**Министерство образования и науки Кыргызской Республики**

**Кыргызский государственный технический университет**

**им. И. Раззакова**

**Факультет информационных технологий**

**Кафедра «Программное обеспечение компьютерных систем»**

**ОТЧЕТ**

**Дисциплина «Процесс проектирования и разработки ПО»**

**Тема: «Расчет примерного времени прибытия транспорта»**

**Выполнил студент группы ПИ(б)-1-16:**

**Абакиров Нурсултан**

**Проверил(а):**

**Мусина И.Р.**

Table of Contents

[Цель 3](#_Toc533795522)

[Контекст 3](#_Toc533795523)

[Функциональные требования 3](#_Toc533795524)

[Нефункциональные требования 3](#_Toc533795525)

[Use Case Diagram 4](#_Toc533795526)

[Аналоги 5](#_Toc533795527)

[V1 6](#_Toc533795528)

[Сервер 6](#_Toc533795529)

[Устройсво 6](#_Toc533795530)

[V2 7](#_Toc533795531)

[Цель – улучшить предсказания прибытия транспорта 7](#_Toc533795532)

[Обработка входных данных 7](#_Toc533795533)

[Расчет стоимости продукта с помощью COCOMO II 10](#_Toc533795534)

[Диаграмма классов 11](#_Toc533795535)

[Диаграмма последовательности 12](#_Toc533795536)

[Результаты 13](#_Toc533795537)

[13](#_Toc533795538)

**Расчет примерного времени прибытия транспорта**

# **Цель**

Рассчитать примерное время прибытия транспорта к остановочным пунктам.

# **Контекст**

В каждом транспорте есть:

* мини-компьютер *orangepi*
* gps датчик
* система голосового оповещения, которая уже определяет, когда транспорт подъезжает и отъезжает от остановки
* База данных маршрутов и остановок

# **Функциональные требования**

На устройсте:

* рассчитать время, когда транспорт прибудет на каждую последующую остановку из заданного набора
* отправить запрос на сервер с расчетами

На сервере:

* принимать запросы от устройств и сохранять данные, до тех пор, пока расчеты не устареют (до след. остановки)
* по запросам отправлять примерное время прибытия определенного транспорта к остановкам
* по запросам отправлять примерное время прибытия транспорта к определенной остановке, фильтровать по маршрутам
* написать документацию для разработчиков (клиентов)

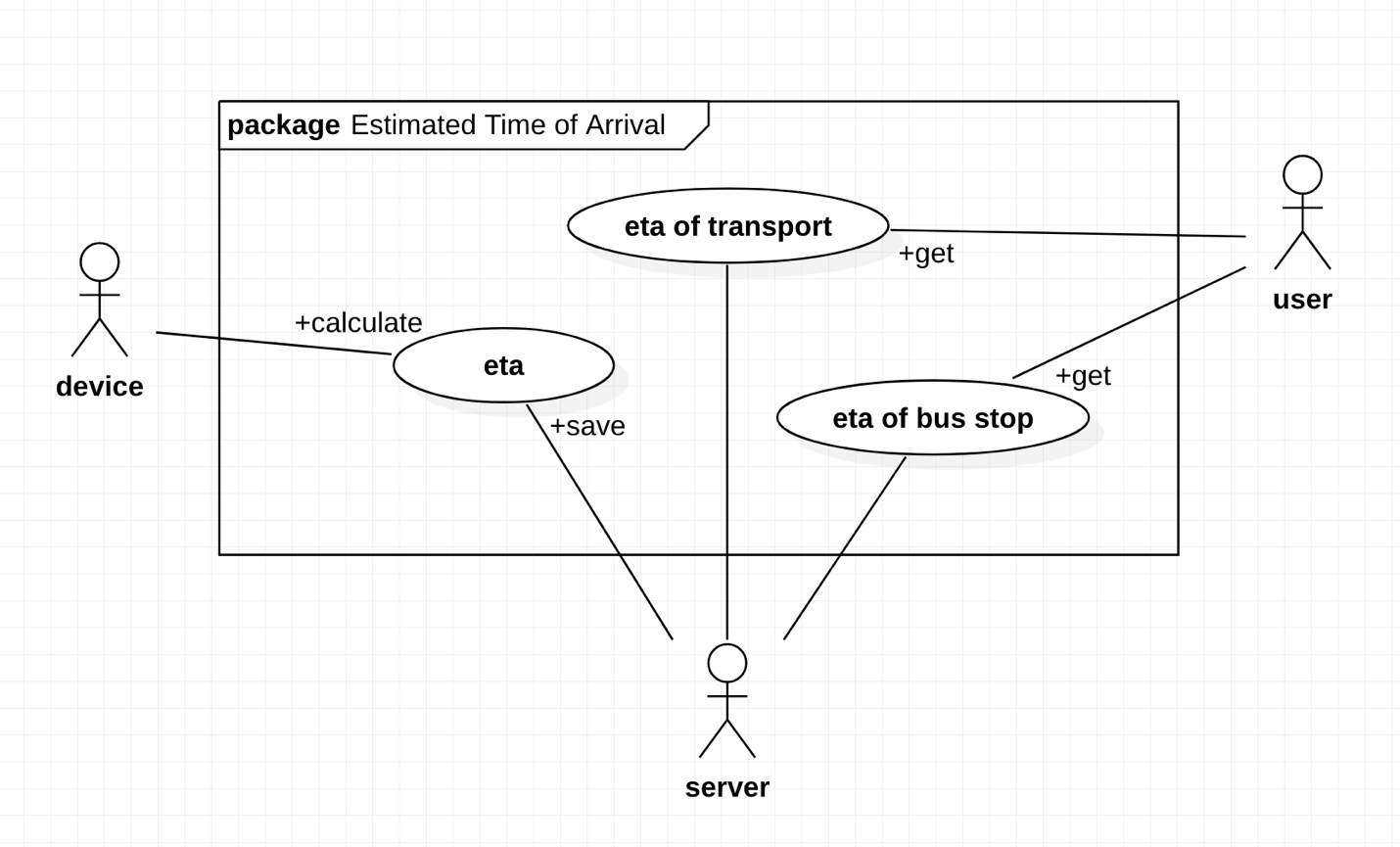
# **Нефункциональные требования**

* Работа скрипта не должна превышать 200 мб оперативной памяти
* Скорость расчетов не должна превышать 3 секунд

