МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РК

НАО «АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ Г. ДАУКЕЕВА»

Кафедра космической инженерии

«ДОПУЩЕН К ЗАЩИТЕ»

Зав.кафедрой: PhD, доцент Алипбаев К.А.

_____ «___» июня 2020г.

(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка рекомендаций по выбору методов оценки надёжности программных средств космического назначения

Специальность: 5В074600 – Космическая техника и технологии

Выполнил: Султанов К.О. Группа: КТТ-16-1

Научный руководитель: к.т.н., проф. Исмаил Е.Е.

_____ «___» июня 2020г.

(подпись)

Консультанты:

по экономической части: к.э.н., проф. Ибришев Н.Н.

_____ «___» июня 2020г.

(подпись)

по безопасности жизнедеятельности: ст. преподаватель Тыщенко Е.М.

_____ «___» июня 2020г.

(подпись)

Нормоконтролер: PhD, доцент Алипбаев К.А.

_____ «___» июня 2020г.

(подпись)

Рецензент: Инчин А.С.

_____ «___» июня 2020г.

(подпись)

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РК

НАО «АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ Г. ДАУКЕЕВА»

Институт космической инженерии и телекоммуникаций (ИКИТК)

Специальность Космическая техника и технологии

Кафедра космической инженерии

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Султанову Кайсару Орынбасарулы

Тема проекта *Разработка рекомендаций по выбору методов оценки надёжности программных средств космического назначения*

утверждена приказом ректора №__ от «___» _____ *2020г.*

Срок сдачи законченной работы «*12*» *июня 2020г.*

Исходные данные к проекту, требуемые параметры результатов проектирования (исследования): *Общие международные и госстандарты по выбору методов оценки надёжности ПСКН*

Поставлены задачи:

1 разработать требования к оценке:

1.1 сформировать цели оценки;

1.2 сформировать спецификации требований к надёжности ПС;

1.3 установить степени жёсткости (строгости) оценки для обеспечения уверенности в надёжности программного продукта (определяется набором оцениваемых показателей, методами и результатами оценки);

2 спецификация оценки:

2.1 установить оцениваемые показатели надёжности, методов и методик для их измерения, исходных данных, инструментальных средств для оценки;

2.2 определить критерии принятия решений по отдельным показателям надёжности ПС (пороговые значения, уровни доверия);

2.3 определить критерий принятия решений для интегрированной оценки надёжности;

3 разработать проект оценки надёжности ПС

3.1 разработать план действий по оценке (цель, бюджет (ресурсы), исполнители, ожидаемые результаты оценки, график основных этапов, условия среды оценки, методы и инструменты оценки, принятые стандарты и др.);

4 выполнить оценки:

4.1 провести измерения;

4.2 применить критерии для показателей надёжности ПС (сравнение результатов измерений с критериями и требованиями);

4.3 применить критерий для интегральной оценки качества ПС (вычислить комплексные показатели и сравнить их с соответствующими критериями и требованиями);

5 подготовить заключение об оценке:

5.1 сделать анализ результатов оценки;

5.2 создать отчёт об оценке;

5.3 сделать анализ пригодности результатов оценки;

5.4 завершить оценку и распорядиться результатами оценки.

Рекомендуемая основная литература:

1. ГОСТ Р 51901.5-2005 (МЭК 60300-3-1:2003) Руководство по применению методов анализа надежности

2. Харченко В.С. Методы моделирования и оценки качества и надежности программного обеспечения / В. С.Харченко, В. В. Скляр, О. М. Тарасюк – Харьков: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2004. – 159 с.].

3. Исмаил Е.Е. Методика выбора моделей оценки надежности для программных средств космического назначения//Вестник Алматинского университета энергетики и связи.- №3 (34), 2016.- с. 20-26

4. Исмаил Е.Е. Показатели надежности программных средств космического назначения //Вестник Алматинского университета энергетики и связи.- №3 (34), 2016.- с. 27-35

5. ISO/IEC 25040:2011 Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). Evaluation process (Системная и программная инженерия. Требования к качеству и оценка системы и программного продукта (SQuaRE). Процесс оценки.- (httphttps://www.iso.org/obp/ui/#iso:std:iso-iec:25040:ed-1:v1:en)

ГРАФИК

подготовки дипломного проекта

№	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1	Введение	01.04.20-03.04.20	
2	Анализ моделей и методов оценки надежности программных средств	07.04.20-12.04.20	
3	Методика выбора моделей оценки надежности ПСКН	16.04.20-28.04.20	
4	Разработка рекомендаций по выбору и применению методов оценки надёжности ПСКН	29.04.20-25.05.20	
5	Экономическое обоснование	17.05.20-29.05.20	
6	Санитарно-гигиеническая оценка условий труда	25.05.20-30.05.20	

Дата выдачи задания «1» апреля 2020г.

Заведующий кафедрой _____ PhD, доцентАлипбаев К.А.

(подпись)

Руководитель _____ к.т.н., проф. Исмаил Е.Е.

(подпись)

Задание принял к исполнению _____ студент Султанов К.О.

(подпись)

Алматы 2020

АҢДАТПА

Осы жылдар ішінде бағдарламалық қамтамасыз ету бизнес-процестерде жалақы төлеу жүйесінен әуе компанияларын брондау жүйелеріне дейін кең таралды. Бағдарламалық жасақтама бизнес-процестерді жеңілдетуде маңызды рөл атқарады. Осы бағдарламалық жасақтама жүйелерінің маңыздылығын ескере отырып, олардың сапасын басқару олар қолдайтын бизнес-процестердің сәтті болуы үшін өте маңызды. Сапа атрибуттары бағдарламалық қамтамасыздандырудың сапалық атрибуты үшін алынған функцияларды жақсырақ түсіну үшін бағдарламалық жасақтама сапасының маңызды болжаушылары болғандықтан, бұл функцияларды бағдарламалық жасақтаманың сапасын жақсарту және оның бизнес-процестерін қанағаттандыру үшін қажетті функционалдық талаптарды анықтау үшін басқаруға болады. Сәйкесінше, бағдарламалық қосымшаны әзірлеу кезінде клиенттің қажеттіліктері толық анықталуы керек, содан кейін оны ертерек талдау кезеңінде бағдарламалық жасақтаманы жасау процесінде қадағалап, оған қайтару керек. Ұсынылған сауалнама бағдарламалық жасақтама әзірлеушілеріне функционалдық сенімділіктің талаптарын тіркеуге және бағдарламалық жасақтама жүйесінің сенімділігіне байланысты функцияларды көрсетуге мүмкіндік береді.

АННОТАЦИЯ

За эти годы программное обеспечение стало повсеместным в бизнес-процессах от систем начисления заработной платы до систем бронирования авиабилетов. Программное обеспечение играет жизненно важную роль в облегчении бизнес-процессов. Учитывая важность этих программных систем, управление их качеством имеет важное значение для успеха бизнес-процессов, которые они поддерживают. Поскольку атрибуты качества являются важными предикторами качества программного обеспечения, чтобы обеспечить лучшее понимание связанных функций, приобретенных для каждого атрибута качества программного обеспечения, этими функциями можно манипулировать, чтобы улучшить качество программного проекта и определить требуемые функциональные требования, необходимые для удовлетворения связанных с ним деловые процессы. Соответственно, потребности клиента должны быть полностью выявлены при разработке программного приложения, а затем отслеживаться и возвращаться к нему на протяжении всего процесса разработки программного обеспечения на этапе раннего анализа требований. Предлагаемый вопросник позволяет разработчикам программного обеспечения фиксировать требования функциональной надежности и указывать функции, связанные с надежностью, для программной системы.

ABSTRACT

Over the years, the software has become ubiquitous in business processes from payroll systems to airline reservation systems. Software plays a vital role in facilitating business processes. Given the importance of these software systems, managing their quality is essential to the success of the business processes they support. Because quality attributes are important predictors of software quality as to provide a better understanding of the related features acquired for each software quality attribute, those features can be manipulated to improve the quality of a software project and determine the desired functional requirements necessary to satisfy the associated business processes. Accordingly, customer needs should be completely elicited in developing the software application then traced and referred back to throughout the software development process during the early requirements analysis phase. The proposed questionnaire empowers software developers to capture the functional reliability requirements and specify reliability-related features for a software system.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ……7

1 Анализ моделей и методов оценки надежности программных средств……....…….9

1.1 Надёжность программы…….....10

1.2 Особенности и показатели надежности программных средств …..11

1.3 Общие требования к оценке надёжности программных средств.....11

1.4 Современные модели и методы оценки надежности программных средств …….....12

2 Методика выбора моделей оценки надежности ПСКН ……..…..…14

2.1 Методы статического анализа программного обеспечения 14

2.2 Анализ потока управления…….....……15

2.3 Анализ потока данных…….....18

2.4 Обработка статистики сбоев программного обеспечения……25

3 Разработка рекомендаций по выбору и применению методов оценки Надежности ПСКН……..27

3.1 Рекомендации по выбору и применению вероятностных методов оценки надежности ПСКН…….....27

3.2 Рекомендации по применению метода анализа видов, последствий и критичности отказов ПО (SFMECA)……29

3.3 Рекомендации по применению метода дерева отказов (SFTA)….....30

4 Экономическая часть……..30

4.1. Принципы оценки экономической эффективности ПС……..…31

4.2. Основные показатели экономической эффективности ПС…….31

4.3 Анализ и расчет затрат на разработку ПС…….33

5 ОБЖ……..41

5.1 Охрана труда и техника безопасности на рабочем месте……41

5.2 Техника безопасности при работе с электрическими приборами……....48
ЗАКЛЮЧЕНИЕ…….……...……...55
Список литературы…….57

ВВЕДЕНИЕ

Мировой опыт и всемирная статистика в области космонавтики, ракетостроения, космических и летательных аппаратов показывает, что около 30% аварий ракетно-космической и спутниковой техники было связано с отказами программно-технических комплексов, а также дефектами программных средств (ПС). Таковые обстоятельства требуют таких решений в области программных средств, которые обеспечат эти средства в космических системах заданными свойствами качества, в том числе надёжностью, безопасностью, а также способностью противостоять сбоям и ошибкам.

Так называемая «модель качества» выполняет функцию основы как для формального описания характеристик качества и их отношений, так и для оценки качества программных средств. Данное понятие используется в передовой практике в виде описания и оценки качества ПС. Как правило, структура модели качества описывается иерархией, где элементы – это множество различных характеристик и подхарактеристик. Также там выражены отношения подчинённости между ними. Таким образом можно подытожить, что МК представляет собой чётко упорядоченное множество взаимосвязанных характеристик и отношений меж ними.

