План реализации проекта

*«Там, где море»*

команды «T»

Оглавление

[1. Общие сведения о проекте 4](#_Toc476162543)

[2. Техническое задание 6](#_Toc476162544)

[2.1 Краткое описание проекта 6](#_Toc476162545)

[2.2. Решаемые задачи 6](#_Toc476162546)

[2.3. Функциональные и количественные ограничения 7](#_Toc476162547)

[2.3. Требования к эффективности 7](#_Toc476162548)

[2.4. Требования к точности 7](#_Toc476162549)

[2.5. Требования к аппаратному и программного обеспечению. 7](#_Toc476162550)

[2.6. Требования к использованию утилиты 8](#_Toc476162551)

[3. Модель команды разработчиков и распределение ролей 9](#_Toc476162552)

[4. Управление рисками 11](#_Toc476162553)

[5. План работы 13](#_Toc476162554)

# 1. Общие сведения о проекте

**Название проекта**:

**Оракул**: Крашенниников Сергей Вениаминович (требования) + Ф.А. Новиков (проектирование, реализация)

**Риски**: (см. раздел 4. Управление рисками )

**Vision/Scope**:

* Взаимодействие лица, представляющего интересы заказчика, с менеджером проекта.
* Представление фронтменом проекта отчета по всем этапам реализации проекта.
* Взаимодействие преподавателя с любым участником команды по вопросам проектирования и реализации.

**Цели:**

1. Реализовать проект в соответствии с техническим заданием (см. раздел 2.Техническое задание)
2. Получение всеми участниками команды опыта разработки прикладного программного обеспечения и опыта командного взаимодействия.

**Критерии:**

* Оценка преподавателя курса.
* Реакция заказчика
* Вдохновленное настроение участников команды по окончании работы над проектом.

**Протокол / Репозиторий:** GitHub.

**Коммуникация:**

1. Внутри команды: очные встречи, удаленное общение (vk.com), динамический план реализации

2. С заказчиком: предоставление прототипа программного продукта, финальная презентация проекта.

**Дисциплина:** В случае несвоевременного выполнения поставленной задачи участником команды при условии отсутствия на то уважительной причины 10 ударов розгами за каждый просроченный день.

**Технологии**: Qt,OpenCV,UML, С++.

**Модель команды**: (см. раздел 3. Модель команды разработчиков)

# 2. Техническое задание

## 2.1 Краткое описание проекта

Заказчик проекта – международная компания «Транзас», производитель высокотехнологичного оборудования, программного обеспечения и системной интеграцией для морской отрасли.

Основная цель проекта «Там, где море» заключается в создании утилиты, предназначенной для обработки изображений, получаемых с радара и последующем анализе данных с целью предоставления информации о движении наблюдаемых при помощи радара объектов.

## 2.2. Решаемые задачи

В файловую систему поступают с различной частотой файлы, содержащие радарное изображение. В одном файле содержится одно радарное изображение.

Радарное изображение – цифровое изображение круга, содержащее на нейтральном фоне контуры объектов. Каждое изображение снабжено географической координатой своего центра.

Проведена первичная обработка изображений, то есть устранен аппаратный шум.

Каждый файл имеет уникально имя в соответствие с установленным форматом.

Требуется определить:

1. для каждого изображения контуры пятен объектов (вопросы, связанные с форматом представления данных)

2. по последовательности изображений набор движущихся объектов с указанием характера и параметров движения.

## 2.3. Функциональные и количественные ограничения

Вывод времени осуществляется в шкале UTC с указанием часового пояса.

Вывод координат осуществляется в стандартных географических координатах (градусы, минуты, секунды, доли секунд).

Утилита должна работать с любым количеством входных файлов любого формата.

Ограничение на размер и количество входных файлов определяется размером оперативной памяти исполняющего компьютера. Требования по размеру использованной памяти будут уточнены позже.

## 2.3. Требования к эффективности

Скорость обработки входных файлов должна быть не меньше скорости их поступления.

## 2.4. Требования к точности

Точность определения координат и скоростей объектов зависит от качества поступающих радарных изображений, их количества, а также от внешних условий, при которых осуществлялись наблюдения. В зависимости от этих параметров координаты объектов должны содержать доверительный интервал для уровня вероятности 90%.

## 2.5. Требования к аппаратному и программного обеспечению.

Утилита должна быть разработана для программных платформ, имеющих возможность выполнения кода, написанного на C++. Это включает:

* выполнение на 32- и 64-битных ОС Linux и Windows (win8/win10) для ПК;
* статическую и динамическую линковку с программными комплексами на C, C++.

## 2.6. Требования к использованию утилиты

Утилита должна быть полностью документирована, а также предоставлять в режиме runtime информацию по ее использованию.

# 3. Модель команды разработчиков и распределение ролей

В качестве модели команды разработчиков была выбрана иерархическая модель. К преимуществам данной модели относятся

* Возможность четкого распределения обязанностей
* Легкость расширяемости команды

К недостатком данной модели стоит отнести

* Несколько большую трудоемкость незапланированных коммуникаций между участниками команды по сравнению с другими моделями (бригада главного программиста, модель равных)

Поскольку размер команды Т небольшой, то недостатки иерархической модели не должны оказать значительного влияния. Поэтому участники команды считают, что выбранная модель позволит достичь наибольшей производительности командной работы.