**Исходные данные:**

* Список остановок:
  + Координаты (долгота, широта)
  + Название
  + Уникальный номер остановки
* Время
* Триггера:
  + Момент прибытия на остановку
  + Момент отъезда от остановки

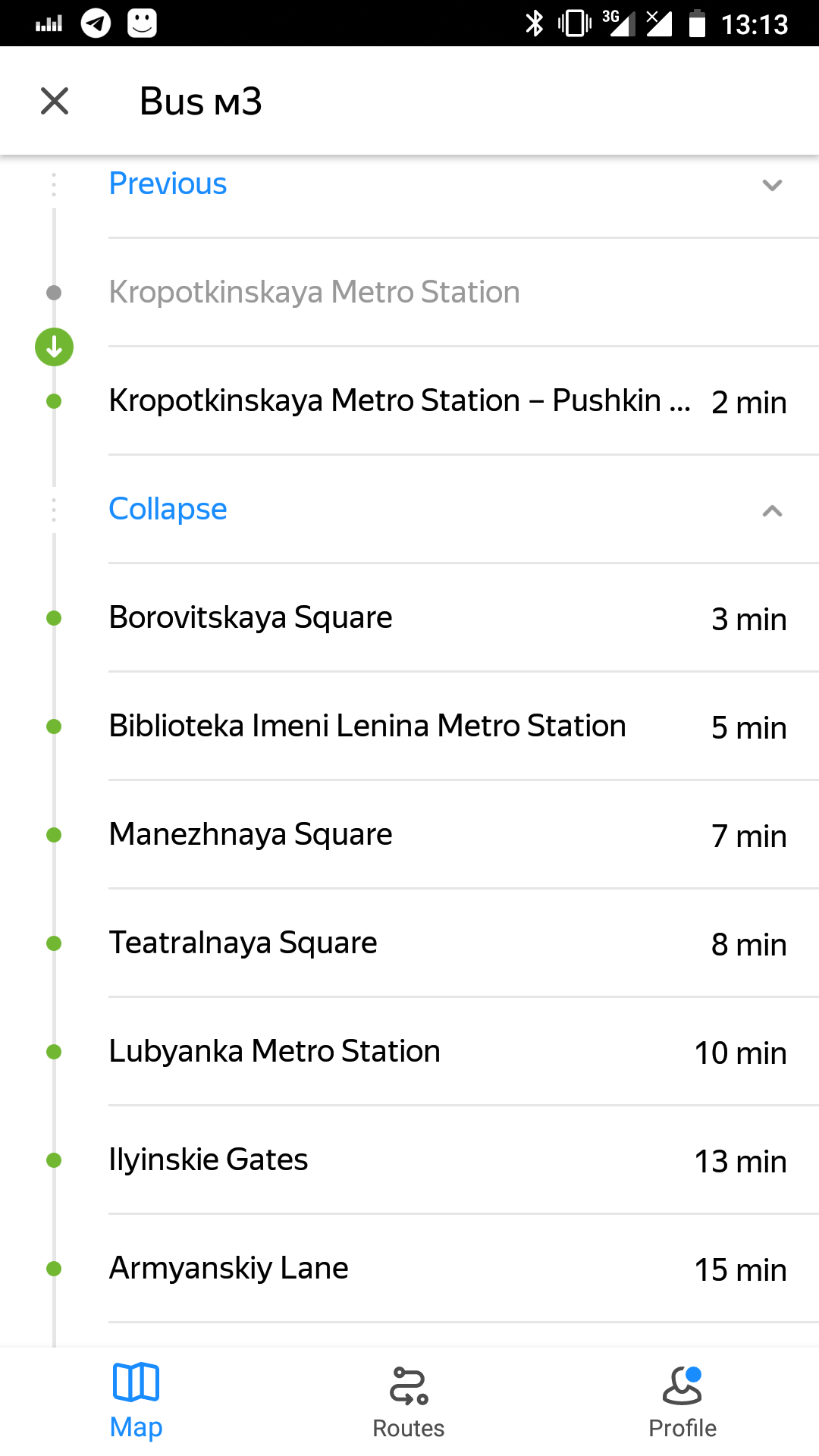
# **Use Case Diagram**

****

# **Аналоги**

**Yandex.Transport**

Является внутренней системой



В России общественный транспорт едет по расписанию (колоссально облегчает расчеты). Их задача была уменьшить погрешность.

Используя алгоритмы кластеризации, они решили свою проблему. Расчеты проходили на внутренней системе искусственного интеллекта.

# **V1**

Задача первой версии - создать альфа версию продукта, то есть рабочую базу для дальнейшего улучшения.