Компоненты системы обеспечивают выполнение таких функций, как:

- подготовку входных данных
- автоматизированное проведение испытаний в режиме реального времени
- непрерывный мониторинг состояния объекта контроля
- формирование протоколов испытаний.

Нынешние достижения в сфере техники автоматического управления различными объектами и процессами имеют довольно высокий уровень. Одним из видов экспериментального исследования, направленного на формирование научного мировоззрения студента, является лабораторный практикум. Рассмотрим назначение компонентов системы:

Система архивирования назначается для ведения электронного архива параметров и команд, полученных от сервера в результате выстоянного периодического опроса. Интерпретатор обеспечивает исполнение алгоритмов испытаний в виде циклограмм на специализированном языке, пошаговую отладку этих самых циклограмм, просмотр значений переменных при пошаговой откладке, а также написание и редактирование циклограмм.

Компьютеризированные системы все чаще используются в космических кораблях как для управления полетами, так и для управления измерительным оборудованием. Но космические миссии требуют больших инвестиций и иногда не могут быть отложены или повторены, потому что они связаны со специальными природными явлениями. По этим причинам отказ компьютерной системы может привести к непоправимым последствиям, как в случае с Фобосом I. В данной статье представлена деятельность по повышению качества программного обеспечения и оценке надежности программного обеспечения. В настоящее время эта деятельность продолжается и касается программного обеспечения, которое будет использоваться для управления измерительным оборудованием во время российско-европейской космической миссии на Марсе. Дано краткое описание миссии и измерительного оборудования. Затем принятые методологии предотвращения ошибок, а также обнаружения и обнаружения ошибок и их сдерживания анализируются вместе с критериями их выбора. Наконец, в документе представлены действия по оценке надежности программного обеспечения, которые будут выполнены с использованием экспериментальной модели, специально разработанной для критического применения.

1 Анализ моделей и методов оценки надежности программных средств

Сегодня критически важные системы становятся все более зависимыми от программных средств управления для выполнения критических функций. Одним из ярких примеров являются космические миссии. К сожалению, сбои программного обеспечения являются одной из наиболее распространенных причин сбоев в современных критически важных системах. Сбои программного обеспечения в таких миссиях могут привести к снижению производительности миссии или даже к полному провалу миссии, что повлечет за собой серьезные научные и экономические потери. Полетное программное обеспечение – это программное обеспечение, которое запускается на борту космического корабля. Это встроенная система программного обеспечения реального времени, область, в которой наблюдается экспоненциальный рост, где размер программного обеспечения увеличивается на порядок каждые десять лет. Это было особенно сложно в области космических кораблей, самолетов и автомобилей. Основным источником роста являются все более амбициозные требования и преимущества размещения новой функциональности в программном, а не аппаратном обеспечении.

Еще одной проблемой, вызывающей обеспокоенность, является растущая сложность архитектуры систем полетов, которая порождает все более сложные взаимозависимые переменные, которые могут вызвать сбой, что делает тестирование таких систем чрезвычайно сложным. Ошибки программного обеспечения («ошибки») являются основными причинами сбоев программного обеспечения.

Эти дефекты могут быть вставлены на любом этапе жизненного цикла программного обеспечения. Чтобы максимизировать вероятность успеха миссии, качество программного обеспечения должно оцениваться и оцениваться на каждом этапе жизненного цикла программного обеспечения. Это может предотвратить распространение дефектов от одной фазы к другой.

Кроме того, измерение качества программного обеспечения с точки зрения надежности программного обеспечения в любое время жизненного цикла программного обеспечения станет мощным инструментом для руководителей проектов, чтобы решить, стоит ли переходить к следующему этапу или тратить дополнительное время и ресурсы на текущем этапе, чтобы гарантировать конкретные уровни. качества программного обеспечения.

Одним из наиболее ценных хранилищ информации, собранной на этапах жизненного цикла программного обеспечения, являются отчеты о дефектах (ошибках). Эти отчеты о дефектах используются для отслеживания дефектов, обнаруженных на любом этапе жизненного цикла программного обеспечения, включая этапы разработки, тестирования и миссии. Хотя большинство организаций используют отчеты о дефектах исключительно для целей отслеживания ошибок, анализ этих отчетов дает представление о процессе разработки программного обеспечения и, следовательно, часто используется для определения качества программного обеспечения.

Важно понимать, что в контексте жизненного цикла программного обеспечения «дефекты» в широком смысле включают в себя ошибки, ошибки или любые другие аномалии, обнаруженные в программном обеспечении, которые могут вызывать или не вызывать сбой. Другой практический способ понять, что «дефект» является необходимым изменением программного обеспечения.

1.1 Надёжность программы

Надежная программа прежде всего должна обеспечивать достаточно низкую вероятность отказа в процессе функционирования в реальном времени. Быстрое реагирование на искажения программ, данных или вычислительного процесса и восстановление работоспособности за время меньшее, чем порог между сбоем и отказом, обеспечивают высокую надежность программ. При этом некорректная программ может

функционировать абсолютно надежно. В реальных условиях по различным причинам исходные данные могут попадать в области значений, вызывающих сбои, не проверенные при испытаниях, а также не заданные требованиями спецификации и технического задания.

Если в этих ситуациях происходит достаточно быстрое восстановление, такое что не фиксируется отказ, то такие события не влияют на основные показатели надежности - наработку на отказ и коэффициент готовности. Следовательно, надежность функционирования программ является понятием динамическим, проявляющимся во времени и существенно отличается от понятия корректности программ.

При оценке качества программ по показателям надежности регистрируются только такие искажения в процессе динамического тестирования с исполнением программ, которые приводят к потере работоспособности ПС или их крупных компонент. Первопричиной нарушения работоспособности программ при безотказности аппаратуры всегда является конфликт между реальными исходными данными, подлежащими обработке, и программой, осуществляющей эту обработку. Работоспособность ПС можно гарантировать при конкретных исходных данных, которые использовались при отладке и испытаниях. Реальные исходные данные могут иметь значения, отличающиеся от заданных техническим заданием и от использованных при применении программ. При таких исходных данных функционирование программ трудно предсказать заранее и весьма вероятны различные аномалии, завершающиеся отказами.

1.2 Особенности и показатели надёжности программных средств

Надёжность программного обеспечения – это вероятность безотказной работы программного обеспечения в течение определенного периода времени в указанной среде. Надежность программного обеспечения также является важным фактором, влияющим на надежность системы. Он отличается от надежности оборудования тем, что отражает совершенство

конструкции, а не совершенство производства. Высокая сложность программного обеспечения является основным фактором, влияющим на надежность программного обеспечения.

Надежность программного обеспечения не зависит от времени - хотя исследователи придумали модели, связанные с этими двумя. Методика моделирования для надежности программного обеспечения достигает своего расцвета, но прежде чем использовать методику, мы должны тщательно выбрать подходящую модель, которая лучше всего подходит для нашего случая. Измерение в программном обеспечении все еще находится в зачаточном состоянии. Хорошие количественные методы не были разработаны для представления надежности программного обеспечения без чрезмерных ограничений. Для повышения надежности программного обеспечения могут использоваться различные подходы, однако сложно сбалансировать время и бюджет разработки с надежностью программного обеспечения.

1.3 Общие требования к оценке надёжности программных средств

При разработке требований к программному обеспечению инженер-программист должен указать и описать нефункциональные требования, описывающие поведение функций системы, в виде функциональных требований, определяющих требуемые функции системы, которые должны быть выполнены, а затем разработать, протестировать и настроить конечные результаты для пользователей системы. ISO 19761 предлагает общую модель функциональных требований к программному обеспечению, которая проясняет границу между аппаратным и программным обеспечением.

Программное обеспечение ограничено постоянными устройствами хранения, такими как жесткий диск, оперативная память или память ПЗУ. Его функциональность встроена в потоки групп данных, которые можно охарактеризовать четырьмя типами перемещения данных. Обмен данными с

пользователями через границу (т. Е. На внешнем интерфейсе) разрешается с помощью перемещений «Входы и выходы», в то время как на внутреннем интерфейсе движения «Чтение и запись» позволяют обмениваться данными с оборудованием постоянного хранения. Одновременно с этим пользователи могут считаться оборудованием ввода-вывода, которое напрямую взаимодействует с программным обеспечением. COSMIC - ISO 19761 предназначен для измерения размера программного обеспечения на основе идентифицируемого функционала.

Исследование требований к надёжности программного обеспечения в программных проектах располагает девятью требованиями. Это различные стандарты ECSS (Европейское сотрудничество в области космической стандартизации) для аэрокосмической промышленности выражают соответствующие элементы надёжности в различных представлениях системы. Надёжность рассматривается как приемлемая вероятность отказа системы, которая основана на характеристиках надёжности и доступности оборудования для проверки следующего:

- Режимы и последствия отказов
- Отказоустойчивость, обнаружение и восстановление отказов
- Статистические данные о сбоях для поддержки прогнозов и оценка риска
- Способность аппаратного обеспечения работать с программным обеспечением или обслуживаться человеком в соответствии со спецификациями
- Демонстрация надежности критических элементов
- Обоснование баз данных, используемых для теоретических демонстраций.

1.4 Современные модели и методы оценки надежности программных средств

1. Модель Ченга (Cheung model) описывает архитектуру системы программного обеспечения, состоящей из n компонентов с помощью поглощающего DTMC с матрицей вероятности перехода P = [Pij], где Pij - вероятность передачи управления от компонента I к компоненту j. Поведение при сбое определяется надежностью компонента Ri. Метод решения является составным; добавляются два поглощающих состояния S и F, представляющих правильный выходной сигнал и сбой соответственно, и матрица P вероятности перехода изменяется соответствующим образом. Исходная вероятность перехода Pij изменяется на Ri Pij, которая представляет вероятность того, что компонент дает правильный результат, и управление передается компоненту j.

2. Модель Кубата (Kubat model) характеризует архитектуру программного обеспечения с помощью SMP, определяемого следующим образом. Переходы между компонентами следуют DTMC с матрицей вероятности переходов, и время пребывания в состоянии pdf. Предполагая постоянную интенсивность отказов, надежность компонента оценивается по мере того, как после оценки надежности компонентов подход к решению сводится к иерархической обработке модели Ченга. Ожидаемое количество раз, которое компонент выполняется во время однократного выполнения программного обеспечения, обозначается как, полученное из DTMC, и надежность системы становится R.

