11th INTERNATIONAL SYSTEMS & CONCURRENT ENGINEERING FOR SPACE APPLICATIONS CONFERENCE (SECESA 2024)

25 - 27 SEPTEMBER 2024

ENHANCING CONCURRENT ENGINEERING FOR SPACE MISSION DESIGN WITH TASK PRIORITIZATION AND LARGE LANGUAGE MODELS

Maria Consiglia Salvemini[1, 2], Federico Volponi[1, 2], Silvia Bucci[1], Carlo Cena[1, 2], Francesco Ferrari[1], Carolina Molteni[1], Alessandro Balossino[1]

[1]Argotec
Via Luigi Burgo, 8 – 10099, San Mauro Torinese (TO), Italy
Reference E-mail: carolina.molteni@argotecgroup.com

[2]Politecnico di Torino
Corso Duca degli Abruzzi, 24 – 10129, Torino (TO), Italy
Reference E-mail: carlo.cena@polito.it

Abstract

Space mission design is a challenging problem for several reasons: large and highly non-linear trade spaces, complex and interconnected analyses, and the need to deal with all mission phases (e.g., procurement, integration, qualification, operations, end-of-life) with technical, programmatic, and stakeholders' needs considerations. As opposed to a traditional sequential approach, which can cause late redesigns, missed opportunities, or suboptimal designs, Concurrent Engineering (CE) fosters teamwork and real-time design sessions to simultaneously advance all aspects of a space mission concept. CE facilitates clients' needs satisfaction through effective trade space exploration, decision-making, and concurrent consideration of all the relevant aspects. This results in valuable, feasible, and consistent designs with a reduced total effort.
Argotec is currently implementing a CE framework to improve the efficiency and effectiveness of its processes for mission formulation and design. The implemented methodologies encompass various aspects related to interactions with the client, the organization and planning of sessions, and a retrospective of the study for continuous improvement. During technical iterations, the trade space exploration and point design are supported by the adoption of an Agile approach using story points and task prioritization, ensuring optimal resource allocation across different phases of design. Additionally, given the growing advancement of Natural Language Processing (NLP) techniques across various application domains, this paper also explores the integration of Large Language Models (LLMs) within CE environments. By integrating LLMs, the aim is to optimize systems engineering processes, particularly in information retrieval, a crucial task considering the substantial volume of internal and external documentation involved in the design process. This paper describes the challenges of implementing CE in an industry setting, the methodologies employed to address them, and innovative ideas that Argotec is integrating into its CE framework. These include an Agile approach to CE sessions, with the objectives of task prioritization and activities planning standardization, and the integration of LLMs in the CE process, with the objectives of supporting sessions' design activities, in particular by leveraging Retrieval-Augmented Generation methodologies to streamline information retrieval processes.

INTRODUCTION

Early-phase space mission design is a problem characterized by a large trade space and difficult estimation of concepts' value: satisfaction of mission objectives must be balanced with technical feasibility, cost, schedule, and risks. All these aspects must be assessed to reach complete and converged designs and compare different mission concepts. However, programmatic aspects may be difficult to assess and have large uncertainties given the limited details in early design phases, while technical aspects typically require performing interconnected analyses, with changes on a subsystem potentially impacting the whole design. Still, it is essential to produce feasible, valuable, robust, and consistent designs in the initial phases, and to explore the trade space as widely as possible. This allows anticipating some design effort to minimize late redesigns, for example due to missed opportunities, suboptimal designs, or unconsidered factors and issues. As a result, it is important to explore a wide range of options and concepts and to reach a sufficient level of detail in each concept design, and this must be done with the limited resources available for early phase designs. The "New Space" paradigm and the context of commercial space companies further exacerbate this problem, with a strong need to produce

valuable mission designs in a very short time. Concurrent Engineering (CE) can be used to perform mission design efficiently and effectively, facilitating the solution of the above-mentioned problems through *real-time* collaborative design sessions involving multi-disciplinary teams, in contrast with a traditional sequential approach that can cause delays, inconsistencies, or missed opportunities. *Argotec* is implementing a CE framework, called the Advanced Concepts Laboratory (*ACLab*), to perform mission design with this methodology and is implementing novel approaches to further improve the efficiency of the design process.

