Decision GPS Version 5

AI Integration - Functional Specifications

# Document reference: R5-FS-AI v0a

**R5**: The *Decision GPT R5* is the first version integrating AI services.

**FS**: Functional Specification. This document does not describe the physical implementation and testing, but focuses on the objectives and AI prompts construction.

**AI**: Artificial Intelligence. This is the new functionality being described.

**V0a**: Version Zero of the document. Version zero is the first draft. Odd numbers are for work-in-progress versions of the document. Even numbers are for stable versions that can be shared with developers and QA testers.

For work-in-progress versions, a letter is added to differentiate the updates. The very first draft of a document is then V0a.

# Document Purpose

In recent years, the usability, reliability, and economics of Artificial Intelligence have greatly improved thanks to the Large Language Models [LLM] using the Transformer Architecture introduced by Google in 2017.

This innovation has led to a massive adoption of LLM platforms such as ChatGPT.

Realizing that it could be an important success factor for the GPS app, it was decided to integrate ChatGPT at different steps in the 8-step model of factor creation, as well as on the Report Card.

# Document Versions

|  |  |  |
| --- | --- | --- |
| Version | Updates | Author |
| V0a | Document Creation | C. Rossello |
| V0b | Update of the prompts after the Aug 26th 2025 workshop | C. Rossello |
| V0c | Added S4 prompt | C. Rossello |

# General Context

The AI usage is not free of charge. For now, we are using ChatGPT, and it comes with a cost. One of the objectives of the AI Integration Architecture is to optimize the overall cost. OpenAI's ChatGPT and GPT-4 models charge users based on the number of tokens processed, with pricing tiers varying based on usage volume and subscription type. Google's AI tools, available through Google Cloud Platform, typically charge based on API calls, computation time, and the specific services used.

# ChatGPT integration

## ChatGPT account

For development and introduction purposes, we are using the private ChatGPT account of the developer: ID = christopherossello@hotmail.com.  
This will change with the usage in volume by the production server. A new Panthera account will be created, and the company will be charged directly.

The update is done simply by replacing the API key in the server code.

## AI model

The current version (5.34) of the server code is using the model « gpt-3.5-turbo » which is a good compromise between performance and cost. Later, an update to ChatGPT4 is planned.

## API connection

The server is using Node.js in an Amazon Web Services Linux server in the Ireland cloud.

OpenAI is providing a Node.js module called « openai » to simply connect to its API using a key that is linked to an account. There is no technical difficulties with this implementation.

# Factor Step One

## Objectives

The AI supports the user by answering the seven Hypothesis Quality questions with a Yes or No answer, plus general feedback.

The user will use the AI inputs to benchmark her/his own answers to the same questions in order to produce the best Hypotesis possible for the factor to work on.

## Challenges

Because the UI is going to change the status of each question individually by selecting Yes or No, it is expecting 8 different inputs from the AI, one boolean for each question, and a text for the feedback field.

Getting this structured information from one call to the API, using only one prompt, is not possible. The challenge is then to develop a more complex prompt architecture allowing the individualisation of each question, while maintaining overall feedback.

A second challenge is to build prompts dynamically using the hypothesis input from the user. The prompt template must be built in a way that final prompts for the 8 calls to the API merge smoothly the Hypothesis with questions asked to the AI model.

The third challenge is to take into account the results from the 7 calls for the 7 questions during the construction of the feedback prompt. It is not easy, as the code has to wait for all the answers before starting the feedback prompt.

## Solutions

The best solution to get 8 differentiated answers is to build 8 different prompts.

Prompts for the questions

#### Commonalities between the prompts

For all questions, the prompt must give the context and the instruction to answer with Yes or No only. The answer will be analysed by the server to extract the string Yes or No and send a boolean True/False to the UI.

#### Prompts used for each questions

**Focused prompt** = "You are a quality control specialist. Is it a focused analysis to declare that "+***HYPOTHESIS***+"? Answer only by yes or no. The broader the scope, the more difficult to assess. A focused scope also indicates a higher degree of clarity on the matter before starting the assessment process. A focused focus makes it easier to stay centered and concentrated in your assessment.".

**Specific prompt** = "You are a quality control specialist. Is it a specific analysis to declare that "+***HYPOTHESIS***+"? Answer only by yes or no. Vague hypotheses are easier to work with as they are rhetorically ambiguous. One can find what one wants to find during the assessment process. Be specific when defining indicators, phenomena, or times. Make your hypothesis unambiguously explicit and definite.".

**Original prompt** = "You are a quality control specialist. Is it an original analysis to declare that "+***HYPOTHESIS***+"? Answer only by yes or no. Your comparative advantage depends on whether you explore something original or common, is it an opportunity or a risk. The more original, the more likely to be ahead of others, as you will explore market phenomena closer to the knowledge frontier.".

**Researchable prompt** = "You are a quality control specialist. Is it a researchable analysis to declare that "+***HYPOTHESIS***+"? Answer only by yes or no. Determine if it builds on observable evidence. Something can be original, but hardly observable, where qualitative and/or quantitative evidence is not the only effort to fully find out.".