3. Модель Гокале (Gokhale et al model) описывает архитектуру с помощью поглощающей DTMC и использует метод иерархического решения. Однако он отличается подходом, используемым для оценки надежности компонентов. Учитывая зависящую от времени интенсивность отказов λi (t) и совокупное ожидаемое время, проведенное Viti в компоненте на выполнение приложения, надежность компонента оценивается как

$$Ri = e\int_{0}^{viti} \lambda i(t)dt$$

4. Кришнамурти (Krishnamurthy et al) используют экспериментальный подход для получения оценок надежности пути; последовательность компонентов вдоль различных путей наблюдается с использованием трасс компонентов, собранных во время тестирования. Трассировка компонента программы P для данного контрольного примера, обозначенная M (P, t), представляет собой последовательность компонентов, выполняемых i, когда выполняется P, что приводит к надежности пути.

Требования, приведенные в следующем вопроснике, представляют некоторые важные функции, которые должен выполнять инженер-программист при выполнении плана управления надежностью жизненного цикла. Эти функциональные требования были определены специалистами по информационным системам, которые имели практический опыт разработки программного обеспечения или, другими словами, были вовлечены в различные аспекты разработки программного обеспечения.

Эти авторы пришли к выводу, что необходимо соблюдать эти требования, если разработчик программного обеспечения должен успешно создавать высоконадежное программное обеспечение. Респондентам вопросника было предложено оценить каждый пункт в инструменте по пятибалльной шкале Лайкерта от «полностью не согласен» до «полностью согласен» в зависимости от их восприятия. Принимая во внимание незапрашиваемую природу полевого исследования, длину вопросника и огромное количество исследовательских опросов, предназначенных для профессионалов ИБ (включая инженеров требований, системных разработчиков, разработчиков программного обеспечения, операционной поддержки системы, специалистов по обслуживанию программного обеспечения, тестировщиков и т. Д.) который в настоящее время и ранее занимался управлением проектами разработки программного обеспечения из Центра информации и документации, а также сотрудниками

вычислительного и технологического факультета Арабской академии наук и технологий (AAST) из-за их опыта в программировании и анализе, реагировании показатель был признан удовлетворительным и вполне сопоставимым с другими аналогичными исследованиями.

Уровень опыта и знаний каждого участника оценивался с использованием подразделения профиля пользователя. Кроме того, все участники были неоплачиваемыми волонтерами, профессионально заинтересованными в разработке требований к программному обеспечению. Однако следует отметить, что была предпринята попытка дать исчерпывающее описание протокола вопросника, чтобы эмпирическая оценка могла быть глубоко исследована и воспроизведена в будущей работе.

2 Методика выбора моделей оценки надежности ПСКН

2.1 Методы статического анализа программного обеспечения

Общие положения для статического анализа программного обеспечения

Статический анализ программного обеспечения – это изучение программного кода (ручного или автоматического) без его непосредственного выполнения для:

- получение общей статистической или метрической информации о программных объектах;
- выявление конкретных типов дефектов в программных объектах;
- выявление отклонений от стандартов программирования;
- проверка правильности программных объектов.

Статический анализ может быть реализован следующими методами (см. Таблицу 1):

Таблица 1 - Методы статического анализа программного обеспечения

Методы анализа	Цели анализа	Объекты анализа

Анализ потока управления	Верификация последовательности выполнения и структурированности кода	Операторы, которые определяют последовательность выполнения кода
Анализ потока данных	Верификация использования переменных	Операнды и вычислительные операторы
Метрическая оценка ПО	Верификация корректности и сложности программных объектов	Измеряемые характеристики программного кода

2.2 Анализ потока управления

Анализ потока управления используется для проверки структуры программного кода в отношении:

- последовательности выполнения кода;
- структурированный код;
- наличие логических ошибок;
- наличие разделов неисполнимого (недостижимого) кода;
- наличие конфликтующих или неполных (неполных) межмодульных интерфейсов;
- определение точек входа и выхода процедур и циклов.

Этот метод не всегда позволяет непосредственно обнаруживать дефекты программного обеспечения, однако, идентификация различных аномалий в потоке управления помогает локализовать области, которые могут содержать потенциальную проблему в программном коде. Во время анализа потока управления программный код из текстовой формы превращается в график (схема потока управления), который легче, чем код, подлежащий анализу. Для сложного программного обеспечения схемы управления должны быть многоуровневыми. Подробные диаграммы строятся для прямого анализа программного кода, в том числе во время проверок. В таблице 2 приведены обозначения, которые используются для построения

блок-схем управления. На рисунке 1 показан фрагмент списка программ, а на рисунке 2 показана схема управления, соответствующая этому списку.

Таблица 2 – Обозначение операторов на схемах управления

Последовательный оператор	Условный оператор	Оператор перехода	Оператор цикла
			Состоит из инициализатора, инкремента и тела цикла

```
        Begin module                          1
L2:     x = x + 1                             2
        y = 7
L3:     if z < 4                              3
                        then x = 2*x          4
                        else x = x – 1        5
        if z = 0        then go to L2         6
        if y = x - z    then go to L3         7
        if z = 2*z                            8
                        then y = y - 1        9
                        w = 2*x
        else if z < 2   then go to L2         10
        end module                            11
```

Рис. 1 – Фрагмент листинга программы

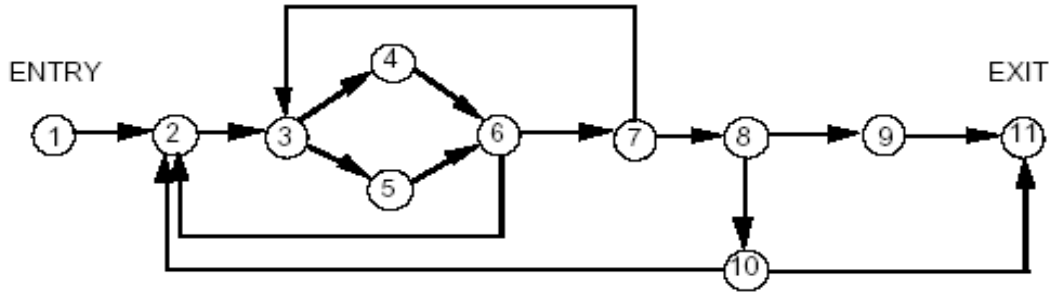


Рис.2 – Пример диаграммы потока управления

Анализ потока управления, как правило, проводится вместе с анализом потока данных. Анализ потока управления может также использоваться для поддержки структурного тестирования программного обеспечения путем:

- анализ синхронизма;
- анализ заморозок;
- контроль целостности программного кода.

В качестве входной информации для анализа потока управления может использоваться следующее:

- требования к программному обеспечению (с точки зрения технических характеристик и требований к интерфейсу) вместе с конструкцией компьютерной системы;
- программный проект;
- программный код вместе с пользовательской документацией.

Анализ потока управления выполняется с различной степенью детализации: от моделей компонентов и диаграмм состояний компьютерной системы более высокого уровня до диаграмм уровня потока управления подпрограмм, модулей и процедур, а также диаграмм, описывающих действия пользователя , Все вышеперечисленные источники входной информации могут быть использованы для создания отчета об анализе потока управления, который должен включать, среди прочего, рекомендации по устранению выявленных несоответствий.

Основным преимуществом анализа потока управления является относительная простота использования, которая достигается за счет использования соответствующих автоматизированных инструментов.

Ограничениями анализа потока управления являются:

- потребность в квалифицированных специалистах и возможные затраты значительных временных ресурсов для интерпретации результатов применения метода;

- для возможности применения метода на ранних этапах разработки программного обеспечения (до написания программного кода) необходимо использовать соответствующие CASE-инструменты.

2.3 Анализ потока данных

Анализ потока данных используется для проверки переменных, используемых в программе, в отношении:

- инициализация, модификация, использование и передача переменных во время выполнения программы;
- отсутствие переменных, для которых значения не установлены;
- проверка доступа к локальным и глобальным переменным;
- отсутствие неиспользуемых переменных;
- использование формальных параметров.

Этот метод позволяет локализовать области, которые могут содержать потенциальную проблему в программном коде. Диаграмма потока данных должна дополнять и интегрироваться в диаграмму потока управления.

Диаграмма потока данных в исходном состоянии. Колонка должна быть передана в виде комментариев. Каждый ящик можно рассматривать как отдельный «белый ящик» или «черный ящик». Изучение программного кода облегчает анализ, поэтому нет необходимости анализировать, как выполняются преобразования данных.

Использование переменной x в программе показано на рисунке 1. Эта переменная используется в операторах, обозначенных вершинами 2, 4, 7, 9 на блок-схеме управления (см. Рисунок 2). На рисунке 3 показана диаграмма потока данных для переменной x. 5,12. Значение переменной присваивается дугам, в которых используются переменные x. Дуги обозначены символами, которые определяют характер использования переменной x:

- d (defined) - определение значений путем его инициализации или присвоения новых значений;

- u (used) - использовать переменную для расчетов.

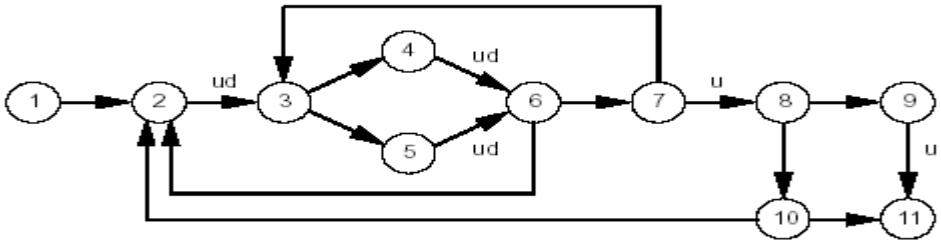


Рис. 3 - Пример диаграммы потока данных

При анализе безопасности критически важного программного обеспечения анализ потока данных выполняется в следующих направлениях:

- идентификация неопределенных входов, выходов или форматов данных;

- анализ динамического представления программного кода;

- контроль целостности данных;

- искать переменные, которые используются для их прямого объявления (подобный стиль программирования недопустим при разработке критически важного программного обеспечения, поскольку он может привести к программной ошибке);

- поиск неиспользуемых переменных, которые являются разделами неисполняемого кода;

- анализ протоколов передачи данных и выявление тех, которые не соответствуют установленным требованиям;

- анализ процедур обработки ошибок связи и выявление необработанных ошибок, которые могут быть распределены между частями программного обеспечения.

