Автотесты

При написании автотестов и подбора параметров я руководствовался:

- Граничными значениями, выбирая значения на границе, по обе стороны, в середине диапазона.
- Позитивные и негативные проверки, с учетом границ (например для невалидного расстояния проверяется 0, -1 и меньшие значения).
- При небольшом наборе входных данных я использовал максимально возможное число комбинаций, так как в данном проекте они будут выполнены за малое время.
- При большом наборе (таких как проверка комбинаций валидных входных значений) я использовал *pairwise*, так как это дает адекватное покрытие при сокращении набора тестовых данных. Для тестов сгенерировал их с помощью утилиты и экспортировал в csv-файл (именно потому использовал передачу параметров тестам через csv-файл).
- Ожидаемые значения получены **не** с помощью функции, которая тестируется. На больших проектах использовал бы собственную функцию или утилиту, выборочно просчитав вручную отдельные значения, чтобы убедиться в правильности.

На проекте автотесты я бы запускал в процессе сборки проекта в качестве отдельного шага. В случае падения тестов вся сборка падает с генерацией отчета. Проект собирался бы с помощью CI, Allure-отчет прикладывался бы к сборке в качестве артефакта.

Настройку CI предпочтительнее делать как *infrastructure-as-a-code*, описывая джобы и пайплайны кодом, в том числе и шаги по генерации отчетов и, опционно, их отправки в месседжер и т.д.

Также можно настроить хранение отчетов и сборку последующих с историей предыдущих прогонов.

При увеличении проекта и соответственно набора автотестов имеет смысл разделять тесты на сьюты, по приоритетам, выделить acceptance набор.