Springer Berlin Heidelberg, 2009. – С. 164-182.
57. Stap, G. Cobol Data Flow Restructuring : дис. – Master's thesis, Vrije Universiteit Amsterdam & University of Amsterdam, 2005.
58. Starkovich, D. P. et al. Automatic COBOL working storage to open/OLTP view conversion : пат. 8041836 США. – 2011.
59. Starix, A. Cobol Production : дис. – Master's programmings, Vrije Universiteit Amsterdam & University of Amsterdam, 2011.
60. Zimmermann T. et al. Improving bug tracking systems //ICSE Companion. – 2009. – С. 247-250.

ПРИЛОЖЕНИЕ 1

Представление программного кода модулей системы отслеживания
ошибок CR Control

```
#####
# MAIN CONTROL PROGRAM
#####

# Getting everything which is a GET request
if ( $ENV{'REQUEST_METHOD'} eq 'GET' ) {
    my          $uri          =          $ENV{'REDIRECT_DOCUMENT_URI'}?
$ENV{'REDIRECT_DOCUMENT_URI'} : $ENV{'DOCUMENT_URI'};
    if ( $uri ne $shtml ) {
        &error('Direct access forbidden!');
        exit;
    }
    if ( !$diracc && $ENV{'HTTP_REFERER'} !~ /$dlloc/ && $q->param('id')) {
        &error('Direct access forbidden!');
        exit;
    }
    if ( $q->url_param('id') ) {
        &view_info;
    } else {
        &welcome;
    }
}

# Getting everything which is a POST request
if ( $ENV{'REQUEST_METHOD'} eq 'POST' ) {
    if ( $ENV{'HTTP_REFERER'} !~ /$dlloc/ && $dirdwl ) {
        print $q->header();
        print          "<html><head><title>ERROR:          Direct          download
forbidden!</title></head>\n";
        print "<body>",&error('Direct download forbidden!'),"</body></html>\n";
        exit;
    } else {
        &download;
    }
}

exit;

#####
# SUBROUTINES
#####

# welcome displays all provided files from your $dbfile
sub welcome {
    print $q->header();
}
```

```

print "<div align='center'>";
print "<table border=$stab_border width=$stab_width cellspacing=0>";
print
bgcolor=$th_col<th>ID</th><th>Filename</th><th>Description</th><th>Count</th></tr>";
open (DF, "<$datafile") || die "Cannot open $datafile: $!";
flock (DF, 2);
my @datalines = <DF>;
flock (DF, 8);
close (DF);
chomp @datalines;
foreach (sort{(split /\|,$a)[0] cmp (split /\|,$b)[0]} @datalines) {
    my ($id, $filename, $desc, $count) = split /\|/;
    my $link = $dlloc."?id=$id";
    if (!$count) {
        $count = '0';
    }
    print      "<tr><td      align='center'><a      href=\"$link\">$id</a></td><td><a
href=\"$link\">$filename</a></td><td>$desc</td><td align='center'>$count</td></tr>";
}

print "</table></div>";
&footer;
exit;
} # end welcome

```

```

# view_info shows the description file if enabled plus
# additional information like filesize, application type
# and created finally the post request for downloading the
# file

```

```

sub view_info {
    print $q->header();
    my ($filename, $filesize, $description, $count);
    if ($info) {
        my $infofile = $datadir.$q->url_param('id').$info_prefix;
        open (IF, "<$infofile") || die "Cannot open $infofile: $!";
        flock (IF, 2);
        print while <IF>;
        flock (IF, 8);
        close (IF);
    }
    open (DF, "$datafile") || die "Cannot open $datafile: $!";
    flock (DF, 2);
    my @datalines = <DF>;
    flock (DF, 8);
    chomp @datalines;
    foreach (@datalines) {
        if ( $q->url_param('id') == (split(/\|/))[0] ) {
            $filename = (split(/\|/))[1];
            my $tempfile = $filedir.$filename;
            $description = (split(/\|/))[2];