Разработка диаграмм потоков данных является итеративным процессом и может продолжаться на нескольких этапах процесса разработки программного обеспечения. В качестве входных данных для анализа потока данных следует использовать:

- требования к программному обеспечению;

- программный проект;

код программы.

Все вышеперечисленные источники входной информации следует использовать для составления отчета об анализе потока данных.

Основным преимуществом анализа потока данных является относительная простота использования, которая достигается за счет использования соответствующих автоматизированных инструментов и визуального представления кода.

Ограничениями анализа потока данных являются:

- в случае несоблюдения правил модульного программирования построение диаграмм потоков данных представляет определенную сложность;

- потребность в квалифицированных специалистах и затраты значительных временных ресурсов для интерпретации результатов применения метода;

- для применения метода на ранних этапах разработки программного обеспечения (до написания программного кода) необходимо использовать соответствующие CASE-инструменты.

2.4 Обработка статистики сбоев программного обеспечения

Производными данными для расчета показателей надежности программного обеспечения является протокол тестирования программного обеспечения, который в простейшем случае содержит информацию о времени между сбоями программного обеспечения или о моментах текущего времени, когда эти отказы произошли. На основании этой информации определяются другие показатели надежности. В этом случае возможны два подхода:

- интервал, когда показатели надежности усредняются за интервал времени;

- точка, когда показатели надежности пересчитываются после каждого сбоя.

При определении показателей надежности методом интервалов одним из основных допущений является то, что существует временной интервал $[t, t+\Delta t]$, для которого $\lambda(t) = const$. Для определения показателей надежности выберите значение τ - время тестирования программного обеспечения. Разделяют это время на i равных интервалов. Тогда $t_i = \frac{\tau}{i}$, где t_i - длина временного интервала, n_i - количество программных сбоев в i-м интервале тестирования.

Общее (кумулятивное) число отказов ПО до момента времени $t = t_i \cdot i$:

$$N_i = n_1 + n_2 + ... + n_i . \quad (1.1)$$

График $n_i(t)$ называют профилем отказов, а $N_i(t)$ – кумулятивным профилем отказов.

Интенсивность отказов ПО (функция риска) на i-м интервале тестирования определяют по формуле:

$$\lambda_i = \frac{n_i}{t_i}, \quad (1.2)$$

Средняя наработка ПО на i-м интервале тестирования:

$$T_i = \frac{t_i}{n_i} = \frac{1}{\lambda_i}. \quad (1.3)$$

ВБР ПО за время t на i-м м интервале тестирования:

$$P_i(t) = e^{-\lambda_i t_i} . \quad (1.4)$$

Пример отчета о тестировании ПО для системы реального времени представлены в таблице 5.8. Основным при обработке статистических данных об отказах ПО является выбор длины временного интервала, при этом рассчитывают интенсивность отказов (функцию риска) и другие показатели надежности. Несмотря на то, что на начальной стадии тестирования отказы происходят чаще, чем на последующих, в ряде случаев целесообразно выбирать различные длины интервалов. Кроме того,

необходимо, чтобы на каждом из интервалов количество отказов ПО была больше нуля. Для расчета показателей надежности ПО были выбраны следующие временные интервалы для расчета ПО: 1000 с - для $0 \leq t \leq 20000$ с; 5000 с - для $20000 \leq t \leq 100000$ с.

3 Разработка рекомендаций по выбору и применению методов оценки надёжности ПСКН

Многие крупные программные проекты изо всех сил пытаются достичь своих целей надёжности. Модели роста надёжности программного обеспечения количественно определяют надёжность по данным о сбоях, собранным во время тестирования программного средства. Здесь представлен инструмент оценки отказоустойчивости и надёжности программного обеспечения (SFRAT), который реализует несколько моделей повышения надёжности программного обеспечения в виде бесплатного приложения с открытым исходным кодом. Природа инструмента с открытым исходным кодом позволяет пользователям интегрировать методы в рабочий процесс своей организации, а исследователи - добавлять дополнительные статистические методы. Инструмент представлен в контексте проекта HACA.

3.1 Рекомендации по выбору и применению вероятностных методов оценки надёжности ПСКН

Инструмент оценки отказоустойчивости и надёжности программного обеспечения – это бесплатный инструмент с открытым исходным кодом, разработанный для содействия регулярной количественной оценке надежности программного обеспечения, улучшения обмена такими оценками и развития диалога между сообществами исследователей и практиков. SFRAT реализует ранние, но популярные модели роста надежности программного обеспечения, в которых используются статистические методы для определения кривой наилучшего соответствия данным о сбоях с целью прогнозирования будущих тенденций. Выводы

позволили включить число отказов, которые будут обнаружены при дополнительном тестировании, насколько интенсивность отказов уменьшится, а также насколько увеличится время до отказа и надежность.

Такое увеличение дополняется хорошими мерами соответствия, позволяющими объективно определить, какая из нескольких моделей может быть предсказана наилучшим образом для планирования планирования выпуска. Например, программы Major Defense Acquisition определяют интенсивность отказов в качестве ключевого системного атрибута, который является минимально приемлемым значением, и, следовательно, разработчики порогового уровня должны достигнуть, чтобы гарантировать, что система функционально эффективна и пригодна при рассмотрении решений о полной скорости производства. Пройдя этот этап, вы получаете доступ к полному производству и сопровождается выделением значительных долларов налогоплательщикам.

SFRAT реализован на языке программирования R, а также на Python и может использоваться на компьютерах под управлением Windows, OSX или Linux. Исходный код R доступен на GitHub и работает в среде RStudio или аналогичной среде R. Версия Python работает с таким интерфейсом, как Anaconda. Природа инструмента с открытым исходным кодом упрощает проверку информации перед использованием конфиденциальных данных о сбоях. Кроме того, веб-экземпляр доступен с веб-сайта проекта, а также примеры наборов данных об ошибках. Это избавляет от необходимости устанавливать программное обеспечение или форматировать данные об ошибках организации в требуемом формате перед оценкой, позволяя заинтересованным лицам оценить функциональность инструмента, чтобы определить, подходит ли он для их потребностей, прежде чем тратить дополнительное время.

Первоначальный выпуск SFRAT поддерживает три формата данных о сбоях, включая время отказов, время отказов и счетчик отказов, а также два трендовых теста для повышения надежности, чтобы определить,

является ли применение моделей подходящим. В настоящее время SFRAT включает в себя две модели степени опасности, в том числе модель Jelinski – Moranda и геометрическую, а также четыре модели подсчета отказов, а именно модель Goel-Okumoto, задержанную S-образную форму, изгиб S в форме и Вейбулла. Два критерия соответствия подходят для выбора модели. Архитектура приложения была разработана для поддержки включения в инструмент дополнительных моделей и показателей степени соответствия. Окамура и Дохи внесли свой вклад в одиннадцать моделей.

3.2 Рекомендации по применению метода анализа видов, последствий и критичности отказов ПО (SFMECA)

Режимы и анализ сбоев программного обеспечения (SFMEA) – это процесс разработки программного обеспечения, который исследует потенциальное влияние предполагаемых сбоев программного обеспечения на систему и ее среду. Когда критичность эффектов также учитывается, метод называется программным режимом сбоя, анализа эффектов и анализа критичности (SFMECA). SFMEA и SFMECA в основном используются для выявления дефектов проектирования программного обеспечения во время разработки программного обеспечения.

Процесс: SFMEA – это структурированный, основанный на таблицах процесс обнаружения и документирования способов, с помощью которых программный компонент может выйти из строя, и последствий этих отказов. Вход в SFMEA – это спецификация проекта с подробными требованиями программного обеспечения. После определения объема анализа (какие компоненты программного обеспечения и какой уровень детализации являются подходящими) аналитик определяет режимы отказов каждого компонента. Процесс SFMEA руководствуется набором стандартизированных режимов отказов. Важно заметить, что SFMEA наиболее эффективен, когда режимы отказа включают аномальные программные данные и события, а не просто ошибочные данные и события.

Процесс SFMEA отслеживает распространение аномалий от причин (режимы отказов) до локальных (подсистема или компонент) воздействий к глобальным (системный или экологический) эффектам. Примером режима отказа является «превышение доступной мощности». Левые столбцы описывают режимы отказа, возможно, с точки зрения ориентировочных слов; правые столбцы описывают локальные и системные эффекты. В SFMECA стандартная оценка критичности присваивается каждому режиму отказа на основе его вероятности возникновения и серьезности его последствий.

SFMEA – это метод восходящего анализа. SFMEA описывает влияние режима предполагаемого сбоя при его распространении через программное обеспечение и систему. Как и в случае FMEA, ограничение заключается в том, что множественные одновременные сбои и сбои по общей причине могут не обнаруживаться. Интеграция SFMEA с методикой обратного анализа SFTA показала некоторый успех в преодолении этого ограничения.

3.3 Рекомендации по применению метода дерева отказов (SFTA)

Анализ дерева программных сбоев (SFTA) – это процесс статического анализа, который исследует возможные связанные с программным обеспечением причины сбоя или опасности. SFTA в основном используется при разработке программного обеспечения для обнаружения дефектов программного обеспечения. Наиболее эффективно применяется при наличии подробных требований или проектной документации. Позже он также использовался в процессе разработки для проверки кода программного обеспечения.

Процесс: SFTA – это структурированный, основанный на дереве процесс обнаружения и документирования причин, приводящих к возникновению нежелательного события, представленного в корневом узле. Вход в SFTA – это опасность для корневого узла или вызывающее

беспокойство событие, а также спецификация проекта или подробные требования к программному обеспечению. В качестве альтернативы SFTA может быть выполнена на программном коде.

SFTA – это нисходящая техника. В нем используются неформальные, обратные оценки отказов, которые можно легко поддерживать по мере развития системы.

4 Экономическая часть

Технико-экономическое обоснование разработки программного средства космического назначения. Работа инженера-разработчика относится к научно-исследовательской деятельности. Главной целью технико-экономической части в проекте это получение с данного проекта максимально качественный и точный программный продукт с минимальными расходами для получения финансовой выгоды. Для достижения такой цели в научно-исследовательской деятельности нужно квалифицированные сотрудники и современное оборудование.