First, task prioritization is used as a formal approach to estimate the time required for various tasks and plan the CE sessions, ensuring that important and/or critical aspects can be tackled with the required effort, while simpler tasks are completed in a limited time. This has been tested in a pilot study, and found to be effective to identify critical tasks.

Secondly, a Retrieval-augmented generation (RAG) pipeline, which relies on new State-Of-The-Art models in the Natural Language Processing (NLP) field, is implemented to support Information Retrieval (IR), an essential and strongly time-consuming task during trade-space exploration and early phase design, where design by analogy is often used in place of analyses that are too complex or detailed for the concept maturity level.

The paper is organized as follows: after a review of the related works of both task prioritization and Artificial Intelligence (AI) in CE, the methodologies used to implement task prioritization in a CE pilot study and to develop and test the RAG pipeline are presented. Subsequently, the respective results are reported together with a critical discussion. Finally, conclusions with potential future works to implement agile-based methodologies in CE studies are reported.

BACKGROUND & RELATED WORKS

Task Prioritization and Agile in Concurrent Engineering

Traditional Concurrent Design methodologies are widely used for space mission design to include all relevant disciplines in real-time design sessions. However, the traditional concurrent design methodologies currently used by the top players in the field of Space CE (e.g., ESA's Concurrent Design Facility, JPL's A-team and Team X) don't envisage a formal and systematic estimation of the effort of the technical and programmatic activities involved in the mission design: the effort is distributed simultaneously across all participants and no prioritization of critical activities is foreseen. To the authors' knowledge, one work was published on the implementation of Agile methodologies in space mission design CE studies, by the Polytechnic University of Madrid [1]. Agile methodologies are widely used in project management to allow a project to go faster in environments where requirements are likely to change and vary at a quick pace, requiring a high degree of adaptability of the project during the development. Scrum/Agile frameworks include the structure of work in "sprints", sprint planning activities, the creation of a product backlog with prioritization and dependencies, and sprint review activities. Phases 0 and A conducted in an industrial framework are a good example of such projects: usually engineers are required to perform fast-paced mission studies to submit proposals within a strict deadline. Since CE was born in the nineties with the similar purpose to enhance mission design outputs in early phases and facilitate the work of following phases, the cited work explores the introduction of a scrum approach during CE sessions to understand the compatibility of these two frameworks. The work focused on prioritization of mission requirements and implementation and consequent re-allocation of resources during design phases, granting a greater number of design experts to be assigned to the disciplines that have priority in the design (e.g., mission analysis, payload, ConOps). The result observed through the application of this methodology was an increase in the optimization of the spacecraft under design (in terms of mass, power, ΔV, …) for a team using the scrum methodology with respect to a team using classical concurrent design approach. For other examples of Agile methodologies in CE, not specifically related to space mission design, see [2, 3].

Artificial Intelligence for Mission Design

The integration of AI in the design and analysis of space missions is a growing area of research, particularly focusing on improving the efficiency and capabilities of CE. Several projects have explored the use of AI-powered virtual assistants to aid engineers during spacecraft mission design. Notable examples are Daphne [4, 5] and SpaceQA [6]. Daphne [4] is a virtual assistant for designing Earth observation distributed spacecraft missions. Its comprehensive question-answering system and cognitive assistance features were assessed through a study at NASA's JPL involving nine people. The findings suggest that Daphne can improve performance during system design tasks compared to traditional tools. Reference [5] delves into the application of Knowledge Representation and Reasoning and Expert Systems as Design Engineering Assistants. It emphasizes the utility of converting unstructured, legacy data into structured data stored in Knowledge Graphs to enhance design processes in CE sessions. This study also explores the use of Ontology Learning methods to automate the knowledge base generation, addressing the challenges of manual data curation. SpaceQA is the first open-domain question-answering system specifically designed for space mission design. Developed under ESA initiative, SpaceQA utilizes an architecture combining dense retrieval and neural reading, with an emphasis on transfer

learning due to the scarcity of domain-specific annotated data. Preliminary evaluations indicate the effectiveness of this approach, though further fine-tuning is necessary for optimal reading comprehension. Collectively, these works underscore the transformative potential of AI-driven tools in streamlining and augmenting the complex processes involved in space mission design, paving the way for more efficient and informed decision-making within concurrent design facilities.

