МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Дальневосточный федеральный университет»**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Кафедра прикладной математики, механики, управления и программного обеспечения**

КОЛЛЕКТИВНАЯ РАЗРАБОТКА ПРОЕКТА ТРАНСЛЯТОРА, ПЕРЕВОДЯЩЕГО ПОДМНОЖЕСТВО ЯЗЫКА C++ В ЭКВИВАЛЕНТНОЕ ПОДМНОЖЕСТВО ЯЗЫКА PASCAL

**КУРСОВАЯ РАБОТА**

по дисциплине «Технологии коллективной промышленной разработки информационных систем» по образовательной программе подготовки бакалавров по направлению 09.03.04 Программная инженерия

Выполнили студенты гр. Б9120-09.03.04

\_\_\_\_\_\_\_\_\_\_\_ Гончарук Д. М.

\_\_\_\_\_\_\_\_\_\_\_ Решетнев Н. Я.

\_\_\_\_\_\_\_\_\_\_\_ Назаров М. С.

\_\_\_\_\_\_\_\_\_\_\_ Курбанов М. М.

Руководитель: доктор технических наук Гриняк В.М.

\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_ 2024 г.

(подпись)

Защищён с оценкой \_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(подпись) И.О. Фамилия

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

г. Владивосток

2024 г.

**Оглавление**

[Введение 4](#_Toc58501542)

[1 Регламент проведения инспекции 5](#_Toc58501543)

[1.1 Процедура организации и проведения формальной инспекции 5](#_Toc58501544)

[1.2 Процедура организации и проведения неформальной инспекции 5](#_Toc58501545)

[1.3 Роли участников инспекции 6](#_Toc58501546)

[1.4 Порядок организации инспекции 6](#_Toc58501547)

[1.5 Порядок подготовки к инспекции 7](#_Toc58501548)

[1.6 Перечень статусов и степени важности замечаний 7](#_Toc58501549)

[1.7 Метрики, характеризующие эффективность инспекций 8](#_Toc58501550)

[2 Модель состояний задач 9](#_Toc58501551)

[2.1 Состояния задач 9](#_Toc58501552)

[2.2 Создание новой задачи 10](#_Toc58501553)

[2.3 Перехода задачи из состояния в состояние 10](#_Toc58501554)

[3 План проекта 10](#_Toc58501555)

[3.1 Проектная команда 10](#_Toc58501556)

[3.2 График разработки 11](#_Toc58501557)

[4 Требования к проекту 12](#_Toc58501558)

[4.1 История изменений документа 12](#_Toc58501559)

Введение

Процесс промышленной разработки информационных систем охватывает различные этапы, начиная с создания плана проекта и завершаясь тестированием для чего требуется сотрудничество многочисленных специалистов различных направлений. Также необходимы унифицированные методы взаимодействия между ними, которые позволяют распределить обязанности среди членов команды в соответствии с их экспертизой.

Исходя из описанного, необходимо применять конкретные технологии коллективной разработки для увеличения эффективности работы в группе и соответствия конечного продукта установленным требованиям.

В данной работе рассматривается проблема коллективной разработки программного средства "Транслятор из C++ в Pascal", а также составление технической документации для данного инструмента.

Таким образом, основной целью данной курсовой работы является создание программного средства "Транслятор из C++ в Pascal" с использованием методов коллективной промышленной разработки.

**Задачи курсовой работы:**

* Разработать план проекта;
* Разработать регламент проведения инспекции;
* Разработать модель состояний задач;
* Разработать требования к проекту;
* Разработать архитектуру проекта;
* Разработать программу измерений проекта;
* Разработать перечень задач проекта;
* Разработать рекомендации по кодированию;
* Разработать план тестирования проекта.

1 Регламент проведения инспекции

1.1 Процедура организации и проведения формальной инспекции

Формальные инспекции направлены на оценку завершенных частей рабочих продуктов, где требуется проверка документации по дизайну или функциональности кода с соответствующими тестами для верификации. Процедура формальной инспекции предполагает участие трех членов команды.

Этапы формальной инспекции включают следующие шаги:

Планирование инспекции;

Назначение инспекции;

Собрания по инспекции;

Завершение инспекции (распространение результатов, проверка исправления недостатков в рабочем продукте).

На этапе "Назначение инспекции" обязанности автора включают инициирование формальной инспекции рабочего продукта, для чего необходимо уведомить руководителя проекта.

Председатель обязан удостовериться, что рабочий продукт соответствует установленным критериям готовности к формальной инспекции.

Инспекторы, в свою очередь, должны независимо от других участников инспекции изучить предоставленный для инспекции рабочий продукт и заполнить соответствующие разделы протокола подготовки к формальной инспекции.

Начало формы

1.2 Процедура организации и проведения неформальной инспекции

Неформальные инспекции подвергаются маленькие части рабочих продуктов, где необходимо проверить работоспособность кода.

В процедуре формальной инспекции могут принимать участие 2 члена команды.