Описание бизнес-проекта

Расширение «Лотос» является дополнением для интернет-браузеров для комфортного и безопасного времени в интернете.

Это расширение обеспечивает скрытие рекламы, отображает статус подключения сайта, что облегчает использование Интернета.

4.1 Принципы оценки экономической эффективности ПС

Преимущества расширения отображаются в двух терминах:

• Комфорт

• Безопасность

• Простота

Разработчики

Разработчик «Лотос» отличаются особыми знаниями в области CSS, HTML и JavaScript, что обеспечит стабильную работу расширения.

Оценка стоимости и специального оборудования для развития расширения

№	Наименование программного средства	Расход, количество	Цена, тенге	Сумма, тенге
1	Лотос	3	0	0

Таблица 1. Программное обеспечение для разработки расширений

№	Наименование устройства	Расход, количество	Цена, тенге	Сумма, тенге
1	Ноутбук	3	350 000	1050000

Таблица 2. Оборудование для развития расширения

4.2 Основные показатели экономической эффективности ПС

Наш бизнес предлагает конкретный продукт в виде приложения для браузера.

Расширение будет полностью бесплатным в первый раз, что соответственно не покроет наши расходы. Этот период будет длиться 3 месяца, чтобы напрямую сообщать о работе расширения и обратной связи с пользователем.

Для продвижения бизнеса будет предпринят ряд мер, таких как:

• ASO оптимизация. Важно правильно выбрать значок, сделать скриншоты, составить описание.

• Работа с пиарщиками. Чем больше средств массовой информации узнает о приложении, тем больше загружаемых расширений.

• анализ результатов

Рис. 1. Взаимодействие системы «Лотос» с аппаратной частью

4.3 Анализ и расчет затрат на разработку ПС

Финансовый анализ – это изучение основных показателей финансового состояния, которые дают объективную оценку состоянию организации.

Целью финансового анализа можно назвать «характеристику» финансового состояния организации, предприятия, предприятия.

Начальные расходы

Создание проекта требует приобретения стартового капитала. Чтобы выяснить, сколько расходов необходимо определить основные объекты расходов, а именно:

• Аренда или покупка помещений

• Коммунальные услуги

• Оборудование и программное обеспечение

• Зарплата

Месячные расходы

Для определения ежемесячных затрат требуется таблица, которая включает все текущие расходы, связанные с проектом.

Сотрудники нашего бизнес-проекта получают заработную плату. Формулы, по которым можно определить заработную плату:

$$\mathbf{РН = О_о + Д_з} \quad (3.1), \text{ где:}$$

$О_о$ – основной оклад

$Д_з$ – дополнительная зарплата

Дополнительная заработная плата составляет 10% от основной заработной платы.

Мы используем следующую формулу для расчёта дополнительной заработной платы:

$$\mathbf{Д_з = О_о * 0.1} \text{ (3.2), где:}$$

$О_о$ = 80000 тг

$Д_з$ = 80000 * 0,1 = 8000 тг - дополнительная заработная плата в месяц

Социальный налог составляет 11% от начисленного дохода.

Формула расчета социального налога:

$$\mathbf{СН = 9.5\%*(ПФ - РС)} \text{ (3.3), где:}$$

РС - Взнос в пенсионный фонд

ПФ - Платежный фонд

Выплата в пенсионный фонд составляет до 10% от фонда оплаты труда. Эти вычеты не облагаются социальным налогом.

Формула, используемая для расчета:

$$\mathbf{РС = ПФ * 0.1} \text{ (3.4), где:}$$

РС = 88000 * 0.1 = 8800 - 1 месяц

СН = 9.5% * (88000 - 8800) = 7 524 – социальный налог

Затраты на электроэнергию

Создание нашего бизнес-проекта может потребовать некоторых затрат в виде коммунальных услуг. Коммунальные услуги включают в себя такие составляющие, как:

• Электричество

• Вода

• Отопление

Следующая формула используется для расчета их энергетических затрат:

$$\mathbf{Э = З_{э.о.} + Д_р} \text{ (3.5), где:}$$

$З_{э.о.}$ - затраты энергии на оборудование

$Д_{р.}$ - Дополнительные расходы

Для расчета затрат энергии на оборудование мы используем следующую формулу:

$$\mathbf{З_{э.о.} = Вт*Ч*Т*К_и} \text{ (3.6), где:}$$

Где:

Вт – Потребляемая мощность (0,9 кВт)

Ч – рабочие часы (1968 часов в год)

Т – тариф (11,01 тг)

$К_и$ – коэффициент использования (0,9)

$$\mathbf{З_{э.о.} = 0.9*5904*11.01*0.9 = 52652 \text{ тг}}$$

Расчеты на дополнительные нужды составляют 5% потребляемой мощности оборудования. Формула, используемая для их расчёта:

$$\mathbf{Д_р = 0.05* З_{э.о.}} \text{ (3.7)}$$

$$Д_р = 0 .05 * 52652 = 2632 \text{ тг}$$

Итоговая стоимость электричества:

$$Э = 52652 + 2632 = 55284 \text{ тг}$$

Расчет цены расширения браузера

Расчет прибыли с процентной доходностью 30%.

Формула для расчета прибыли:

$$\mathbf{П = C_р*R/100} \text{ (3.8), где:}$$

$С_р$ – стоимость

$П_п$ - процент прибыльности

$$П = 110\ 000*30/100 = 33\ 000 \text{ тг}$$

Стоимость программы равна сумме общих затрат и прибыли. Он рассчитывается по формуле:

$$\mathbf{C_п = C_р + P} \text{ (3.9)}$$

$C_п$ = 110 000 + 300 = 110 300 тг

Формулы, используемые для определения НДС и стоимости программы, включая НДС:

НДС = $C_п$*18/100 (3.10)

НДС = 110 300*18/100 = 19 854 тг

Цена = НДС + $C_п$ (3.11)

Цена = 19 854 + 110 300 = 130 134 тг

Ориентировочная стоимость «Лотос» составляет 130 134 тенге.

Чтобы рассчитать срок окупаемости проекта, мы должны разделить цену проекта на приблизительный доход от расширения.

Окупаемость = 1534/300 = 5.1 месяцев

В течение года проекта экономическая эффективность будет выглядеть так, что за каждый вложенный тенге мы получим - 7,8 тг.

Эффективность = 12000/1534 = 7.8 тг

Амортизационные отчисления

Таблица 1. Оборудование необходимое для разработки программного средства

Наименование	Характеристики	Количество	Цена за единицу, тг	Общая сумма, тг
Системный блок Neo Game PBA	- Процессор: Intel Core Ci3-9100F 3,6GHz - ОЗУ: 8GB - Память: 1TB - Видеокарта: GTX1650 4GB	1	209 990	209 990
Монитор Acer K272HLEbid черный	- ЖК-монитор с диагональю 23.8"; - тип матрицы экрана	1	60 990	60 990

	IPS; - разрешение 1920x1080 (16:9); - подключение: VGA, HDMI; - яркость 250 кд/м2; - контрастность 3000:1; - время отклика 5 мс; - встроенные динамики.			
Клавиатура Genius Scorpion K220 LED Black	- интерфейс USB; - для настольного компьютера; - классическая клавиатура; - мембранные; - ширина 46.6 см.	1	7 990	7 990
Компьютерная мышь A4Tech N-708X Black	- проводная мышь; - интерфейс USB; - 6 кнопок управления; - масса 70 грамм.	1	2 990	2 990
Принтер Epson L-805 СНПЧ A4-W	- МФУ (принтер/сканер/копир); - цветная термическая струйная печать; - применение персональный; - максимальный формат A4	1	107 990	107 990
Итого:		5	389 950	

АО (Амортизационные отчисления) переводит постепенно с основных производственных фондов на стоимость производимого программного продукта.

В данном проекте выбран линейный метод расчета АО, потому что производится постепенный и равномерный переход стоимости оборудования к предоставляемой услуге. Срок службы оборудования – 4 года.

Норма амортизации: $\text{На} = \frac{1}{\text{Т}_{\text{сл}}} \cdot 100\% = \frac{1}{4} \cdot 100\% = 25\%$

АО для оборудования: $\text{А}_1 = \Phi_1 \cdot \frac{\text{На}}{100\%} = 209\,990 \cdot \frac{25\%}{100\%} = 52\,498$ тг

$$\text{А}_2 = \Phi_2 \cdot \frac{\text{На}}{100\%} = 60\,990 \cdot \frac{25\%}{100\%} = 15\,248 \text{ тг}$$

$$\text{А}_3 = \Phi_3 \cdot \frac{\text{На}}{100\%} = 7\,990 \cdot \frac{25\%}{100\%} = 1\,998 \text{ тг}$$

$$\text{А}_4 = \Phi_4 \cdot \frac{\text{На}}{100\%} = 2\,990 \cdot \frac{25\%}{100\%} = 748 \text{ тг}$$

$$\text{А}_5 = \Phi_5 \cdot \frac{\text{На}}{100\%} = 107\,990 \cdot \frac{25\%}{100\%} = 26\,998 \text{ тг}$$

Таблица 2. АО для каждого элемента оборудования в год

Наименование	Первоначальная стоимость, тг	Норма амортизации, %	Сумма амортизации, тг
Системный блок Neo Game PBA	209 990	25	52 498
Монитор Acer K272HLEbid черный	60 990		15 248
Клавиатура Genius Scorpion K220 LED Black	7 990		1 998
Компьютерная мышь A4Tech N-708X Black	2 990		748

Принтер Epson L-805 СНПЧ A4-W	107 990		26 998
Итого:			97 490

АО на время разработки программного средства для каждого элемента оборудования:

$$\mathrm{A}_1 = \frac{\mathrm{На} \cdot \mathrm{C}_{\text{пер}} \cdot N}{100 \cdot 12 \cdot n} = \frac{25\% \cdot 209\,990 \cdot 42}{100\% \cdot 12 \cdot 21} = 8\,750 \text{ тг}$$

$$\mathrm{A}_2 = \frac{25\% \cdot 60\,990 \cdot 42}{100\% \cdot 12 \cdot 21} = 2\,541 \text{ тг}$$

$$\mathrm{A}_3 = \frac{25\% \cdot 7\,990 \cdot 42}{100\% \cdot 12 \cdot 21} = 333 \text{ тг}$$

$$\mathrm{A}_4 = \frac{25\% \cdot 2\,990 \cdot 42}{100\% \cdot 12 \cdot 21} = 125 \text{ тг}$$

$$\mathrm{A}_5 = \frac{25\% \cdot 107\,990 \cdot 42}{100\% \cdot 12 \cdot 21} = 4\,500 \text{ тг}$$

Таблица 3. АО на время разработки проекта

Наименование	Первоначальная стоимость, тг	Норма амортизации, %	Сумма амортизации, тг
Системный блок Neo Game PBA	209 990	25	8 750
Монитор Acer K272HLEbid черный	60 990		2 541
Клавиатура Genius Scorpion K220 LED Black	7 990		333
Компьютерная мышь A4Tech N-708X Black	2 990		125
Принтер Epson	107 990		4 500

L-805 СНПЧ А4-W			
Итого:			16 249

Затраты на сертификацию и лицензирование (при среднем написании 140 строк за рабочий день и общем объеме 1 000 строк выходит 1 строка = 68 тенге)

$$Р_л = 1\,000 \cdot 68 = 68\,000 \text{ тг}$$

Затраты на аренду

Помещение находится в корпусе ТОО «Қазақстан Ғарыш Сапары», г.Нур-Султан. Площадь лаборатории 50 кв.м. Ежемесячная стоимость помещения – 300 000 тг. Арендная плата за весь срок проекта (2 месяца) – 600 000 тг.