## **Сервер**

* Подготовить 3 API endpoints:

1. Принимать с устройств расчеты
2. Отправлять расчеты по определенной остановке
3. Отправлять расчеты по определенному транспорту

* Организовать хранение расчетов на определенный срок для дальнейшего оперативного использования.
* Описать как работать с API в документацию

## **Устройсво**

Определять время прибытия к остановке на основе среднестатистического времени поездки этого отрезка маршрута

* Логировать каждый проход промежутков между остановками.
* Делать выборку из логов по фильтрам:
  1. Промежуток между остановками
  2. день недели
  3. четверть дня
* После расчета отправить на сервер HTTP запрос с данными на подготовленный API

# **V2**

## **Цель** – улучшить предсказания прибытия транспорта

Сейчас

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Транспорт (id) | Маршрут (id) | Остановка1 (id) | Остановка2 (id) | Время (сек) |
| 1 | 2 | 234 | 332 | 43 |

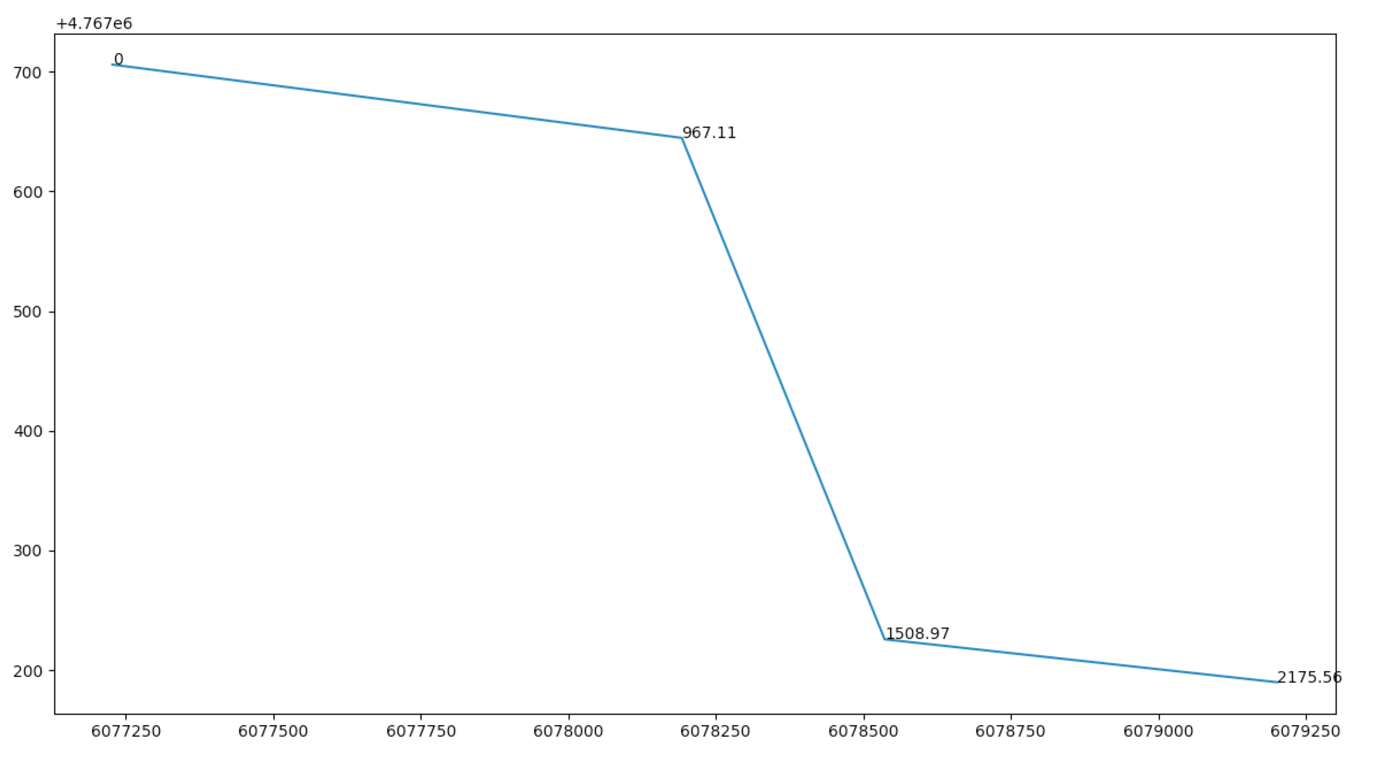
Надо перевести в

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Транспорт (id) | Маршрут (id) | Начало (м) | Конец (км) | Время (сек) |
| 1 | 2 | 50 | 120 | 43 |