**Этапы неформальной инспекции:**

1. Назначение инспекции;

2. Собрания по инспекции;

3.Завершение инспекции (распространение результатов, проверка исправления недостатков в рабочем продукте).

1.3 Роли участников инспекции

1. **Автор (Author)** - сотрудник, разработавший инспектируемый рабочий продукт, либо сделавший инспектируемые изменения в существующем рабочем продукте.

**В роли могут быть: Гончарук Данил;**

1. **Председатель (Moderator)** - ответственный сотрудник, выполняющий роль председателя инспекции.

**В роли могут быть: Решетнев Никита;**

1. **Ведущий (Presenter)** - сотрудник, представляющий рабочий продукт инспекторам.

**В роли могут быть: Курбанов Марат;**

1. **Инспектор (Inspector)** - сотрудник, ответственный за эффективную

проверку инспектируемого рабочего продукта.

**В роли могут быть: Назаров Максим, Курбанов Марат;**

1. **Инженер по качеству** – сотрудник, ответственный за анализ метрик после инспекции.

**В роли могут быть: Решетнев Никита, Назаров Максим.**

1.4 Порядок организации инспекции

Этапы проведения инспекции включают следующие шаги:

**Планирование инспекции:** На этом этапе коллективно определяются дата, время, формат (очный или заочный) и платформа (при заочной инспекции) для проведения инспекции.

**Подготовка к инспекции:** Инспектор самостоятельно изучает предоставленный для инспекции рабочий продукт, используя свой опыт и соответствующие стандарты.

**Собрание по инспекции:** На собрании обсуждаются замечания и рекомендации, вынесенные инспектором по рабочему продукту. Обязательно присутствие инспектора и автора рабочего продукта, который подлежит инспекции. Участие остальных членов команды разработки опционально.

**Завершение инспекции:** Если рабочий продукт требует корректировок, автор фиксирует все замечания и рекомендации инспектора, разрабатывает план последующих работ и согласовывает его с инспектором. В случае, если доработки не требуются, инспектор подтверждает слияние рабочей ветки в основную ветку ("master").

1.5 Порядок подготовки к инспекции

На этапе назначения инспекции устанавливаются время и дата проведения, исходя из количества участников. Выбираются оптимальные сроки, удовлетворяющие всем членам команды, в пределах 2 дней после планирования инспекции. Важно отметить, что даты инспекций не назначаются на субботу, воскресенье или праздничные дни.

1.6 Перечень статусов и степени важности замечаний

**Дефект (Defect):** Проблема, выявленная на этапе, отличном от этапа внесения.

**Ошибка (Error):** Проблема, обнаруженная на том же этапе, на котором была выявлена.

**Комментарий (Comment):** Наблюдение, предложение, рекомендация или улучшение, предложенное для будущего выпуска рабочего продукта, или вопрос, требующий разъяснения.

**Замечание для исследования (Investigate):** Проблема, природа которой не может быть определена на собрании и требует дополнительного исследования.

Допустимые степени серьезности замечаний включают:

**Особо важная (Major):** Проблема, требующая срочного внимания и решения.

**Средняя (Moderate):** Проблема, которая влияет на работоспособность продукта, но может быть решена без немедленных мер.

**Мелкая, незначительная (Minor):** Небольшая проблема, оказывающая ограниченное воздействие на функциональность.

**Другие (Other):** Дополнительные категории для классификации замечаний, не подпадающих под основные степени серьезности.

**Порядок верификации учёта замечаний**

* Особо важная (Major) – неисправность или ошибки в программном продукте более чем 50%;
* Средняя (Moderate) - неисправность или ошибки в программном продукте более чем 20 %;
* Мелкая, незначительная (Minor) - ошибки в программном продукте более чем 10%;
* Другие (Other) – в иных случаях.

1.7 Метрики, характеризующие эффективность инспекций

Метрики по инспекциям собираются инженером по качеству **Решетневым Никитой** после каждой инспекции и анализируются с помощью контрольных карт.

**Метрики по инспекциям:**

* **Inspection Fault Density (IFD)**

IFD = (Количество найденных ошибок / Размер рабочего продукта)

Стратегическая цель метрики – повысить качество разрабатываемого ПО.

Изучаемый объект метрики – инспекция, измеряемый атрибут – плотность найденных в ходе инспекции ошибок.

Единица измерения – ошибка / <страница, требование, LOC, тест>.

* **Inspection Preparation Rate (IPR)**

Стратегическая цель метрики – повысить качество разрабатываемого

ПО.

В качестве метрики, характеризующей эффективность инспекции, была

выбрана Inspection Preparation Rate (IPR):

IPR = (Количество инспекторов \* Размер продукта) / Общее время

подготовки

Изучаемый объект метрики – подготовка к инспекции, измеряемый

атрибут – производительность подготовки к инспекции.