Затраты на рекламу (10% от суммы затрат)

1 000 000 · 10% = 10 000 тг

Расходы на дополнительный фонд (20% от суммы затрат) = 200 000 тг

Накладные расходы (50% от суммы затрат) = 500 000 тг

Общая сумма всех затрат (себестоимость): С = 1 000 000+ 200 000 + 100 000 + 500 000 = 1 800 000 тг.

Таблица 4. Структура затрат продукта

№	Затраты	Сумма, тг	Доля от себестоимости, %
1	ФОТ	88 000	4.67
2	Социальный налог	7	0.47
3	Оборудование	389 950	20.73
4	АО	16 249	0.86
5	Арендная плата	600 000	31.9
6	Лицензирование и	68 000	3.61

	сертификация		
7	Реклама и поиск покупателя	10 000	0.53
8	Дополнительный фонд	200 000	10.63
9	Накладные расходы	500 000	26.58

Окупаемость бизнеса

На основе сравнений существующих аналогов и оценке их характеристик было принято взять цену разрабатываемого программного продукта: **Ориентировочная стоимость «Лотос» составляет 130 134 тенге.**

За капитальные вложения примем значение себестоимости продукта - 1 800 000 тенге.

В первый год планируется выпуск 6 копий ПП и каждый последующий год повышать количество продаж на 30%.

Чистый поток платежей на 4 года: P1 = 680 000 тг, P2 = 1 040 000 тг, P3 = 1 280 000 тг, P4 = 1 420 000 тг.

Норма дисконта – 18%

Накопленная общая величина дисконтированных доходов:

$$PV = \sum_{i=1}^{n} \frac{P_t}{(1+r)^t}$$

$$PV1 = \frac{680\ 000}{(1+0.18)^1} = 576\ 271 \text{ тг};$$

$$PV2 = \frac{1\ 040\ 000}{(1+0.18)^2} = 746\ 911 \text{ тг};$$

$$PV3 = \frac{1\ 280\ 000}{(1+0.18)^3} = 779\ 047 \text{ тг};$$

$$PV4 = \frac{1\ 420\ 000}{(1+0.18)^4} = 732\ 420 \text{ тг}.$$

Общая величина PV = 576 271 + 746 911 + 779 047 + 732 420 = 2 834 649 тг.

PV > I => проект является прибыльным и полностью окупается меньше чем за 4 года.

Чистая текущая стоимость проекта:

$$NPV = \sum_{i=1}^{n} \frac{PV}{(1+r)^t} - I = 2\,834\,649 - 1\,800\,000 = 1\,034\,649 \text{ тг.}$$

NPV = 1 034 649 тг > 0 Это доказывает о прибыльности данного проекта.

Индекс рентабельности:

$$PI = \sum_{i=1}^{n} \frac{\frac{PV}{(1+r)^t}}{I} = \frac{2\,834\,649}{1\,800\,000} = 1.57$$

PI = 1.57 – это объем компенсированной единицы.

Внутренняя норма доходности:

Для расчета IRR воспользуемся командой ВСД в MS Excel. IRR = 26%

Срок окупаемости инвестиций:

$$DPP = t + \frac{I - PV_4}{PV_{t+1}} = 1 + \frac{1\,800\,000 - 732\,420}{681\,890} = 1.56.$$

DPP = 1.56 года ≈ 1 год 7 месяцев.

Таблица 5.8. Показатели экономической эффективности

Показатель	Значение
I, тг	1 800 000
NPV, тг	1 034 649
PI	1.57
IRR, %	26
DPP, лет	1.56

Прогноз продаж

Прогноз продаж – это объем продаж, который может быть достигнут при соблюдении условий.

Существует прогнозная модель, основанная на:

• Исторические данные

• Рыночные тренды

• Усилия по увеличению будущих продаж

Процесс прогнозирования продаж можно разделить на несколько этапов:

• Оценка текущей ситуации

• Анализ исторической информации

• Планирование желаемых показателей

Вывод: Данный проект по разработке программного средства является окупаемым и прибыльным. При капитальном вложении на сумму 1 800 000 тенге и цене продукта в 130 134 тенге проект окупается в 1 год 7 месяцев. Все вышеперечисленные формулы и коэффициенты были взяты с методичек: «Методические указания к выполнению экономической части дипломной работы для студентов специальности 5В070400 – вычислительная техника и программное обеспечение», «Методические указания для экономической части выпускной работы для студентов всех форм обучения специальности 050719 - Радиотехника электроника и телекоммуникации».

5 Охрана труда и безопасность жизнедеятельности (ОБЖ)

5.1 Охрана труда и техника безопасности на рабочем месте

Охрана труда (ОТ) - система законодательных актов, социально-экономических, организационных, технических, гигиенических, лечебно профилактических мероприятий, обеспечивающих безопасность, здоровье и работоспособность человека в процессе труда.

Охрана труда РК регламентирована нормами, заложенными в Конституции, законах Республики Казахстан, нормативных правовых актах. Требования к выполнению правил охраны труда и технике безопасности распространяются на все организации, независимо от формы собственности, сферы хозяйственной деятельности и ведомственной подчиненности. Условия труда должны соответствовать требованиям техники безопасности и охраны труда работников.

Обязанности работодателя в области безопасности и охраны труда определены в Трудовом кодексе Республики Казахстан. Дополнительные требования по безопасности и охране труда на предприятиях устанавливаются актами работодателя, трудовым, коллективным договорами, соглашениями в соответствии с законодательством. Таким образом, можно сказать, что охрана труда – это обязательное условие для любой организации.

Оснащенность рабочего места.

вид	количество
Работник	14
Компьютер	10
Телефон	10
Рабочий стол	14
Офисный стул	20

Физические характеристики помещения.

Параметр	Значение	Дата измерения
Площадь офиса	100 кв. м.	15.07.2019
Уровень освещенности	300 Лк	15.07.2019
Уровень шума	20 Дб	15.07.2019
Температура воздуха	22 °C	15.07.2019
Влажность воздуха	50%	15.07.2019

Условия труда

Количество рабочих часов	8часов
Время для отдыха	1час

Содержание работы	Обработка кодов
Характер трудовых нагрузок	Преимущество умственные

Описание оборудования:

Компьютер HP с процессорами Intel 6-го и последующих поколений, для которых выполнен переход с ОС Windows 7, Windows 8, Windows 8.1 или Windows 10 Профессиональная на предыдущую версию Windows 7 Профессиональная, Windows 8 Профессиональная или Windows 8.1 распространяется следующее условие: Microsoft предоставляет ограниченную поддержку этой версии ОС Windows, используемой с процессорами или чипсетами в данной системе.

Что необходимо знать при организации безопасности и охраны труда на предприятии:

- принимать меры по предотвращению любых рисков на рабочих местах и в технологических процессах путем проведения профилактики, замены производственного оборудования и технологических процессов на более безопасные;
- проводить обучение, инструктирование, проверку знаний работников по вопросам безопасности и охраны труда, а также обеспечивать документами по безопасному ведению производственного процесса и работ за счет собственных средств;
- организовать обучение и проверку знаний по вопросам безопасности и охраны труда руководящих работников и лиц, ответственных за обеспечение безопасности и охраны труда, периодически не реже одного раза в три года в организациях, осуществляющих повышение квалификации кадров, в порядке, установленном уполномоченным органом по труду, согласно списку, утвержденному актом работодателя;
- создать работникам необходимые санитарно-гигиенические условия;

- не допускать к переноске и передвижению работников, не достигших восемнадцатилетнего возраста, тяжестей, превышающих установленные для них предельные нормы;

Техника безопасности при работе с оборудованием

К работе на персональных компьютерах -допускаются работники, не имеющие медицинских противопоказаний к выполнению данного вида работ, прошедшие инструктаж по вопросам охраны труда, с группой по электробезопасности не ниже первой. Женщины со времени установления беременности и в период кормления грудью к выполнению всех видов работ, связанных с использованием ПК не допускаются.

При выполнении работ на ПК возможно неблагоприятное воздействие на работников следующих опасных и вредных производственных факторов:

- повышенный уровень электромагнитных излучений;
- повышенный уровень ионизирующих излучений;
- повышенный уровень статического электричества;
- повышенная напряженность электростатического поля;
- повышенная или пониженная ионизация воздуха;
- повышенная яркость света;

Рабочий стол с учетом характера выполняемой работы должен иметь достаточный размер для рационального размещения **монитора (дисплея), клавиатуры**, другого использования оборудования и документов, поверхность, обладающую низкой отражающей способностью.Клавиатуру следует располагать на поверхности стола таким образом, чтобы пространство перед клавиатурой было достаточным для опоры рук работника.

Рабочий стул должен быть устойчивым, место сидения должно регулироваться по высоте, а спинка сиденья — по высоте, углам наклона, а

также расстоянию спинки от переднего края сиденья. Регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Рабочее место размещается таким образом, чтобы естественный свет падал сбоку(слева). Для снижения яркости в поле зрения при естественном освещении применяются регулируемые жалюзи, плотные шторы.

Для обеспечения безопасности работников на соседних рабочих местах, расположением между рабочими столами с мониторами должно быть не менее 2,0 м, а расстояние между боковыми поверхностями мониторов – не менее 1,2 м.