```

```

        $filesize = -s $tempfile;
        last;
    }
}
print "<div align='center'><table border=$tab_border>";
print "<tr><td bgcolor=$td_col>Description: <b>$description</b></td></tr>\n";
print "<tr><td bgcolor=$td_col>Filename: <b>$filename</b></td></tr>\n";
print "<tr><td bgcolor=$td_col>Filesize: <b>$filesize</b> bytes</td></tr>\n";
my ($ext) = $filename =~ /\.(\w+)$/;
print "<tr><td bgcolor=$td_col>Type: $mimetypes{$ext}</td></tr>\n";
# print warning if user comes through a proxyserver
# this might make problems
if (($ENV{'HTTP_VIA'} || $ENV{'HTTP_X_FORWARDED_FOR'}) && $check_proxy){
    print "<tr><td><table border=1 cellspacing=0 cellpadding=2>\n";
    print "<tr><th align='center' colspan=2'><font size='2' face='Arial,Helvetica,sans-
serif'>! DOWNLOAD DISABLED !</font></th></tr>\n";
    print "<tr><td align='center' colspan=2'><font size='1' face='Arial,Helvetica,sans-
serif'>Sorry, but because of you're coming to this site via a proxy the download is disabled.<br>".
        "The reason is that most proxies would deliver rather a HTML-file than the
desired file<br>which description you're looking at.<br>".
        "To get the file please disable your proxy and try again.";
    print "</table></td></tr>\n";
} else {
    if ($proxy_warn) {
        $shtml =~ s/^\/g; $shtml =~ s/^(.+\.)(.+)$/{1}htm/;
        print "<tr><td align='center'><table border=1 cellspacing=0 cellpadding=2
width='300'>\n";
        print "<tr><td align='left'><font size='1' face='Arial,Helvetica,sans-serif'>".
            "<u>Attention proxy-users!</u> Some proxies rather try to get the
currently displayed HTML than pipe the desired file through. This could be but must not happen. If
you see something appearing in the download window looking like \"$shtml\" then switch of your
proxy and try again.".
            "</font></td></tr></table></td></tr>\n";
    }
    print "<tr><td align='center' bgcolor=$td_col>";
    print "<form action=\"${ENV{'SCRIPT_NAME'}}\" method='POST'>\n";
    print "    <input type=hidden name='id' value=\"\",$q->url_param('id'),\"><div
align='center'><input type=submit value='Download'>\n";
    print "</form></td></tr>\n";
}
print "</table>\n";
print "<a href=\"$dlloc\">Back do download</a><br>\n";
print "<a href=\"$home\">Back to home</a><br>\n";
&footer;
exit;
}

sub download {
    my ($filename, $count);
    open (DF, "<$datafile") || die "Cannot open $datafile: $!";
    flock (DF, 2);
}

```

```

my @datalines = <DF>;
flock (DF, 8);
close (DF);
chomp @datalines;
foreach(@datalines) {
    if ( $q->param('id') == (split(/\|/))[0] ) {
        ($filename, $count) = (split(/\|/))[1,3];
        last;
    }
}
my ($ext) = $filename =~ /\.(\w+)/;
$ext = lc($ext);
my $file = $filedir.$filename;
my $filesize = -s $file;
print "Content-length: $filesize\n";
print "Content-Disposition: attachment; filename=$filename\n";
print "Content-type: $mimetypes{$ext}\n\n";
open (F, "<$file") || die "Cannot open $filename: $!";
# Thanks to Christian Lackas for this ;)
binmode F;
{
    local $/ = \4096; # sending in 4k blocks
    print while <F>;
}
close (F);

$count++;
open (DF, ">$datafile") || die "Cannot open $datafile: $!";
flock (DF, 2);
foreach (@datalines) {
    if ( $q->param('id') == (split(/\|/))[0] ) {
        my ($id,$filename,$desc) = (split(/\|/))[0..2];
        print DF "$id|$filename|$desc|$count\n";
    } else {
        print DF "$_\n";
    }
}
flock (DF, 8);
close (DF);
exit;
}

sub footer {
    print "<hr align='center' size=1 width=350>";
    print "<div align='center'><font size=1 face='Arial, Helvetica, sans-serif'><a
href='http://www.the-piper.net/'>Doc Piper's Downloader v.$version \&copy; 2000, 2001 Tom
'Doc Piper' Klinger</font></a></div>";
    print "<hr align='center' size=1 width=350>";
}

sub error {