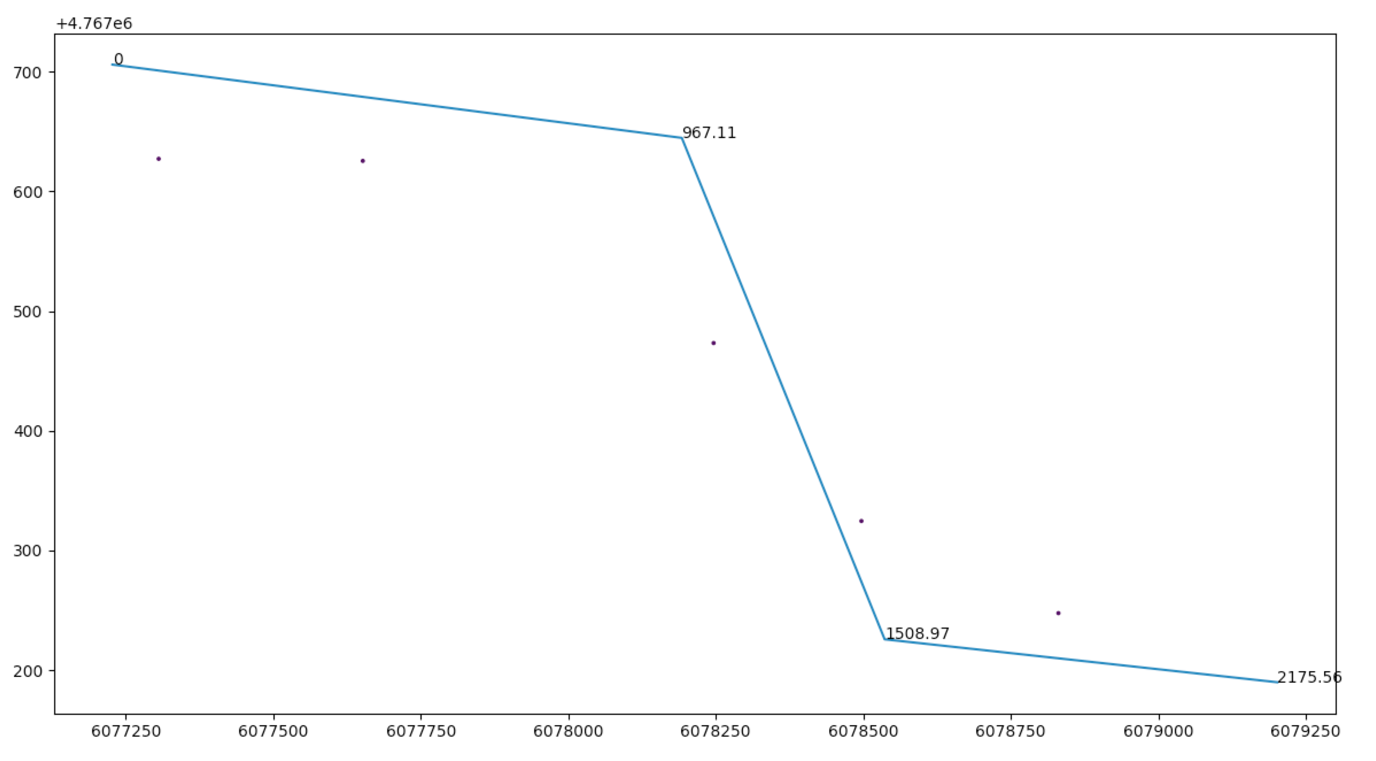
## **Обработка входных данных**

1. Сплющить координаты

Координаты (в радианах) в метры

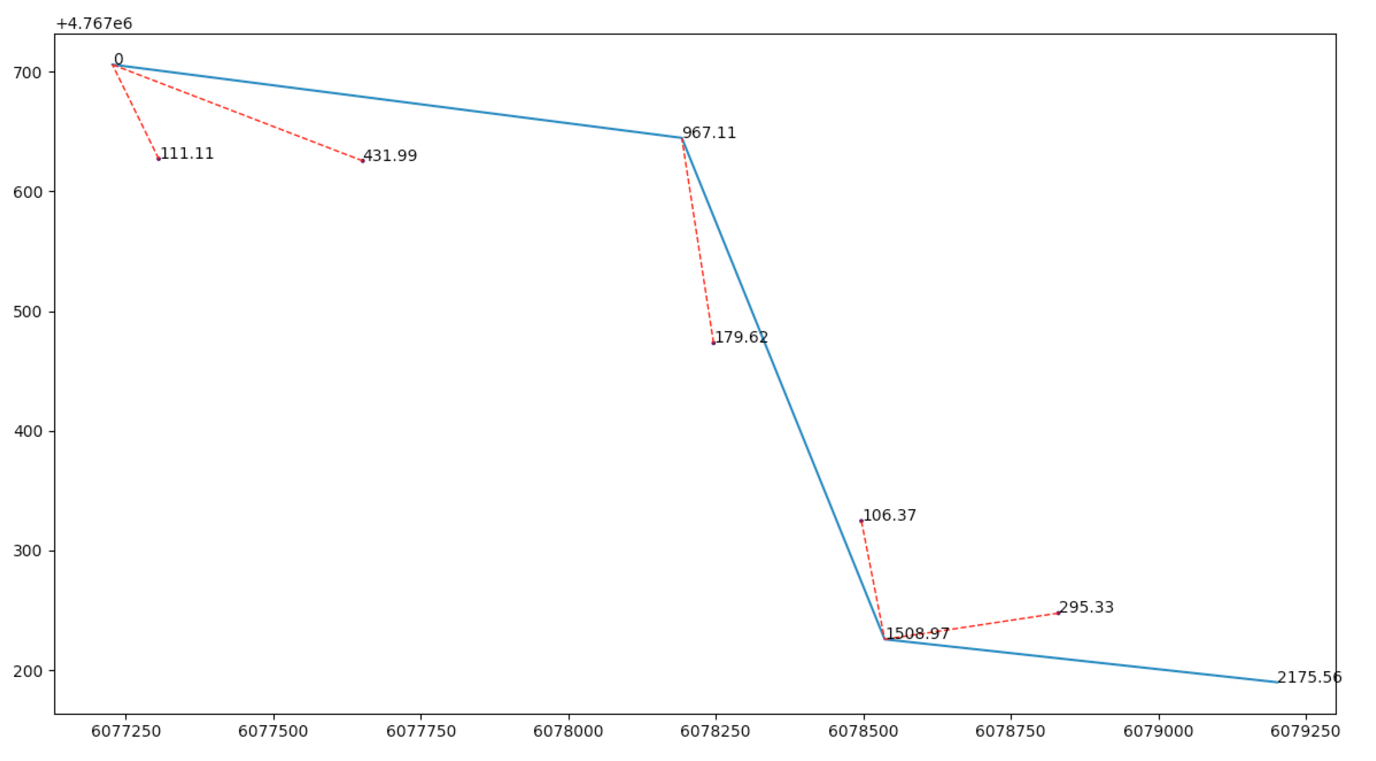


1. Спроецировать координаты (остановок, транспорта) для определие расстояния от начала маршрута в метрах



Так как маршрут – это набор координат, найдем ближайшую координату маршрута к искомой используя евклидово расстояние