Для снижения уровня напряженности электростатического поля при необходимости применяются экранные защитные фильтры. При эксплуатации защитный фильтр должен быть плотно установлен на экране монитора и заземлен.

Для обеспечения оптимальных параметров микроклимата проводятся регулярное в течение рабочего дня проветривание и ежедневная влажная уборка помещений, используются увлажнители воздуха.

При работе с ПК обеспечивается доступ работников к первичным средствам пожаротушения, аптечкам первой медицинской помощи

Анализ опасных и вредных производственных факторов в офисе

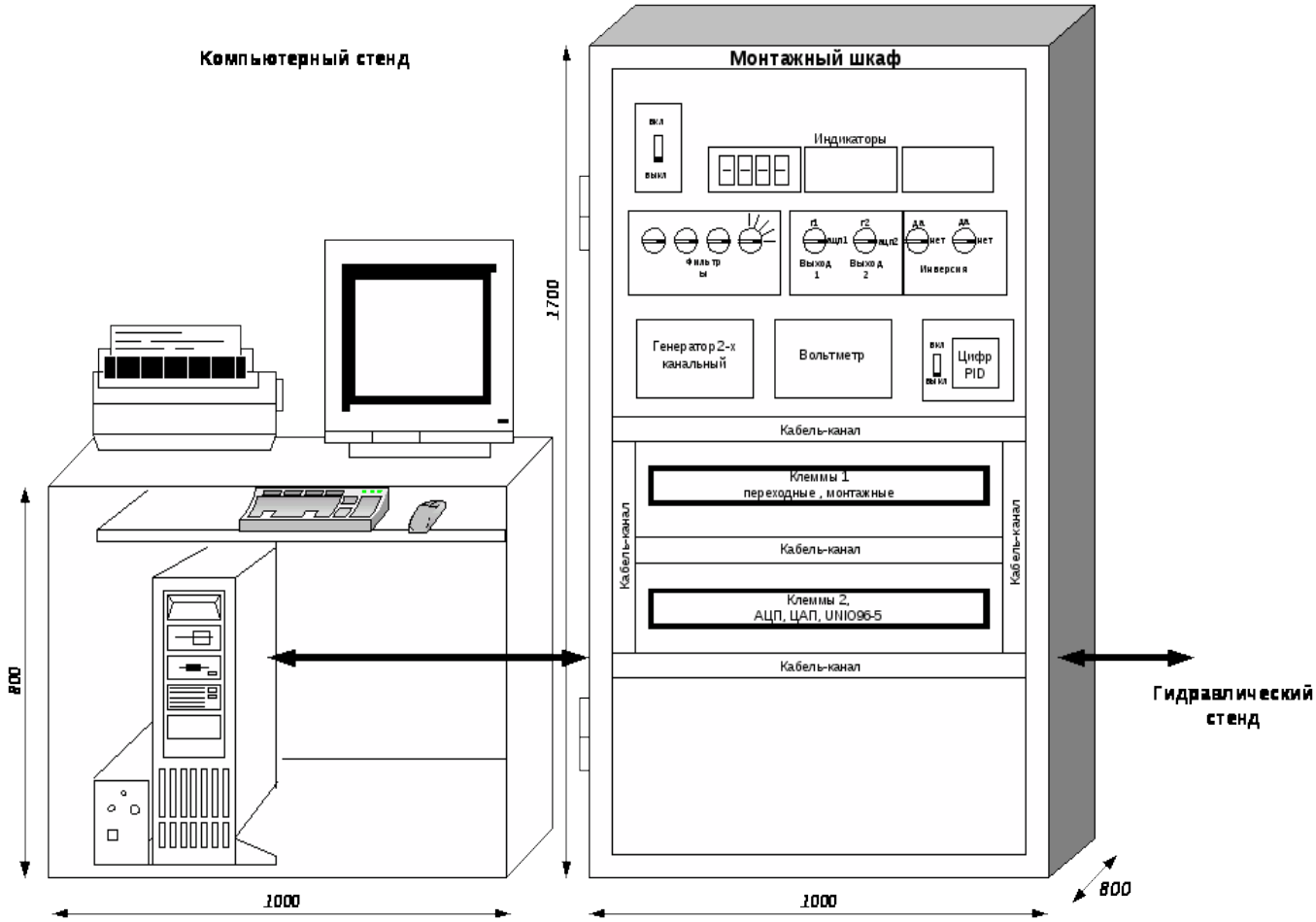
Рассмотрим воздействующие на человека опасные и вредные производственные факторы:

- Физические опасные и вредные производственные факторы

Опасность получить какую-либо серьезную травму исключена. Каждая жила провода(кабеля)изолируется в оболочку, которая, в свою очередь, *покрывается* защитным покровом.

- Пониженная или повышенная температура поверхностей оборудования, материалов. Стационарная лабораторная установка, состоящая из компьютерного стола, монтажного шкафа и кабелей

соединения. Устройств, образующих холод, нет. Устройства, вырабатывающие тепло, скрыты защитной оболочкой



- Повышенная или пониженная температура воздуха рабочей зоны.

Офис имеет автоматическую систему поддержания температуры воздуха в пределах комнатной темперетуры.

- Повышенный уровень шума на рабочем месте

Самым громким источником шума является компьютерное оборудование. Воздействие шума отражается как на органах слуха, так и на общем психологическом состоянии человека. Возможны глухота, нервные расстройства.

- Повышенный уровень вибрации. Источников вибрации нет.
- Биологические опасные и вредные производственные факторы.

Бактерии, вирусы. Скапливаются в местах, труднодоступных для проведения

уборки: например, клавиатура. Могут повлечь различные по тяжести заболевания.

Производственная санитария и гигиена труда

Производственная санитария – система организационных, гигиенических и санитарно-технических мероприятий и средств, предотвращающих воздействие на работающих вредных производственных факторов.

Гигиена труда - медицинская наука, изучающая воздействие окружающей производственной среды, характера трудовой деятельности на организм работающего. Разработка санитарно-гигиенических нормативов и практических мероприятий, устранение неблагоприятных производственных факторов, предупреждение или ослабление их влияния на организм человека являются основными задачами гигиены труда.

- Все работники должны соблюдать правила личной гигиены:

работать в удобной обуви, плотно сидящей на ноге, на непромокаемой и нескользящей подошве, в опрятной одежде

- Принимать пищу только в предназначенных для этой цели местах, отвечающих санитарно-гигиеническим требованиям. Прием пищи на рабочем месте строго запрещается.
- Для обеспечения санитарно-бытовых удобств работающих на предприятиях должны быть оборудованы:

- комната для отдыха

- гардеробы для хранения одежды и личных вещей, душевые, умывальники;

- помещения для личной гигиены женщин в наиболее крупных предприятиях;

- ответственность за соблюдение правил личной гигиены и содержание рабочего места в надлежащем состоянии несет каждый работник предприятия.

- На каждом предприятии и на рабочих местах должны быть созданы необходимые санитарно-гигиенические условия труда в соответствии с нормативами. Этими нормами регламентируются необходимые для здоровья и благоприятного труда площадь и объем производственных помещений, освещение и отопление, метеорологические условия (температура, влажность, давление воздуха), шум и вибрация, содержание пыли в воздухе.
- Освещение производственных помещений может быть естественным и искусственным. Искусственное освещение бывает: общее, местное и комбинированное. Требования к освещению: достаточная освещенность рабочих поверхностей, рациональное направление света на них, отсутствие резких теней и бликов на рабочих местах.

5.2 Техника безопасности при работе с электрическими приборами.

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества Поражение электричеством является самой основной причиной большинства несчастных случаев в офисе.

Токопроводящие элементы могут находиться не только лишь на самом производстве, но и на вашем рабочем месте под столом, за тем же компьютером и принтером.

Правила для соблюдения электробезопасности в офисе:

- все приборы требуют питания от сети 220 Вольт.
- в офисе запрещает пользоваться шнурами питания, изоляция которых имеет небольшие трещины и другие повреждения.
- Никаких скрученных проводов изоляционной лентой не должно быть и близко.

- Все электрошнуры не должны быть сильно короткими и слишком длинными. Категорически запрещается прибивать их к полу гвоздями либо накрывать ковром, ковролином, линолеумом и другими напольными покрытиями.
- Все розетки, независимо от того, где они установлены, нельзя перегружать. Они рассчитаны на определённую величину нагрузки. В связи с этим, использование тройников крайне нежелательно.
- В тех компаниях, в которых применяется довольно большое количество воды, вся электрическая сеть в обязательном порядке должна быть защищена не только лишь автоматическими выключателями, но и устройствами защитного отключения

Пожаробезопасность

Пожарная безопасность - состояние объекта, при котором исключается возможность пожара, а в случае его возникновения предотвращается воздействие на людей опасных факторов и обеспечивается защита материальных ценностей.

Согласно статистике, пожары в офисе чаще всего происходят из-за курения сотрудников, неисправных отопительных приборов, хранения легковоспламеняющихся материалов, возгорания электропроводки и неаккуратного обращения с огнем во время приготовления пищи.

Основное назначение важного элемента пожарной сигнализации — системы оповещения — предупреждение людей о возгорании. Для этого используются звуковые (сирены, динамики) и световые сигналы тревоги (табло, указатели, мигающие оповещатели).

Наш офис относится в категорию Д, так как в офисе установлены:

- пожарные извещатели автоматической сигнализации, передающие сигнал в помещение пожарного поста, охраны или диспетчерской здания, а так же иметься воздушно-пенные, воздушно-эмульсионные, порошковые огнетушители.

Расчёты

$$Фл = \frac{Ен * S * k * z}{N * \eta * n}$$

http://elektrika-svoimi-rykami.com

1.Фл – световой поток лампы,

2. Ен – норма освещенности

3. S – площадь помещения

4. k - коэффициент запаса

5. z – поправочный коэффициент

6. N – количество принятых светильников

7. η – коэффициент использования светового потока

8. n – число ламп в светильнике.

Решение:

1.Ен-нормированная освещенность. В офисах общего назначения с использованием компьютеров равна от 200-300Лк. У нас она равна на 300Лк.