Language Models

The RAG is an architecture incorporating a dense retrieval and a generator module. This term was introduced by [7], where the proposed approach combines pre-trained parametric and non-parametric memory for language generation. This method addresses the limitations of the parametric memory of large language models (LLMs), acquired during pre-training, which often lack up-to-date information and domain-specific knowledge. By integrating an external knowledge source accessed by a retriever model, RAG mitigates these limitations. Although the pipeline can be fine-tuned end-to-end, [8] demonstrates that domain adaptation of the retriever alone leads to improved results. Reference [9] further shows that effective supervised training can be performed on synthetically generated questions along with the given context, addressing the issue of training data scarcity.

METHODOLOGY

Argotec's High Level Concurrent Engineering Methodology

The developed CE methodology is tailored to *Argotec's* corporate environment, which requires adaptation of classical methodologies: for example, flexibility on the number of domains covered by a single person is added to accommodate the fact that subject matter experts are sometimes busy on flight missions or that assigned resources are limited. For this reason, the developed CE methodology allows a single individual to cover for multiple roles and/or domains if necessary. Also, it was observed that the constraints of a study in an industrial framework are not always well specified and formulated, which leads to misunderstandings. *ACLab's* mission formulation methodology is based on Study Progress Levels (SPLs), inspired by JPL's Concept Maturity Levels [10]. Each SPL, described in Table 1, represents the achievement of specific goals and a point in the design process where results are consolidated.

Table 1: *ACLab* SPLs description

SPL	Description	Goal
1	• Consolidation of the study proposal with the client • Definition of study objectives and constraints	Go for starting the study
2	• Study preparation: tasks quotation, team assembly, sessions' planning • Study presentation • Mission objectives flow-down	• Study planned (sessions scheduled, required resources allocated) • Mission objectives defined
3	• FOM (Figures Of Merit) selection • Trade space exploration – concept push • Trade-off analysis – concept pull	• Trade space explored • Mission concepts ranked/selected
4	• Technical and programmatic analyses	• Point design • System budgets consolidation

Before starting the study, the *ACLab* workflow requires iterations with the client to consolidate or define the study's objectives and constraints. This step is crucial for a better understanding of the client's needs and to avoid studies with unclear objectives. This step marks the achievement of the SPL 1. Once the study is approved, a Study Lead (SL) is assigned. The SL is responsible for understanding the client's requests, preparing the study, and guiding discussions during the sessions. During SPL 2, the SL identifies the required areas of expertise, assembles the team, and plans the sessions. A task prioritization approach, which will be described in detail later, is used to help the SL plan the sessions and allocate resources effectively.
After the study preparation, the design sessions can begin. SPL 2 is concluded with the mission objectives flow-down, while SPL 3 corresponds to trade space exploration, in particular concepts generation and ranking/selection. Finally, SPL 4 concerns the point design of the selected concept(s). The RAG pipeline is employed for IR, supporting the state-of-the-art study and preliminary feasibility assessment. The retrieved data allows for initial analyses based on analogies, for example with similar past missions whose details are stored in the RAG Knowledge Base (KB). This method allows for quick estimations and supports concepts and ideas generation; these tasks can be time-consuming, especially for unplanned research that must be done during CE sessions. Thus, the RAG tool will facilitate these researches and enable

saving study resources. At the end of the study, the final report is given in input to the KB, keeping the model's knowledge up to date for future session.
The workflow described above represents the current baseline of the *ACLab* CE methodology, represented in Fig. 1.