**Feasible prompt** = "You are a quality control specialist. Is it a feasible analysis to declare that "+***HYPOTHESIS***+"? Answer only by yes or no.".

**Arguable prompt** = "You are a quality control specialist. Is it an arguable analysis to declare that "+***HYPOTHESIS***+"? Answer only by yes or no. When starting the assessment process to test your investment hypothesis, are you expecting that you have to cross legal or ethical lines to reach a validation/falsification point?".

**Relevant prompt** = "You are a quality control specialist. Is it a relevant analysis to declare that "+***HYPOTHESIS***+"? Answer only by yes or no. Determine if it matters to your investment objective. Given all of the above, determine if the investment hypothesis is worth.".

### Prompt for the feedback

#### Cascading prompts architecture

The idea is to inject answers for the question into the feedback prompt construction using the following architecture :

Build prompts and made API call for each questions

Send S1 Hypothesis to server

Make a new call to the API

Build a prompt for the feedback

**POSITIVE** = List of questions with a positive answer.

**NEGATIVE** = List of questions with a negative answer.

**Feedback prompt** = ""You are a quality control specialist in the finance sector. You are stating that this analysis: "+***HYPOTHESIS***+" is "+**POSITIVE** +", but not "+**NEGATIVE**+". Could you explain why?".

Note: If POSITIVE is empty, the prompt does not use it. Same with NEGATIVE.

# Factor Step Two

## Objectives

At step 2, the objective of the AI integration is to support the user with a selection of **Assessment Focus**, and a general feedback explaining those choices.

## Challenges

The challenges are the same as for step 1, meaning several answers are expected, but at a larger scale and dynamically.

First challenge identical to step 1. We need a Boolean answer for each of the Assessment focus because there are only two choices per focus: Propose it or not. That means building several prompts from a template.

Second challenge. The list of **Assessment Focus** is not fixed butcustomized per team. It means that the code **doesn’t know in advance how many focuses to work on, and the type of focuses**. This is much more complex than for step 1.

Third challenge. Because the number of focuses is unknown in advance, a different algorithm is needed to build a cascade prompt for the feedback.

## Solutions selected

At the Aug 26th 2025 workshop, it was decided to implement only one call to the API to receive a text, not a boolean per focus points in the S2 list.

The inputs are:

* Hypothesis from S1
* List of currently selected focus points
* List of unselected focus points

## Prompt

This is the prompt using all inputs:

**Feedback prompt** = "You are a quality control specialist. Your investment hypothesis is "+***HYPOTHESIS***+". Your objective is to select focus points related to your hypothesis. You have already selected these focus points: “+**SELECTED\_LIST**+”. Do you think that one of these focus points: “+**UNSELECTED\_LIST**+”, or others, might be pertinent for your hypothesis?

# Factor Step Four

## Objectives

The first objective is to validate the user's analysis for accuracy. The second is to provide alternative analysis if possible.

## Challenges

In order for the AI to process the user’s analysis, we have to provide the attached files. That is a challenge for three reasons:

* A file could be very large, and because OpenAI is invoicing based on the number of tokens processed, we could end up with a large bill for a small benefit.
* Actually, the files are not passing through the server. The server is asking AWS a dedicated and temporary URL to send the file to AWS S3. Same thing to download the file. The server is asking for a download URL. If the file’s content has to be pushed to the OpenAI API through the server, it will create a major increase in memory demand that will be costly over time.
* It’s not obvious to create a prompt combining small texts from the analysis conclusions, and large ones from the files, without putting too much weight on the files' content…

## Solutions

In the current version 5.34, the AI integration at step 4 has been put on hold. More tests and studies are needed.  
This decision was restated in the Aug 2nd, 2025 workshop.

# Factor Step Seven

## Objectives

The objective is for the LLM to suggest suitable KPI(s) to track. What « suitable » means is not precisely defined at this stage.

## Challenges

We need to be sure that the KPI suggested by the AI is available to the user.   
The second challenge is to build a comprehensive prompt using as much information as possible.  
The third challenge is to extract individual KPIs from an unstructured list of KPIs merged in a text block coming from the API.

## Solutions

Entering the entire list of available KPIs in the prompt is impossible.

In the Aug 26th, 2025 workshop, it was decided to feed the AI with the list of conclusions selected in step 5, and ask for a general list of potential KPIs, independently of the KPIs available to GPS.

## Prompt

**Feedback prompt** = "You are a quality control specialist. Your investment hypothesis is "+***HYPOTHESIS***+", and your conclusions are “+**S5\_CONCLUSIONS**+”. List pertinent KPIs to monitor both your investment hypothesis and your conclusions.”

# Report Card

## Objectives

The objective of AI integration is to assess the overall quality of the Report Card.

## Challenges

The Report Card does not exist at the server side. It has to be rebuilt using all the factor’s data in DB that are spread in several tables: Analysis, Conclusions, KPIs…

## Solutions

Duplicate the client-side code on the server to build a text-based report card for the prompt.