Единица измерения – <страница, требование, LOC, тест> / час

1. Модель состояний задач

2.1 Состояния задач

**Перечень возможных состояний задачи и их интерпретация:**

* New (Новая задача). Аналитиками создана новая задача;
* Task assignment (Задача назначена). Аналитик назначает задачу на конкретного автора;
* Coding (Задача в разработке). Задача в разработке у автора;
* Сheck or testing (Проверка или тестирование). Задача находится в состоянии проверки или тестируется;
* Closed (Задача выполнена). Задача закрыта.

2.2 Создание новой задачи

**Правила создания новой задачи:**

Список сотрудников, которые могут создавать новые задачи: **Гончарук Данил, Решетнев Никита.**

Новые задачи могут создаваться в случае выдвижения новых требований клиентом (Артемьевой Ириной Леонидовной).

2.3 Перехода задачи из состояния в состояние

**Правила перехода задачи из состояния в состояние:**

* **New** →**Task assignment**. Переход в случае назначения аналитиком автора на задачу;
* **Task assignment** → **Coding.** Переход, если назначенный автор начал работать над задачей
* **Coding** → **Сheck or testing.** Переход в случае, если автор закончил работу над задачей и можно провести тестирование.
* **Сheck or testing** → **Closed.** Переход в случае, если задача протестирована и 80% тестов прошли удачно.

1. План проекта

Документ, описывающий план проекта, предоставляет детальную информацию о проекте, включая данные об участниках, поставленных задачах и установленных сроках. Этот документ представляет собой конечный результат этапа планирования, утверждается перед началом любых работ и становится основным и надежным источником сведений о предстоящем проекте.

3.1 Проектная команда

Таблица 1. Проектная команда

|  |  |
| --- | --- |
| **Проектная команда** | **Члены команды** |
| Тимлид | Гончарук Данил |
| Кодер 1 | Гончарук Данил |
| Кодер 2 | Решетнев Никита |
| Инженер по качеству | Курбанов Марат |
| Технический писатель | Назаров Максим |
| Тестировщик 1 | Назаров Максим |
| Тестировщик 2 | Курбанов Марат |

3.2 График разработки

Таблица 2. График разработки

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Дата**  **Разработка** | **14.9.2023-21.9.2023** | **22.9.2023-30.9.2023** | **1.10.2023-12.10.2023** | **13.10.2023-21.10.2023** | **22.10.2023-31.10.2023** | **1.11.2023-15.11.2023** | **16.11.2023-31.11.2023** | **1.12.2023-10.12.2023** | **11.12.2023-20.12.2023** | **21.12.2023-31.12.2023** | **1.1.2024-10.1.2024** | **11.1.2024-20.1.2024** | **21.1.2024-10.1.2024** |
| **План проекта программного Обеспечения** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Спецификация маркетинговых требований** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Спецификация требований к программному обеспечению** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Спецификация проектирования Программного Обеспечения** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Разработка грамматики для лексического анализатора** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Разработка лексического анализатора** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Тестирование лексического анализатора** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Разработка грамматики для синтаксического анализатора** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Разработка синтаксического анализатора** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Тестирование синтаксического анализатора** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Разработка семантического анализатора** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Тестирование семантического анализатора** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Составление документации** |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Требования к проекту

4.1 История изменений документа

Таблица 3. История изменений

|  |  |  |
| --- | --- | --- |
| **Дата** | **Автор** | **Внесённые изменения** |
| 22.09.2023 |  | Исходная версия требований |
| 25.09.2023 |  |  |
| 1.10.2023 |  |  |
| 7.10.2023 |  |  |
| 12.10.2023 |  |  |

Программный продукт *“Транслятор C++ на Pascal”* предназначен для компиляции программ, написанных на языке С++, в программу на языке Pascal.

4.2 Подсистемы проекта

4.3 Требования к подсистеме «Пользовательский интерфейс»

1. Архитектура проекта

5.1 История изменений документа

5.2 Архитектурно-контекстная диаграмма системы

Заключение

В ходе курсовой работы были решены следующие задачи:

* Разработан регламент проведения инспекции;
* Разработана модель состояний задач;
* Разработан план проекта;
* Разработаны требования к проекту;
* Разработана архитектуру проекта;
* Разработана программа измерений проекта;
* Разработан перечень задач проекта;
* Разработаны рекомендации по кодированию;
* Разработан план тестирования проекта.

Список литературы

1. Ахо, А. Компиляторы: принципы, технологии и инструментарий компиляторов / И. В. Красиков. – М.: Вильямс, 1986. – 1184 с. ;
2. Бруно, К. Л. LLVM: инфраструктура для разработки компиляторов / К. Л. Бруно, А. Рафаэль. – М.: ДМК Пресс, 2015. – 342 с.;
3. Вирт, Н. Построение компиляторов / Н. Вирт. – М.: ДМК Пресс, 2010. – 192 с.;
4. Racket Documentation [Электронный ресурс]. URL: <https://docs.racket-lang.org/> (дата обращения 16.05.2020).