```

```

    my $error = $_[0];
    print $q->header(), "<h3>ERROR: $error</h3>";
    print "<font size=2 face=\"Arial, Helvetica, sans-serif\">Please use <a href=\"\${dlloc}\">this
link</a> to return ...</font>";
    return;
}

if (length ($ENV{'QUERY_STRING'}) > 0){
    $buffer = $ENV{'QUERY_STRING'};
    #seperate all name/value pairs at the &
    @pairs = split(/&/, $buffer);
    foreach $pair (@pairs){
        #seperate each name/value pair at =
        ($name, $value) = split(/=/, $pair);
        $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
        $in{$name} = $value;
    }
}

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
    @pairs = split(/&/,$buffer);
    foreach $pair(@pairs) {
        ($name, $value) = split(/=/,$pair);
        $value =~ tr/+//;
        $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
        $FORM{$name} = $value;
    }

    &getdate;
    $function = $in{'function'};
    if ($function eq 'admin') {

        &passhtml;
    }

    elsif ($function eq 'password') {

        $passin = $FORM{'passin'};
        $passin = crypt ($passin, Cd);

        if ($passin ne $password) {
            print "the password was incorrect. Try again.";
        }

        else {
            print "Content-type: text/html\n\n";
        }
    }
print<<"EOF";

<html>
<head>
<title>C22download - Admin</title>

```



```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#0000CC" vlink="#CCCCCC" text="#FFFFFF" link="#CCCCCC"
alink="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr bgcolor="#3333FF">
    <td width="50%"><font face="Arial, Helvetica, sans-serif"><b><font
size="+4">c22download</font></b></font></td>
    <td width="50%">
      <div align="right"><font face="Arial, Helvetica, sans-serif"><a href="http://www.cache-
22.co.uk">Cache-22.co.uk</a><br>
      Version number - 1.01</font></div>
    </td>
  </tr>
</table>
<div align="left">
  <p>&nbsp;</p>
  <form method="post" action="c22download.cgi?function=post">
    <p><font face="Arial, Helvetica, sans-serif">Add
    <input type="radio" name="action" value="add" checked>
    / Remove
    <input type="radio" name="action" value="remove">
    / Modify
    <input type="radio" name="action" value="modify">
    </font></p>
    <p>
    <input type="text" name="name" value="Name">
    <input type="text" name="url" value="URL">
    </p>
    <p><font face="Arial, Helvetica, sans-serif">Change to:</font></p>
    <p>
    <input type="text" name="chname" value="Name">
    <input type="text" name="churl" value="URL">
    </p>
    <p>
    <input type="submit" name="Submit" value="Submit">
    </p>
  </form>
  <p><font face="Arial, Helvetica, sans-serif">Current Stats.</font></p>
  <table width="80%" border="1" cellspacing="1" cellpadding="1">
    <tr>
      <td><b><font face="Arial, Helvetica, sans-serif">File</font></b></td>
      <td><b><font face="Arial, Helvetica, sans-serif">URL</font></b></td>
      <td><b><font face="Arial, Helvetica, sans-serif">Times downloaded</font></b></td>
      <td><b><font face="Arial, Helvetica, sans-serif">Last downloaded</font></b></td>
    </tr>
  </table>

```

EOF

```

open (DATA, "c22download/download.txt") || die "cant open download.txt\n";
@data = <DATA>;

```

```

close (DATA);

$i = 0;
while ($i <=#data) {
    $data[$i] =~ s/--//g;
    $data[$i+1] =~ s/--//g;
    $data[$i+2] =~ s/--//g;
    $data[$i+3] =~ s/--//g;
print<<"EOF";
    <tr>
    <td><font face="Arial, Helvetica, sans-serif">$data[$i]</font></td>
    <td><font face="Arial, Helvetica, sans-serif">$data[$i+1]</font></td>
    <td><font face="Arial, Helvetica, sans-serif">$data[$i+2]</font></td>
    <td><font face="Arial, Helvetica, sans-serif">$data[$i+3]</font></td>
    </tr>

EOF
    $i = $i + 4;
    }

print<<"EOF";
    </table>
    <p>&nbsp;</p>
</div>
</body>
</html>
EOF
}
}
elseif ($function eq 'post'){

print "Content-type: text/html\n\n";
    &getdate;
    $action = $FORM{'action'};
    $url = $FORM{'url'};
    $name = $FORM{'name'};
    $churl = $FORM{'churl'};
    $chname = $FORM{'chname'};

    if ($action eq 'add') {
        open (DATA, ">>c22download/download.txt") || die "cant open download.txt\n";
        print DATA "--$name--\n";
        print DATA "$url\n";
        print DATA "0\n";
        print DATA "\n";
        close (DATA);
        &addhtml;
    }
    elseif ($action eq 'remove') {

        open (DATA, "c22download/download.txt") || die "cant open download.txt\n";