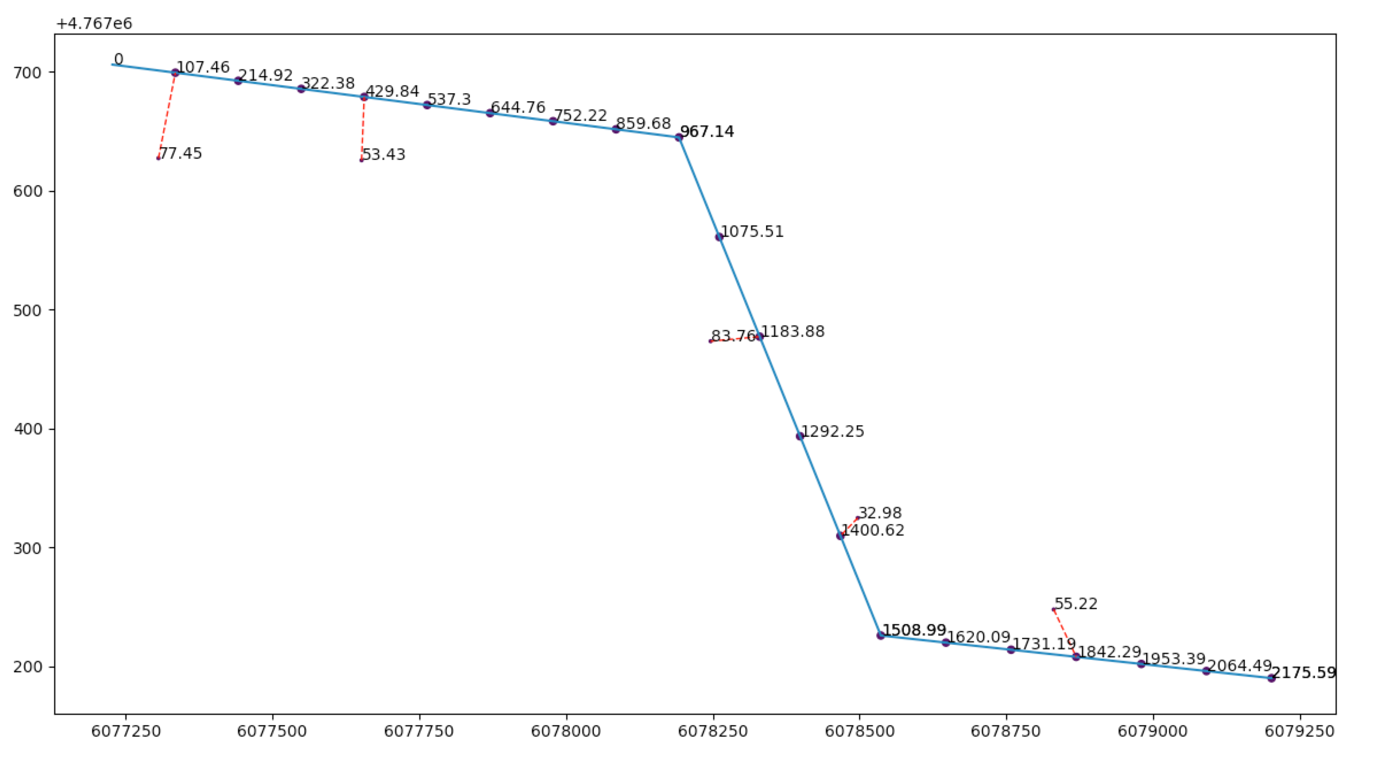
**«Евклидова метрика** (*евклидово расстояние*) — [метрика](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D1%80%D0%B8%D0%BA%D0%B0_(%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%8F)) в [евклидовом пространстве](https://ru.wikipedia.org/wiki/%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4%D0%BE%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE) — расстояние между двумя точками евклидова пространства, вычисляемое по [теореме Пифагора](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%9F%D0%B8%D1%84%D0%B0%D0%B3%D0%BE%D1%80%D0%B0).»