Ен=300

Представим значение в формулу:

Фл = (Ен * S * k * z) / (N * η * n)

Фл = (300 * S * k * z) / (N * η * n)

2.S-площадь помещения

S - площадь помещения (метры квадратные - м2)

a - длина помещения 20 м

b - ширина помещения 5 м

Подставим наши значения

S = a * b = 20*5=100м2

S = 100м2

Подставим данные в формулу:

Фл = (Ен * S * k * z) / (N * η * n)

Фл = (300 * 100 * k * z) / (N * η * n)

3.k-коэффициент запаса

Запыленность значительно менее 1 мг/м3, отсутствие паров кислот и щелочей	Жилые, административные и офисные и т.п. помещения	K=1

K = 1.

Подставим значение в формулу:

Фл = (Ен * S * k * z) / (N * η * n)

Фл = (300 * 100 * 1 * z) / (N * η * n)

4.z – поправочный коэффициент

z - поправочный коэффициент, применяемый в помещениях где требуется освещенность больше чем нормируемая минимальная

Данный коэффициент следует применять в помещениях где планируется выполнение точной зрительной работы, например, читать или писать.

В нашем случае z=1,1

Вставляем данные в формулу:

Фл = (Ен * S * k * z) / (N * η * n)

Фл = (300 * 100* 1 * 1,1) / (N * η * n)

5. N – количество принятых люстр

Освящать офис будет 9 люстр, расположенный в метре друг от друга

N = 0

Фл = (Ен * S * k * z) / (N * η * n)

Фл = (300 * 100 * 1 * 1,1) / (9 * η * n)

6. η – коэффициент использования светового потока

Для того что бы найти коэффициент использования светового потока нам потребуется рассчитать индекс помещения – i. Воспользуемся следующей формулой:

i = S / ((a + b) * h)

где,

i - индекс помещения,

S - площадь помещения 100 м2;

a - длина комнаты 20 м;

b - ширина комнаты 5 м;

h - высота подвеса светильника от пола 3 м;

Считаем:

i = S / ((a + b) * h) = 100 / ((20 + 5) * 3) = 1.3 округляем до значения близкого к:

0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 3, 3.5, 4, 5

В нашем случае это значение 1.25

Теперь нам потребуются данные о дизайне нашей комнаты. Конкретно интересуют три вещи пол, потолок и стены их цветовой оттенок в формате белый - светлый - темный - серый - черный. Например, бежевые стены будут относиться к светлым, красные, вишневые, коричневые к темным, с черным и белым и так все понятно.

Эти оттенки называются коэффициентом отражения (P) и выражаются в процентном соотношении следующим образом:

70% - белый;50% - светлый;30% - серый;10% - темный;0% - черн

70	Белый									
50	Светлый									
30	Серый									
10	Темный									
0	Черный	Р – коэффициент отражения								i
Потолок		70				50		30	0	
Стены		50		30		50	30		0	
Пол		30	10	30	10	10		10	0	
		0.28	0.28	0.21	0.21	0.25	0.19	0.15	0.13	0.5
		0.35	0.34	0.27	0.26	0.31	0.24	0.18	0.17	0.6
		0.44	0.39	0.32	0.31	0.39	0.31	0.25	0.24	0.7
		0.49	0.46	0.38	0.36	0.43	0.36	0.29	0.28	0.8
		0.51	0.48	0.4	0.39	0.46	0.39	0.31	0.3	0.9
		0.54	0.5	0.43	0.41	0.48	0.41	0.34	0.32	1
		0.56	0.52	0.46	0.43	0.5	0.43	0.35	0.33	1.1
		0.59	0.55	0.49	0.46	0.53	0.45	0.38	0.35	1.25
		0.64	0.59	0.53	0.5	0.56	0.49	0.42	0.39	1.5
		0.68	0.62	0.57	0.54	0.6	0.53	0.45	0.42	1.75
		0.73	0.65	0.61	0.56	0.63	0.56	0.48	0.45	2
		0.76	0.68	0.65	0.6	0.66	0.59	0.51	0.48	2.25
		0.79	0.7	0.68	0.63	0.68	0.61	0.54	0.51	2.5
		0.83	0.75	0.73	0.67	0.72	0.65	0.58	0.55	3
		0.87	0.78	0.77	0.7	0.75	0.68	0.61	0.59	3.5
		0.91	0.8	0.81	0.73	0.78	0.72	0.65	0.62	4
		0.95	0.83	0.86	0.77	0.8	0.75	0.69	0.65	5

http://elektrika-svoimi-rykami.com/

$\eta = 0.51$

Подставим полученные данные в формулу:

Фл = (Ен * S * k * z) / (N * η * n)

Фл = (300 * 100 * 1 * 1,1) / (9 * 0.51 * n)

7. n – число ламп в люстре

Люстра в нашем примере пяти рожковая, в ее конструкции предусмотрена установка 5 ламп.

n = 5

Вставляем данное значение в формулу:

Фл = (Ен * S * k * z) / (N * η * n)

Фл = (300 * 100 * 1 * 1,1) / (9 * 0.51 * 5)

Все необходимые значения найдены, теперь мы можем рассчитать Фл – световой поток лампы.

Считаем:

$Фл = (Ен * S * k * z) / (N * \eta * n)$

$Фл = (300 * 100 * 1 * 1,1) / (9 * 0.51 * 5) = 33000/22.95=1437,9$

Если округлим 1437,9 получим 1438.

Световой поток лампы измеряется в Люменах (Лм), равен:

Фл = 1438 Лм

Офисные здания относиться к объектам, оказывающим минимальное негативное воздействие на окружающую среду, то есть к IV категории (при условии, если в здании нет источников выбросов в атмосферный воздух и в системы водоотведения). Утилизация оргтехники и компьютеров для ИП и юридических лиц представляет собой длительный процесс. Сначала необходимо провести экспертизу для оценки состояния имущества, вышедшего из строя или устаревшего технически и морально, на основании которой имущество считается непригодным для дальнейшего использования. Акт с экспертной оценкой выдает только организация, имеющая соответствующий сертификат.

После акта технического осмотра составляется акт списания, после списания бухгалтерией основных средств с баланса предприятия может быть проведена **утилизация оргтехники и компьютеров**. Рассмотрим процедуру утилизации. Вначале компьютер или другая оргтехника разбирается по отдельным составляющим, производится сортировка. Металлические части направляются в переплавку. Неразбирающиеся части дробятся в специальных устройствах. Затем происходит разделение полученной крошки на фракции – пыль из пластмассы засасывается в воздухоприемник, а тяжелые металлические частицы оседают, где производится идентификация различных металлов, в том числе и драгоценных.

ЗАКЛЮЧЕНИЕ

Надлежащее определение требований к надежности программных систем в спецификациях программных требований в договорной документации позволяет избежать большинства проблем, возникающих после фазы разработки при практическом использовании программного обеспечения. Для пользователей системы важна надежность системы в целом, но для аналитиков и тестировщиков важно отделить требования к программному обеспечению от требований к оборудованию, поскольку существуют некоторые существенные различия. Нет никаких сомнений в том, что любая часть оборудования со временем изнашивается и выходит из строя, поэтому надежность аппаратного продукта может быть выражена с точки зрения времени до отказа или времени между отказами, при условии, что части оборудования будут продолжать работать в течение различных отрезков времени до отказа. В настоящее время неверно допущено, что работа со стандартными пакетами программного обеспечения, работающими на аналогичной конфигурации серверов определенным числом пользователей. Существуют различные варианты операционных платформ, предпочтения пользователей в отношении настроек и приоритетов, которые следует принимать во внимание. Следуя иллюстрированному подходу взвешивания, мы получаем осязаемую меру надежности продукта, которая позволяет превратить меру надежности в патронизированную форму, которая соответствует реальной среде. Для подтверждения действительности требуется больше исследований. Анкета может быть полезна как для исследователей, так и для практиков в понимании критически важных функций надежности программного обеспечения. Исследователи могут извлечь выгоду из инструмента с точки зрения качества создаваемого программного обеспечения. Он также предоставляет план по повышению надежности программного обеспечения.

Список литературы

1. «Методические указания для экономической части выпускной работы для студентов всех форм обучения специальности 050719 - Радиотехника электроника и телекоммуникации»
2. Липаев В.В. Выбор и оценивание характеристик качества программных средств. Методы и стандарты.—Москва: Синтег, 2001.-228с., 20 илл
3. ISO/IEC TR 9126-3:2003 Software Engineering – Product Quality – Part 3: Internal metrics (Инжиниринг программного обеспечения - Качество продукта - Часть 3: Внутренние метрики)
4. ISO/IEC 9126-1:2001 Software Engineering – Product Quality – Part 1: Quality model (Инжиниринг программного обеспечения - Качество продукта - Часть 1: Модель качества
5. ISO/IEC TR 9126-2:2003 Software Engineering – Product Quality – Part 2: External metrics (Инжиниринг программного обеспечения - Качество продукта - Часть 2: Внешние метрики)
6. ISO/IEC 60880, 1986 Software for Computers in the safety systems of nuclear power stations (Программное обеспечение для ЭВМ систем обеспечения безопасности АЭС)
7. RTCA DO 178B–1992 Software Considerations in Airborne Systems and Equipment Certification (Рассмотрение программного обеспечения при сертификации бортовых систем и оборудования)
8. “NASA Study on Flight Software Complexity,” NASA officeof Chief Engineer, Tech. Rep., 2009. [Online]. Available:http://www.nasa.gov/pdf/418878main FSWC Final Report.pdf
9. W. A. Madden and K. Y. Rone, “Design, Development, Integration:Space Shuttle Primary Flight Software System,” ACM Comm., vol. 27,no. 9, pp. 914–925, Sep. 1984
10. ГОСТ Р 51901.5-2005 (МЭК 60300-3-1:2003) Руководство по применению методов анализа надежности

11. Харченко В.С. Методы моделирования и оценки качества и надежности программного обеспечения / В. С.Харченко, В. В. Скляр, О. М. Тарасюк – Харьков: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2004. – 159 с.].

12. Исмаил Е.Е. Методика выбора моделей оценки надежности для программных средств космического назначения//Вестник Алматинского университета энергетики и связи.- №3 (34), 2016.- с. 20-26

13. Исмаил Е.Е. Показатели надежности программных средств космического назначения //Вестник Алматинского университета энергетики и связи.- №3 (34), 2016.- с. 27-35

14. Маевский Д А. Яремчук С.А. Анализ моделей надежности программного обеспечения гарантоспособных информационных систем // Електромашинобудування та електрообладнання, МНТЖ, Одесса, ОНПУ, 2010 – №76, К:. Техніка, 2010 – С. 68-79.