```

```

    @data = <DATA>;
    close (DATA);

    for ($i = 0; $i <=#data; $i++) {
    if ($data[$i] =~ /--$name--/i) {
        splice @data, $i, 4;
        last;
    }
    }
    open (DATA, ">c22download/download.txt") || die "cant open download.txt\n";
    print DATA @data;
    close (DATA);

    &remhtml;
}

elsif ($action eq 'modify') {

    open (DATA, "c22download/download.txt") || die "cant open download.txt\n";
    @data = <DATA>;
    close (DATA);

    for ($i = 0; $i <=#data; $i++) {
    if ($data[$i] =~ /--$name--/i) {

        $data[$i] = "--$chname--\n";
        $data[$i+1] = "$churl\n";
    }
    }
    open (DATA, ">c22download/download.txt") || die "cant open download.txt\n";
    print DATA @data;
    close (DATA);
    &modhtml;
}
}

elsif ($function eq 'download'){
    $download = $FORM{'download'};
    $name = $FORM{'name'};
    $email = $FORM{'email'};

    open (DATA, "c22download/download.txt") || die "cant open download.txt\n";
    @data = <DATA>;
    close (DATA);

    for ($i = 0; $i <=#data; $i++) {

        if (" $download" == "$i") {
            $count = $data[$i+2];
            $count = $count + 1;
            $data[$i+2] = "$count\n";

```

```

    $data[$i+3] = "$date\n";

    open (DATA, ">c22download/download.txt") || die "cant open download.txt\n";
    print DATA @data;
    close (DATA);

    open (LOGS, ">>c22download/logs.txt") || die "cant open logs.txt\n";
    print LOGS "date = $date name = $name e-mail = $email file = $data[$i]\n";
    close (LOGS);
    print "location: $data[$i+1]\n\n";
    }
}

else {

print "Content-type: text/html\n\n";

    open (DATA, "c22download/download.txt") || die "cant open download.txt\n";
    @data = <DATA>;
    close (DATA);
    print<form method="post" action="$cgibinpath/c22download.cgi?function=download">";

    if ($iname == 1) {
print<<"EOF";
        <p>
        <input type="text" name="name" value="your name">
        </p>
EOF
    }
    elsif ($iname == 2) {

print<<'EOF';

<script language="JavaScript">
<!--
function MM_findObj(n, d) { //v3.0
    var p,i,x; if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
    if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document); return x;
}

function MM_validateForm() { //v3.0
    var i,p,q,nm,test,num,min,max,errors=",args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=MM_findObj(args[i]);
    if (val) { nm=val.name; if ((val=val.value)!="") {
        if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
            if (p<1 || p==(val.length-1)) errors+= '- '+nm+' must contain an e-mail address.\n';
        } else if (test!='R') { num = parseFloat(val);
            if (val!="+num) errors+= '- '+nm+' must contain a number.\n';

```

```

    if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
        min=test.substring(8,p); max=test.substring(p+1);
        if (num<min || max<num) errors+="- "+nm+" must contain a number between '+min+' and
'+max+'.\n';
    } } else if (test.charAt(0) == 'R') errors += "- "+nm+" is required.\n"; }
} if (errors) alert("The following error(s) occurred:\n"+errors);
document.MM_returnValue = (errors == "");
}
//-->
</script>
EOF
print<<"EOF";
    <p>
    <input type="text" name="name" value="your name">
    </p>

EOF
    $namereq = "name','R','";
    $validate = "onClick=\"MM_validateForm($namereq$emailreq);return
document.MM_returnValue\"";
    }

    if ($incemail == 1) {

print<<"EOF";

    <p>
    <input type="text" name="email" value="email address">
    </p>
    <p>
EOF    }

    elseif ($incemail == 2) {

print<<'EOF';

<script language="JavaScript">
<!--
function MM_findObj(n, d) { //v3.0
    var p,i,x; if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
    if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document); return x;
}

function MM_validateForm() { //v3.0
    var i,p,q,nm,test,num,min,max,errors="",args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=MM_findObj(args[i]);
        if (val) { nm=val.name; if ((val=val.value)!="") {
            if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
                if (p<1 || p==(val.length-1)) errors+="- "+nm+" must contain an e-mail address.\n";

```

```

} else if (test!='R') { num = parseFloat(val);
if (val!="+num) errors+="- '+nm+' must contain a number.\n";
if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
min=test.substring(8,p); max=test.substring(p+1);
if (num<min || max<num) errors+="- '+nm+' must contain a number between '+min+' and
'+max+'.\n";
} } } else if (test.charAt(0) == 'R') errors += '- '+nm+' is required.\n'; }
} if (errors) alert("The following error(s) occurred:\n'+errors);
document.MM_returnValue = (errors == "");
}
//-->
</script>