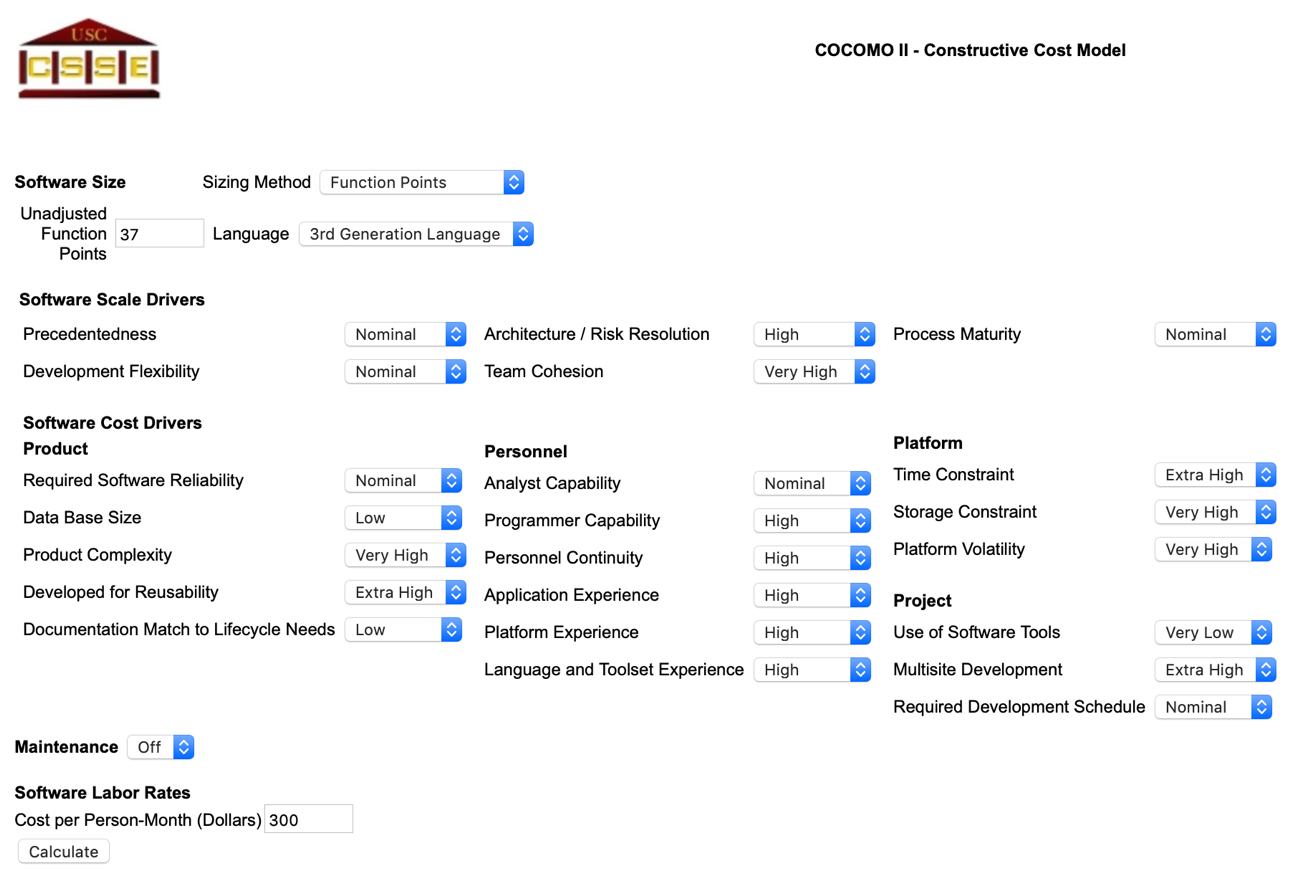
Из изображение выше видим, что не хватает координат к маршруту

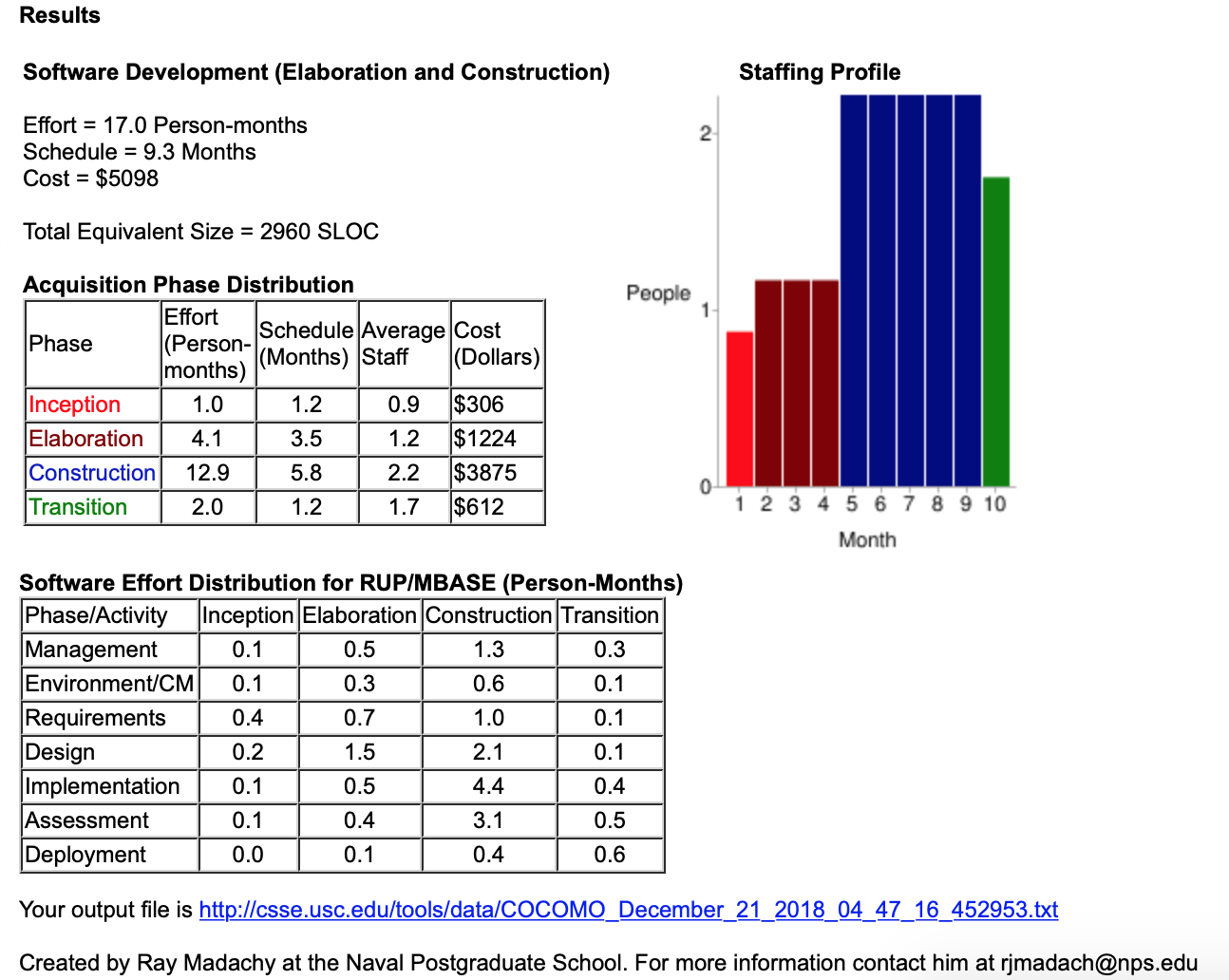
«**Лине́йная интерполя́ция** — [интерполяция](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D1%8F) алгебраическим [двучленом](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D1%87%D0%BB%D0%B5%D0%BD) ***P1(x) = ax + b*** функции ***f***, заданной в двух точках ***x0*** и ***x1*** отрезка ***[a, b]***. В случае, если заданы значения в нескольких точках, функция заменяется [кусочно-линейной функцией](https://ru.wikipedia.org/wiki/%D0%9A%D1%83%D1%81%D0%BE%D1%87%D0%BD%D0%BE-%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F).»