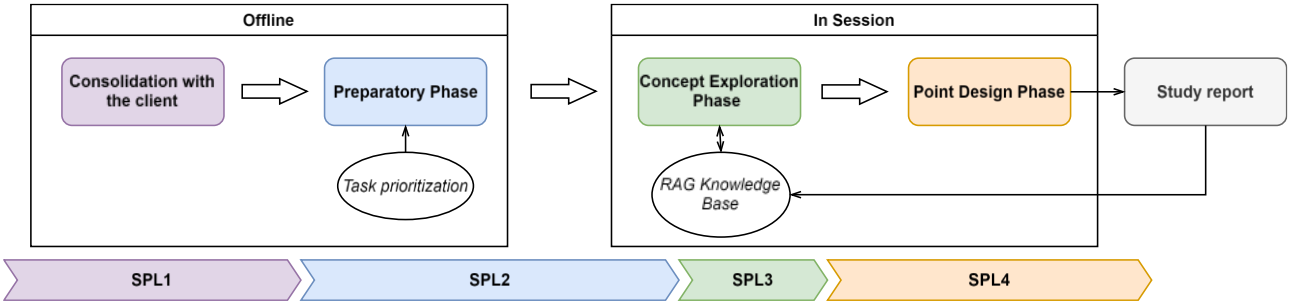

Fig. 1. *ACLab* high level methodology

Task Prioritization in Argotec's *ACLab*

In *ACLab's* CE methodology, task prioritization is employed in the study preparation to support resources allocation according to the criticality of the tasks. The SL, with the support of the Subject Matter Experts (SMEs), begins the task prioritization by compiling a backlog of tasks based on the study goals; during a preparatory meeting, the team reviews the backlog and assigns story points to tasks considering effort, complexity, and associated risks as evaluation parameters. If the team disagrees on story points allocation, a brief discussion ensues, and the voting is repeated. This process repeats until an agreement is reached. Moreover, during this activity, tasks are divided between those to be carried out in session and "offline". The SL then converts the story points into session hours, starting from the expected time required to complete the task with the least effort and then scaling this time proportionally for the other tasks.

RAG Pipeline and Embedding Models Fine-Tuning

The purpose of the RAG pipeline is to facilitate the retrieval of information from documents, for example past missions' descriptions, to support the early-phase design with estimations by analogy or the trade-space exploration with ideas generation. Fig. 2 shows a high-level overview of the pipeline.

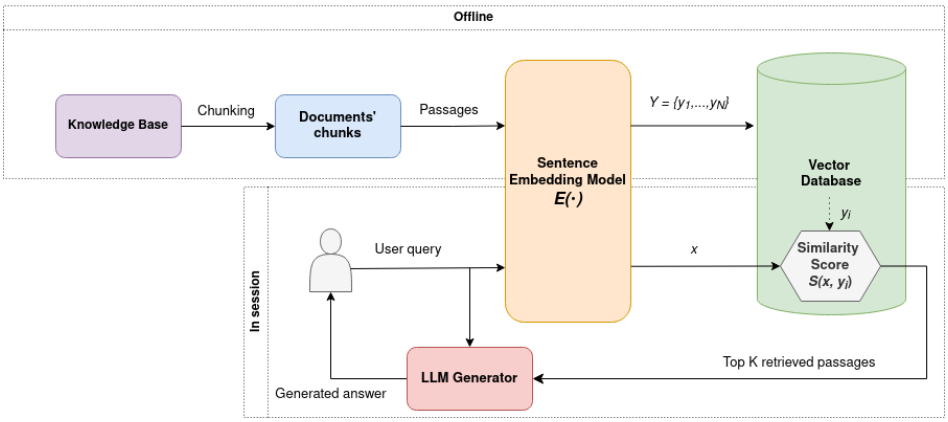

Fig. 2. Retrieval-augmented generation pipeline

To construct the non-parametric memory of the pipeline, a KB of scientific publications is defined. Each document is further split into smaller chunks of text, then encoded into semantically meaningful embeddings using a sentence embedding model [11] and stored into a vector database (Y). During the retrieval process, the user's query is encoded and compared with the vector representations of all documents' chunks to identify the top k most semantically similar passages using the cosine similarity. The retrieved passages are then given in input to the generator, together with the user query. In view of incorporating *Argotec's* private documents into the KB at a more advanced stage, state-of-the-art open-source models, such as Mistral [12], has been leveraged as generator; the use of closed-source model APIs has been excluded for Intellectual Property (IP) rights reasons. z