```

EOF

```
$smailreq = "email","RisEmail";
```

```
$validate = "onClick=\"MM_validateForm($namereq$smailreq);return
document.MM_returnValue\"";
```

```
print<<"EOF";
```

```

<p>
<input type="text" name="email" value="email address">
</p>
<p>

```

EOF }

```

print '<select name="download">';
while ($i <=$#data) {
$file = $data[$i];
$file =~ s/--//g;
$file =~ s/.zip//g;
print "<option value=\"\$i\">$file</option>";
$i = $i + 4;
}
print<<"EOF";

</select>
</p>
<p>
<input type="submit" name="Submit" value="Download" $validate>
</p>
</form>
EOF
}

```

```

sub getdate{
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time + (3600*$timeoffset));
$mon = $mon + 1;
$year = $year + 1900;
}

```

```

$date = "$hour:$min - $mday/$mon/$year";
}
sub passhtml {
print<<"EOF";
<html>
<head>
<title>C22download - Admin Password Required</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#0000CC" vlink="#CCCCCC" text="#FFFFFF" link="#CCCCCC"
alink="#FFFFFF">
<div align="left">
<table width="100%" border="0" cellspacing="0" cellpadding="00" align="center"
height="90%">
<tr align="center" valign="middle">
<td>
<td>
<form method="post" action="c22download.cgi?function=password" name="email">
<table width="200" border="1" cellspacing="1" cellpadding="1" bgcolor="#99FFFF"
bordercolor="#000000">
<tr bgcolor="#6666FF">
<td bordercolor="#99FFFF" colspan="2">
<div align="center"><b><font face="Arial, Helvetica, sans-serif">Enter
admin password</font></b></div>
</td>
</tr>
<tr>
<td bordercolor="#99FFFF" colspan="2">
<div align="center">
<input type="password" name="passin">
<input type="submit" name="Submit" value="Go">
</div>
</td>
</tr>
</table>
</form>
</td>
</tr>
</table>
<p>&nbsp;</p>
</div>
</body>
</html>

EOF
}
sub addhtml {
print<<"EOF";
<html>
<head>
<title>C22download - infomation added</title>

```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
```

```
<body bgcolor="#0000CC" vlink="#CCCCCC" text="#FFFFFF" link="#CCCCCC"
alink="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr bgcolor="#3333FF">
    <td width="50%"><font face="Arial, Helvetica, sans-serif"><b><font
size="+4">c22download</font></b></font></td>
    <td width="50%">
      <div align="right"><font face="Arial, Helvetica, sans-serif"><a href="http://www.cache-
22.co.uk">Cache-22.co.uk</a><br>
      Version number - 1.01</font></div>
    </td>
  </tr>
</table>
<div align="left">
  <p>&nbsp;</p>
  <p><font face="Arial, Helvetica, sans-serif">The information has been added.</font></p>
</div>
</body>
</html>
EOF
}
```

```
sub remhtml {
print<<"EOF";
<html>
<head>
<title>C22download - infomation removed</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
```

```
<body bgcolor="#0000CC" vlink="#CCCCCC" text="#FFFFFF" link="#CCCCCC"
alink="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr bgcolor="#3333FF">
    <td width="50%"><font face="Arial, Helvetica, sans-serif"><b><font
size="+4">c22download</font></b></font></td>
    <td width="50%">
      <div align="right"><font face="Arial, Helvetica, sans-serif"><a href="http://www.cache-
22.co.uk">Cache-22.co.uk</a><br>
      Version number - 1.01</font></div>
    </td>
  </tr>
</table>
<div align="left">
  <p>&nbsp;</p>
  <p><font face="Arial, Helvetica, sans-serif">The information has been removed.</font></p>
</div>
</body>
```



```

</html>
EOF
}
sub modhtml {

print<<"EOF";
<html>
<head>
<title>C22download - infomation modified</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#0000CC" vlink="#CCCCCC" text="#FFFFFF" link="#CCCCCC"
alink="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr bgcolor="#3333FF">
    <td width="50%"><font face="Arial, Helvetica, sans-serif"><b><font
size="+4">c22download</font></b></font></td>
    <td width="50%">
      <div align="right"><font face="Arial, Helvetica, sans-serif"><a href="http://www.cache-
22.co.uk">Cache-22.co.uk</a><br>
      Version number - 1.01</font></div>
    </td>
  </tr>
</table>
<div align="left">
  <p>&nbsp;</p>
  <p><font face="Arial, Helvetica, sans-serif">The information has been modified.</font></p>
</div>
</body>
</html>
EOF
}

```