

# MSD Reading Questions self-assessment bot – Assignment 3

Kristina Alyabeva, Saveliy Chertkov, Kirill Korikov

October 2024

You can also read the document in a different format

## 1 Review with the customer

We sent the customer a file in advance with a description of the strategic plan and the commands that will be used in the bot. At the meeting, the customer explained that the documents did not tell him anything to understand the project. For him, the best solution to discussing the work done is the result of product development. In order to coordinate the main development steps with him, we need to make a detailed list of all the functions that our bot will implement to check RQ. Also, set priorities for the development of these functions. This list should be sent to coordinate tasks for the MUP and further work. The main tasks allocated by the customer at the moment for the first stage of project development (MUP):

- Checking for citations in each question
- The ratio of the number of citations in the response to its volume
- The number of explicit examples given in the work
- Checking the availability of the References section

## 2 Improve the product vision

After meeting with the customer, we found out that to check RQ, all tasks related to formatting verification can be omitted. During the check, all students apply for documents in different ways and this item is not significant compared to the main task of the bot.

Use case diagram

To compile User stories, we conducted Customer Development among MSD students and tutors. The results of the student questionnaire.

Also, according to the tutors' garbage, they asked to add the following functions:

- Checking the presence of an example
- Checking citations for relevance (assuming IEEE referencing style)

Based on this, a User story was compiled.

List features based on user stories:

- Checking for GPT generation
- Checking the coherence of the text
- Checking for references
- Checking the correctness of the answer
- Checking if all the questions have an answer
- Standard formatting
- Predicting the grades
- Checking the presence of an example
- Checking citations for relevance (assuming IEEE referencing style)

All functions described at this stage are covered by user stories.

### 3 Minimum Usable Product

At this stage, the following functions are implemented in the bot:

- Checking the availability of a section with references
- Checking for quotes in the answer
- Checking the ratio of the answer and the number of quotes in it

The results of the work are shown in the screenshots, and you can also check them in the repository.

The deployment of the application is specified in the README in the repository. At the moment, it is calculated that the bot will be deployed locally. In the future, this item may be reworked.

### 4 Sprint cycle

- Describe a single sprint cycle. Describe processes related to a sprint cycle.  
Initial plan to have each sprint cycle to be 2 weeks long had to be reworked to adapt to assignments deadlines. Then, the current sprint cycle looks as following:

**Tuesday** — sprint planning: team call/meeting to discuss the tasks to be done.

**Wednesday-Thursday** — implementation: coding, model training.

**Friday** — sprint review: interview with a customer, writing documentation(assignment).

**Monday** — sprint retrospection: review of completed work and comparison with other teams.

- Pick an appropriate git workflow process including the branching rules, commit messages format, pull request review, and other aspects. Make sure it works for your team. Make sure all parts of the workflow have a purpose. Feel free to draw inspiration from existing processes but describe the purpose of each part if you end up using it.

Given the project size, our time didn't see any need in using complex branching. Current **workflow** is feature-driven. Each new feature is being developed in a separate **branch**, named after their respective feature, which is merged into "main" branch after completion.

**Commits** are named shortly to tell which operation was done on what part of the code.

Our team didn't see any value in using a **Kanban** process, more so, it was an irritating **overhead**, used in retrospect.

- Conduct sprint planning.

Sprint planning was done in the Telegram call, plans were written after that:

- implement length check (1 sentence is too short)
- implement number of references check
- collect feedback (questionnaire)
- extra: implement examples count check

- Describe the sprint goal.

Completion of the Iteration 1

- Populate the sprint backlog.

- Execute the sprint. There should be completed sprint artifacts like tasks, code, merge requests, etc.

Iteration 1 commits. MUP of the bot.

- Conduct a sprint retrospective. Describe the results of the retrospective.

- Sprint retrospection wasn't done at the moment of writing this document because of the sprint structure our team decided on. So we had to conduct a trial of this process earlier.

- During the trial retrospection our team once again noted the issues in conducting interviews. Discussed change of roles (**Saveliy** now being the developer, and **Kirill** instead taking more planning and management tasks).
- Also, our team saw that the main source of the requirements - students, is not being elicited from, so initial questionnaire was planned and then developed.