According to [13], various methods can enhance RAG performance. Given the specific semantics and terminology of the aerospace domain, and the fact that general-purpose embedding models are trained on non-specific corpora, the retrieval stage can be improved fine-tuning the embedding model on aerospace-related data. To train the embedding model, we employed the Multiple Negative Ranking Loss [14]. This loss function $L(x, y)$ takes as input a batch consisting of B sentence pairs $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where (x_i, y_i) are positive pairs and (x_i, y_j) with $i \neq j$ are negative pairs. Here, x represents the question and y the reference passage. The loss for a single batch is defined as in (1), where $S(x_i, y_i)$ denotes the cosine similarity score, which is computed by measuring the cosine of the angle between the vector representations of x_i and y_i.

$$L(x,\ y)\ =\ -\frac{1}{B}\sum_{i=1}^{B}\left[S(x_i,\ y_i)\ -\ \log\sum_{j=1}^{B}e^{S(x_i, y_j)}\right] \tag{1}$$

CASE-STUDY VALIDATION: RESULTS AND DISCUSSION

The first integration of task prioritization and the RAG tool in the *ACLab* context was tested during a pilot study aimed at validating the developed methodologies and tools. The study consisted of a Phase 0 mission design concerning a deep space CubeSat propelled by a solar sail. The team consisted of one Study Lead and two systems engineers. The corporate environment does not always allow the possibility to have a SME available for each area. Hence, one of the systems engineers covered the areas of Propulsion/Sail, GNC and Cost, the other covered the areas of OBC, TT&C and Mission Analysis, while the SL, who also took on the role of LSE (Lead System Engineer), supported the EPS design.

Validation of Task Prioritization in Concurrent Engineering Sessions

During the task quotation activity, performed as described above, an agreement on the effort allocated to each task was reached in most cases within one iteration, and in all cases within 3 iterations. The result of this process was the scheduling of 8 sessions, each lasting 4 hours, to be held weekly to accommodate offline work between sessions. During the study, the hours of both in-session and offline work were tracked to verify the effectiveness of the task's time allocations.

The first trial of integrating task prioritization into a CE framework highlighted certain issues in the initially envisaged method. Firstly, feedback collected at the end of the quotation session revealed that the process took a large amount of time, totalling 9 man-hours, almost equivalent to an entire session. This clashes with CE's principles of time and costs reduction. To address this issue, possible solutions are leaving the task quotation process and the consequent session planning solely to the SL, with on-call support of experts when needed, or imposing timing constraint on tasks quotation.

The time allocation for the tasks conducted during SPL 2 and 3 worked successfully. It was observed that the initial session planning logic could work since the entire team works on the same tasks regardless of the assigned roles, as the sessions are mostly brainstorming and discussions involving the whole team, focusing on the expansion of concepts and ideas. After the first two sessions, it was found that tasks shifted to different sessions or that effort needed to be re-allocated due to discrepancies between planned and actual work (for a maximum of a 3-hours discrepancy for one task). Despite these discrepancies, the task prioritization proved useful to identify critical tasks, required SMEs, and to give a rough estimate of tasks' required effort; performing more studies will likely lead to improved quotations.

On the contrary, the process of converting story points into session hours was found inefficient for tasks located in SPL 4, as this approach diminishes the ability to perform activities concurrently: creating a session schedule based on the hours allocated to tasks almost inevitably leads to planning the session design process sequentially or otherwise inefficiently. Thus, tasks prioritization does not seem suitable to support corporate CE point designs. This difficulty is especially due to the limitation of having only three participants who are responsible for the design of more than one subsystem. As a result, tasks would overlap, making management difficult and not helpful for the session planning process; on the other hand, a "complete" team would likely result in some team members being inactive during sessions.