, где

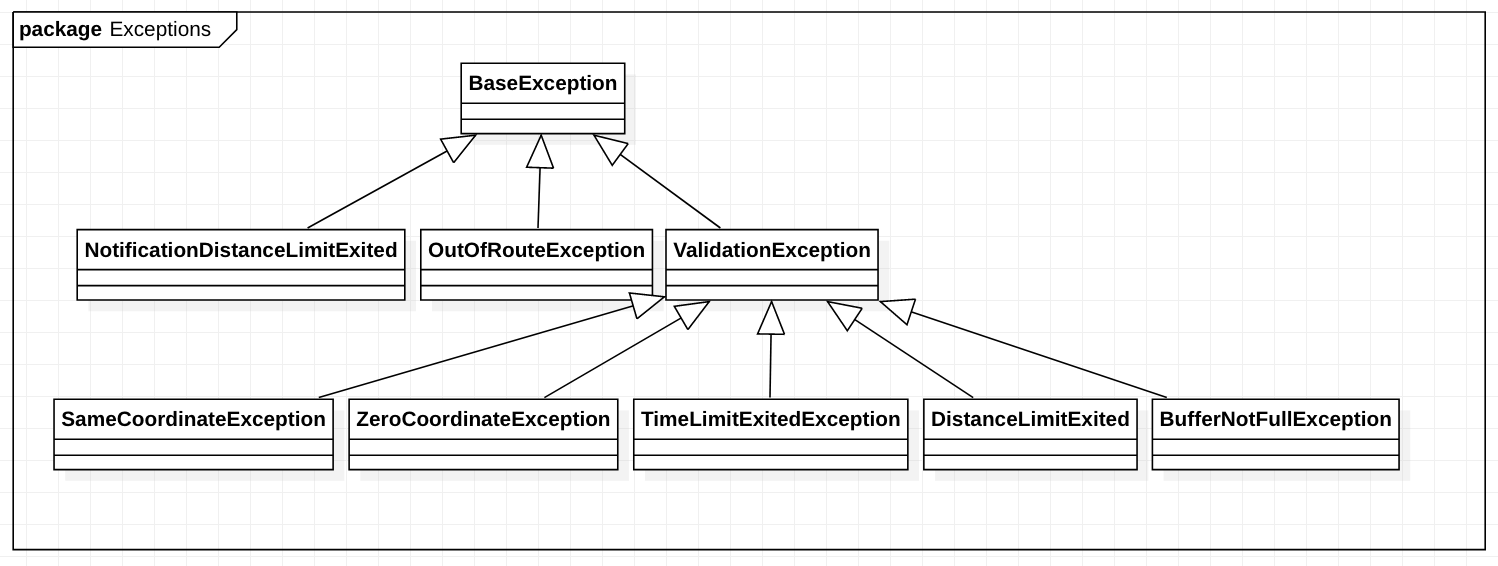
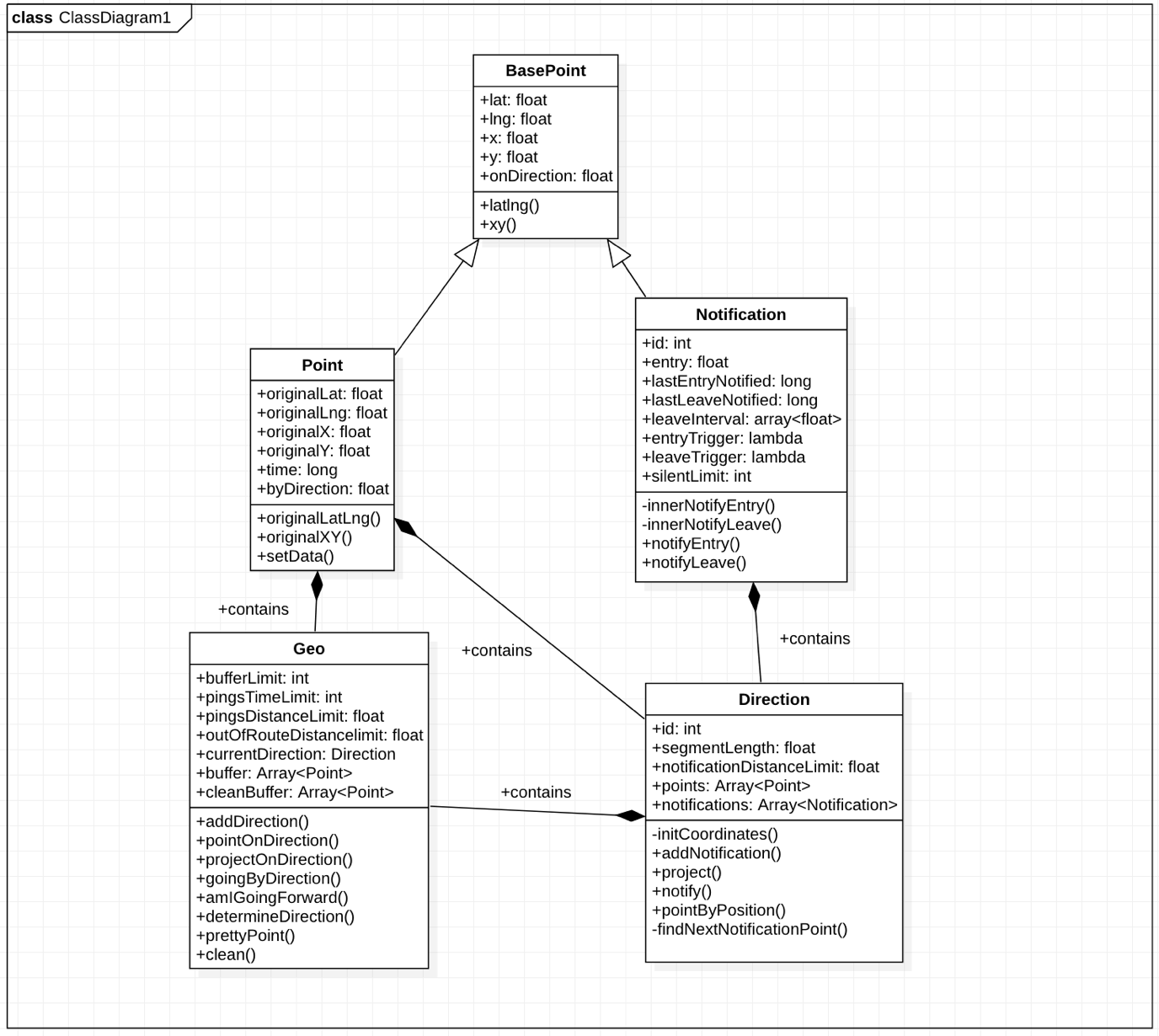