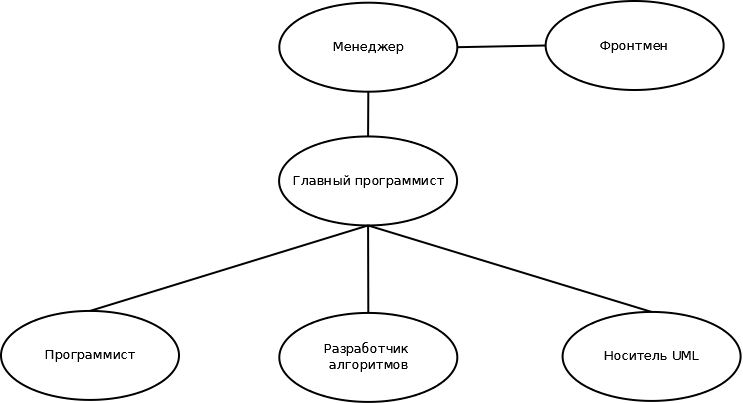


Рисунок 1 – Схема взаимодействия команды Т

Распределение ролей в команде Т осуществляется в соответствие с компетентностями ее участников. Однако в процессе работы над проектом возможны будут корректировки ролей из соображений равенства вклада, вносимого в проект каждым из участников. Распределение ролей приведено в табл. 1.

Таблица 11. Распределение ролей в команде Т

|  |  |
| --- | --- |
| Роль | Участник |
| Фронтмен | Н. Лысикова |
| Менеджер | Р. Тхакушинова |
| Первый носитель UML | Н. Пестова |
| Разработчик алгоритмов | А. Зефиров |
| Главный программист | М. Выменец |
| Программист | А. Цуканов |

# 4. Управление рисками

Риском проекта называется любое событие или условие, которое может оказать негативное влияние на ход выполнения проекта. Под управлением рисками понимается процесс их предсказания, анализа и эффективной превентивной работы над ними. Далее приведен список выявленных рисков и планы по уменьшению их отрицательного влияния.

*Технологические риски*

К технологическим рискам можно отнести несоответствие запланированного инструментария поставленной задаче; такой риск минимален при условии неизменного состояния технического задания.

*Риски, связанные с персоналом*

В связи с особенностью данного проекта могут потребоваться знания работы алгоритмов обработки изображений, а также узкопрофильные знания предметной области. Это накладывает большую ответственность на разработчика алгоритмов и программистов. На данном этапе работы над проектом оценить данный риск сложно.

*Организационные риски*

К организационным рискам относят возникновение ошибок планирования.

Отведение одной из ролей в команде для менеджера позволяет смягчить последствия ошибок планирования, поскольку в таком случае участник команды, исполняющий эту роль, может контролировать качество работы в режиме реального времени, а значит применять технику адаптивного менеджмента, тем самым смягчая и даже нивелируя эффект ошибок планирования.

*Риски, связанные с требованиями заказчика*

Мы рассчитываем на то, что поставленные требования в техническом задании не изменятся вплоть до завершения работы над проектом. Любые корректировки, касающиеся расширения функциональности, могут быть рассмотрены как постановка задач для нового проекта.

# **5. П**лан работ**ы**

Формирование плана-графика проекта позволяет эффективно спланировать работу над проектом. Результатом его формирования является календарный план работ, позволяющий учитывать этапы и результаты работы над проектом, определять длительность работ и распределять обязанности участников.

В таблице 2 показ план-график первого этапа работы над проектом.

Таблица 22. План-график первого и второго этапа (18.02.2017- 25.04.2017)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Роль | Задача | Артефакт | Сроки | Результат | Комментарии |
| Менеджер | Составить план реализации проекта «Там, где море» | План реализации  проекта (.doc файл) | 4.03 | Выполнено |  |
| Уточнить окончательное ТЗ | Модернизация плана реализации проекта | 11.03 | выполнено | ТЗ может уточняться в дальнейшем, но требования по функциональности и представлению данных останутся неизменными |
| Фронтмен | Представить план проекта на занятии | - | 11.03 | выполнено |  |
| Разработчик алгоритмов | Проанализировать алгоритмы  обработки изображений | - | 11.03 |  |  |
| Главный программист | Осмыслить программную архитектуру и используемые технологий | - | 11.03 | выполнено |  |
| Носитель UML | Разработать диаграмму деятельности, описывающий весь процесс работы утилиты | диаграмма взаимодействия | 11.03 | выполнено |  |
| Носитель UML и Главный программист | Разработать диаграмму размещения исходных файлов проекта | диаграмма размещения | 11.03 | выполнено |  |
| Разработчик алгоритмов | Освоение библиотеки OpenCV. Тестирование ее на реальных изображений. | Отчет перед командой о результатах | 18.03 |  |  |
| Разработчик алгоритмов | Написание модуля обработки изображений, предоставляющий API остальным модулям. | Библиотека обработки изображений | 25.03 |  |  |
| Разработчик алгоритмов | Документирование API |  | 1.04 |  |  |
| Разработчик алгоритмов | Написание модуля, формирующего log-файла |  | 8.04 |  |  |
| Разработчик алгоритмов | Написание модулей обработки изображения | алгоритмический модуль | 22.04 |  |  |
| Главный программист | Освоение Qt | - | 18.03 |  |  |
| Главный программист | Проектирование архитектуры программы. Написание оберток модулей. | Программный модуль | 25.03 |  |  |
| Главный программист | Написание модулей взаимодействия с интерфейсом и с алгоритмами обработки изображений. | Программные модули | 1.04 |  |  |
| Главный программист | Объединение модулей проекта | Модули проекта | 22.04 |  |  |
| Фронтмен | Освоение Git | - | 18.03-  25.03 |  |  |
| Менеджер | Освоение Git | - | 18.03 |  |  |
| Носитель UML | Создание интерфейса программы | Файлы формата .qml с интерфейсом | 25.03 |  |  |