Although the initial task quotation turned out to be inefficient for concurrent work, it enabled the identification of the most critical areas, hence an alternative solution for planning the sessions based on the estimated effort was envisaged. This approach (especially suitable after SPL 3) involves identifying goals to be achieved by the end of each session, rather than tasks to be completed. To achieve these goals, during the concurrent design of the identified high-effort areas (e.g., solar sail design or ADCS), the support from additional SMEs was required for a time proportional to the quotation. This

involvement was crucial for advancing the design, covered the initial issue of not having a SME for each area, and confirmed that the areas identified as critical prior to the study were indeed the ones requiring more effort.

Results of the Fine-Tuned Embedding Models

The lack of space-related labeled data for supervised fine-tuning and performance evaluation needs the creation of a custom dataset. Based on [9], which demonstrates how text embeddings fine-tuned on synthetic data perform competitively with those fine-tuned on human-labeled data, we generated Question and Answer (QA) pairs from the document chunks chosen for the KB using Llama 3 (8B parameters) [15]. To assess the quality of the generated pairs, we used the LLM-as-a-judge open-source model Prometheus 2 [16], which scored the pairs on a scale from 1 to 3 based on their alignment with the input source information and the coherence of the questions. After filtering out samples that scored less than two, the resulting dataset comprised 8348 pairs. This dataset was then divided into training, validation, and test sets, containing 7232, 804, and 312 samples, respectively. To ensure more meaningful results, the test set pairs were further human-validated.
The KB is composed of 906 scientific papers on the space domain, split in chunks of a maximum dimension of 512 tokens, for a total of 8348 chunks. The chosen embedding models are from the BGE [17] and GTE [18] families, specifically the large (approximately 400M parameters) and base (approximately 150M parameters) versions. These models have been selected due to their high performance on retrieval tasks, as demonstrated on the MTEB [19] leaderboard, and their relatively small size compared to the top-ranking models with 7B parameters. This smaller size is crucial for running them locally with limited computational resources. To evaluate the models' capabilities, we selected the following IR metrics: Accuracy@k, which measures if at least one relevant document is present in the top k retrieved documents; Normalized Discounted Cumulative Gain (NDCG@k); and Mean Reciprocal Rank (MRR@k), both of which are ranking quality metrics. Since the pipeline runs locally, we also explore the possibility of reducing the memory footprint of the vector database. Thus, we evaluate the embedding models following the Matryoshka Representation Learning principle [20], truncating the embeddings to smaller sizes compared to the original. The comparison between the baseline and fine-tuned models is presented in Table 2. We evaluated the accuracy at $k = 5$ because it is the number of passages that are given in input to the generator, while NDCG and MRR at $k = 10$ to have a general overview of the retriever performance.

Table 2: IR metrics for the embedding models before and after fine-tuning. Where the embedding dimension is not specified, the default one of the models is intended.

Embedding Models		**Accuracy@5**			**NDCG@10**			**MRR@10**		
		Embedding Dimension			Embedding Dimension			Embedding Dimension		
	Baseline (B) / Fine-tuned (F)	-	512	256	-	512	256	-	512	256
gte-base-en-v1.5	B	0.84	0.85	0.83	0.73	0.73	0.71	0.68	0.68	0.65
	F	**0.92**	**0.90**	**0.89**	**0.80**	**0.77**	**0.76**	**0.74**	**0.7**	**0.70**
gte-large-en-v1.5	B	0.85	0.84	0.80	0.73	0.72	0.69	0.66	0.66	0.63
	F	**0.88**	**0.87**	**0.85**	**0.76**	**0.76**	**0.75**	**0.71**	**0.71**	**0.69**
bge-base-en-v1.5	B	0.76	0.76	0.70	0.67	0.66	0.61	0.61	0.60	0.55
	F	**0.83**	**0.83**	**0.82**	**0.74**	**0.73**	**0.72**	**0.68**	**0.68**	**0.66**
bge-large-en-v1.5	B	0.85	0.82	0.79	0.74	0.72	0.68	0.69	0.66	0.62
	F	**0.88**	**0.86**	**0.83**	**0.76**	**0.76**	**0.74**	**0.71**	**0.71**	**0.69**

Fine-tuning improves all metrics across all models and embedding dimensions tested. The smaller models, *bge-base* and *gte-base*, exhibit more significant improvements (up to 8 points) compared to the larger models. Notably, the *gte-base* model outperforms all others, despite the larger models starting with higher baseline metrics. While truncating the size of the embeddings slightly reduces model quality, it provides an excellent trade-off between performance and storage requirements.