# **Расчет стоимости продукта с помощью COCOMO II**

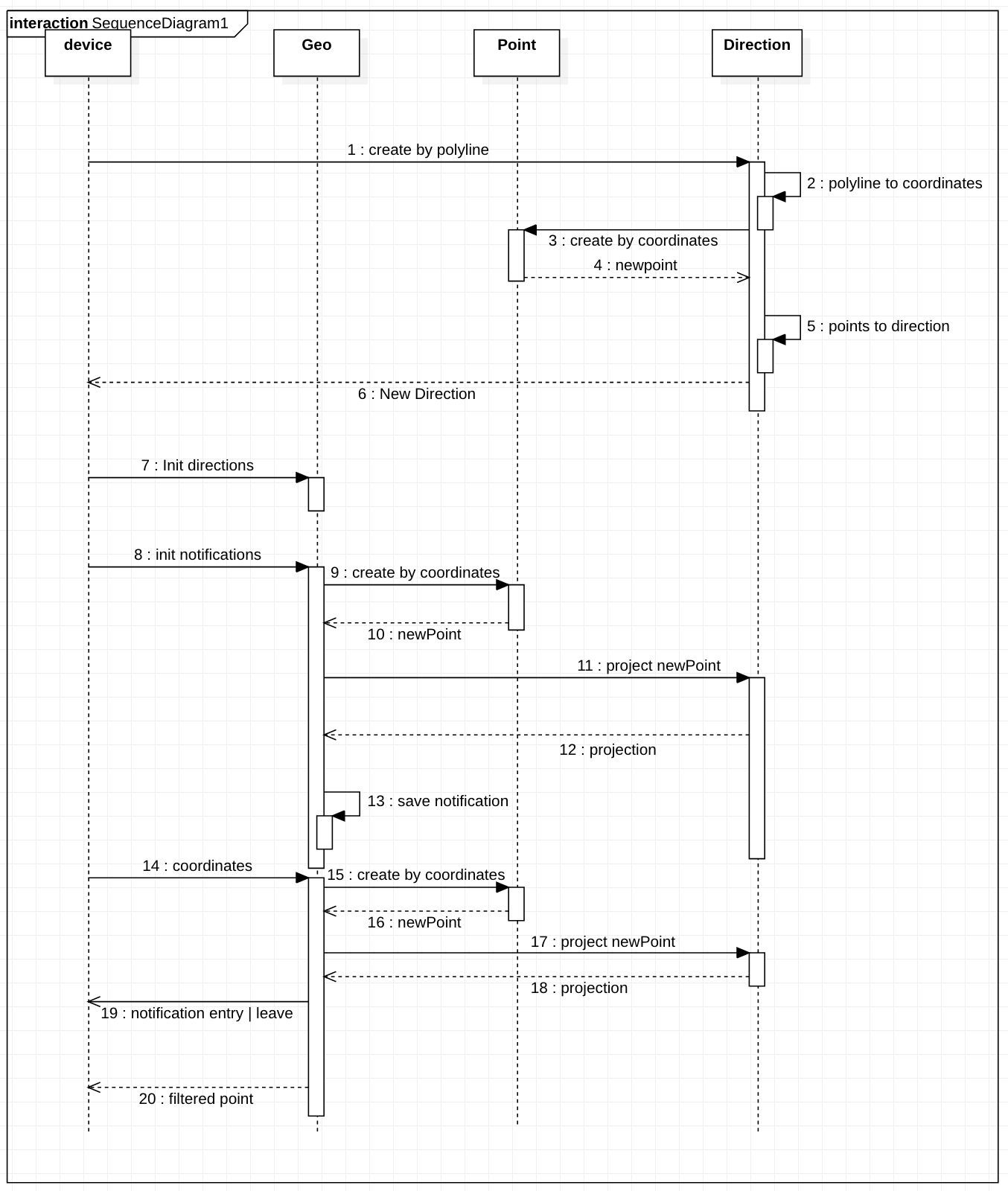




# **Диаграмма классов**



# **Диаграмма последовательности**



# **Результаты**

# 