To understand the impact of retriever performance on the generation stage, the quality of the answers is evaluated by combining traditional evaluation metrics with LLM-as-judge evaluation. The traditional evaluation metrics used are BLEU [21], ROUGE [22], BERTScore [23], SemScore [24], and METEOR [25]. The LLM-as-a-judge model (Prometheus 2) assesses the faithfulness of the response to the relevant documents (this metric ranges from 0 to 5). As generator we tested the Mistral version 0.3 (7B parameters) model with $k = 5$ passeges given in input. Table 3 presents the results for the best model before and after fine-tuning.

Table 3: Text generation metrics for Mistral 7B v0.3 with the best embedding model.

Embedding Model	Faithfulness	BLEU	ROUGE-L	BERTScore (F1)	SemScore	METEOR
gte-base-en-v1.5 (baseline)	3.38	9.8	29.7	65.9	63.2	48.8
gte-base-en-v1.5 (fine-tuned)	**3.39**	**11.3**	**31.4**	**66.6**	**64.9**	**51.3**

All the metrics show an improvement meaning that, as expected, the retriever performance affects the generator. However, the evaluation of text generated by LLMs is still an open research field.

To assess the effectiveness of the RAG pipeline, it has been tested with questions that have been already answered during the above-mentioned pilot study. During this process, 16 questions have been given in input to the model. The three participants in the pilot study assessed the quality of the responses, providing scores between 1 and 5 based on four criteria: coverage, consistency, correctness, and clarity [26]. Table 4 reports the average scores.

Table 4: Human evaluation on four criteria of the responses generated by the pipeline.

Coverage	Consistency	Correctness	Clarity
4.21	3.98	4.29	4.44

The results demonstrate that the pipeline effectively satisfies users' requests by providing complete and correct answers. Outputs with low scores were primarily due to missed retrieval of the correct passages; however, this issue can be mitigated by rephrasing the question and adding more details.

CONCLUSION

Based on the experience gained from the case study, *Argotec* intends to keep testing the implementation of Agile-based methodologies and AI-based tools into its CE framework. Task prioritization, performed by the SL during the study preparation, will aim at identifying activities requiring higher effort, and will be crucial to fine-tune the goals to be achieved at the end of each session. Moreover, it will help in the planning of sessions before starting the point design and in the identification of SMEs to involve during the study.
Other agile-based methodologies are planned to be implemented in CE studies. Critical issues of corporate environments are the involvement of required resources, and the (often) fast-changing client needs/goals. The client, internal team members, and external partners may not be involved in sessions (for example due to their availability, or due to IP rights concerns in case of external partners). By organizing brief reviews of the performed work at key points during the design, it is possible to consider feedback or changes of needs/constraints/goals from the client, or to integrate inputs from SMEs or external partners unable to participate in sessions. These reviews will be tested in future *ACLab's* CE studies, with the goals of increasing the flexibility of the methodology and mitigating issues of the corporate environment.
Finally, the RAG pipeline has been tested with real questions that were asked during sessions and required some literature research, to check its reliability and time-saving possibilities.
The main limitation of this study is the limited resources available for the study, as only 3 people attended all sessions. Although a more "complete" study with more people and SMEs may potentially impact the validity of the results, this situation is uncommon in corporate environments.

ACKNOWLEDGEMENTS

This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 117/2023. We also acknowledge ESA Academy for the Conference Student Sponsorship awarded to Federico Volponi.

REFERENCES

[1] J. M. Álvarez and E. Roibás-Millán, "Agile methodologies applied to Integrated Concurrent Engineering for spacecraft design," *Research in Engineering Design,* vol. 32, p. 431–450, July 2021.

[2] H. Sen Yan and J. Jiang, "Agile concurrent engineering," *Integrated Manufacturing Systems,* vol. 10, p. 103–113, April 1999.

[3] T. Žužek, J. Kušar, L. Rihar and T. Berlec, "Agile-Concurrent hybrid: A framework for concurrent product development using Scrum," *Concurrent Engineering,* vol. 28, p. 255–264, September 2020.

[4] A. V. i. Martin and D. Selva, “Daphne: A Virtual Assistant for Designing Earth Observation Distributed Spacecraft Missions,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 13, p. 30–48, 2020.

[5] A. Berquand, F. Murdaca, A. Riccardi, T. Soares, S. Genere, N. Brauer and K. Kumar, “Artificial Intelligence for the Early Design Phases of Space Missions,” in *2019 IEEE Aerospace Conference*, 2019.

[6] A. Garcia-Silva, C. Berrio, J. M. Gomez-Perez, J. A. Martínez-Heras, A. Donati and I. Roma, “SpaceQA: Answering Questions about the Design of Space Missions and Space Craft Concepts,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2022.

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel and D. Kiela, *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,* arXiv, 2020.

[8] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana and S. Nanayakkara, *Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering,* arXiv, 2022.

[9] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder and F. Wei, *Improving Text Embeddings with Large Language Models,* arXiv, 2024.

[10] R. Wessen, C. S. Borden, J. K. Ziemer, R. C. Moeller, J. Ervin and J. Lang, “Space Mission Concept Development using Concept Maturity Levels,” in *AIAA SPACE 2013 Conference and Exposition*, 2013.

[11] N. Reimers and I. Gurevych, *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,* arXiv, 2019.

[12] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix and W. E. Sayed, *Mistral 7B,* arXiv, 2023.

[13] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang and H. Wang, *Retrieval-Augmented Generation for Large Language Models: A Survey,* arXiv, 2023.

[14] M. Henderson, R. Al-Rfou, B. Strope, Y.-h. Sung, L. Lukacs, R. Guo, S. Kumar, B. Miklos and R. Kurzweil, *Efficient Natural Language Response Suggestion for Smart Reply,* arXiv, 2017.

[15] Meta, “Llama 3,” [Online]. Available: https://llama.meta.com/llama3/.

[16] S. Kim, J. Suk, S. Longpre, B. Y. Lin, J. Shin, S. Welleck, G. Neubig, M. Lee, K. Lee and M. Seo, *Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models,* arXiv, 2024.

[17] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian and J.-Y. Nie, *C-Pack: Packaged Resources To Advance General Chinese Embedding,* arXiv, 2023.

[18] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie and M. Zhang, *Towards General Text Embeddings with Multi-stage Contrastive Learning,* arXiv, 2023.

[19] N. Muennighoff, N. Tazi, L. Magne and N. Reimers, *MTEB: Massive Text Embedding Benchmark,* arXiv, 2022.

[20] A. Kusupati, G. Bhatt, A. Rege, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain and A. Farhadi, *Matryoshka Representation Learning,* arXiv, 2022.

[21] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL ’02*, 2001.

[22] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Text Summarization Branches Out*, Barcelona, 2004.

[23] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger and Y. Artzi, *BERTScore: Evaluating Text Generation with BERT,* arXiv, 2019.

[24] A. Aynetdinov and A. Akbik, *SemScore: Automated Evaluation of Instruction-Tuned LLMs based on Semantic Textual Similarity,* arXiv, 2024.

[25] S. Banerjee and A. Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,” in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann, 2005.

[26] L. Gienapp, H. Scells, N. Deckers, J. Bevendorff, S. Wang, J. Kiesel, S. Syed, M. Fröbe, G. Zuccon, B. Stein, M. Hagen and M. Potthast, “Evaluating Generative Ad Hoc Information Retrieval,” 2023.