REQUIREMENTS ENGINEERING PROCESS FOR SERANIS, A SMALL SATELLITE MISSION

E. Gadzo[1*], F. Porcelli[1*], R. Förstner[1], K. Paetzold[2]
[1]Universität der Bundeswehr München, Germany
[2]Technische Universität Dresden, Germany
∗ Corresponding Authors

Abstract

Traditionally, space missions were exclusive for big governmental institutions. Nowadays, the technological and budget resources for carrying on a space mission are becoming more widely available, giving the possibility to universities, start-ups, and private companies to reach the space. The new players usually develop scaled down space missions, given the constraint they have, especially on budget and human resources. One of the reasons for this is that, in view of the needs of users and operators of small satellite mission systems, the system of the individual satellite is no longer the only decisive development objective. Rather, in future projects, the focus of the development will be on the services and performance of the emerging space network, consisting of different systems that interact and cooperate in a multidisciplinary manner. The systems form a network that can meet the needs and desires of the users while considering the constraints of the operators. To ensure the successful development of such a system of systems, it is necessary to use efficient, goal-oriented and, above all, systematic approaches in the product development process when developing the individual systems.

This paper presents a new systematic approach for the development, management, and control of the requirements of a small satellite mission, as well as for increasing the information maturity of the requirements. It further highlights the differences between the stricter and time-consuming processes, required for large missions, and the flexibility achieved in the approach presented in this paper. Both, the reference of ECSS standards and the project constraints and limiting factors of a small satellite mission in a stage gate driven process such as a tight schedule, budget and human resources have been taken into consideration. This way, it has been possible to develop an effective process for the development and engineering of the requirements, given the complexity of a mission carrying more than 20 experimental payloads.

Keywords: requirements engineering, small satellite mission, process development, new space

doi: 10.25967/570212

ABBREVIATION LIST

AI	Action Item
AIT	Assembly, Integration, and Test
AR	Acceptance Review
CCB	Change Control Board
CDR	Critical Design Review
Conops	Concept of Operations
COTS	Commercial Off The Shelf
DSD	Domain Specific Developer
ECSS	European Cooperation for Space Standardization
EID	Experiment Interface Document
EMT	Electrical, Mechanical, Thermal
FBC	Fast, Better, Cheaper
IEEE	Institute of Electrical and Electronics Engineers
LU	Laboratory Unit
OBC	On Board Computer
PA	Product Assurance
PM	Project Management
RCCP	Requirements Change Control Phase
PDR	Preliminary Design Review
RID	Review Item Discrepancy
SE	System Engineering
SeRANIS	Seamless Radio Access Network for Internet of Space
SRR	System Requirements Review

1. INTRODUCTION

The rapidly growing number of connected users as well as the constantly increasing demands on quality and availability of communication networks pose great challenges for current network technologies [1].This leads to a shift in the focus of the development goal to the services and performance of the emerging network, which consists of various complex systems that work together interactively and multidisciplinary. They form a network that can meet the needs and desires of users while accommodating the interests of operators. Small satellite missions can be well deployed in such a network of networks and are suitable to fill existing gaps. On the one hand, since small satellite missions require lower technological, financial, and human resources, new players can enter the scene and reach the space with the development of such downsized space missions, e.g., universities, startups, and private companies. On the other hand, in such networks, increasing customer focus for specific requirements leads to increasing complexity of products and systems [2]. In addition, product development must strike a balance between development resources, time, and costs from the operator's point of view [3]. Overall, the development process itself is often very complex, depending on the complexity of the product to which it relates [4]. To ensure the successful development of such a system of systems, it is necessary to apply efficient and systematic procedures and processes to the development of each system during product creation. The needs and requirements of the various stakeholders must be considered [5].

In order to systematically address these challenges in space projects and to support organizations in overcoming them, the European Cooperation for Space Standardization (ECSS) has defined a set of standards. The ECSS is a cooperation between the European Space Agency, European industry associations and national space agencies with the aim of developing and maintaining common standards [6]. However, the process requirements defined in the standards refers only to what is to be achieved and no reference is made to the organization and execution of the required work, which allows the use of existing organizational structures and methods, thus ensuring the further development of these structures and methods without having to rewrite the standards [ibid]. Hence, boundary conditions from the project and the organization can be considered in the process design.

This leads to the following research questions:

RQ1: In which manner the requirement engineering process must be designed and tailored to meet the requirements as defined in the ECSS standards on the one hand and to consider the project constraints of a small satellite project on the other?

RQ2: Is it possible to develop and embed an alternative requirement engineering process that addresses the needs and constraints of a small satellite mission in a more specifically and efficiently manner while being implemented in a dynamic and interdisciplinary environment?

The processes presented in this paper focus on the development phase between (and including) the SRR and PDR and the respective boundary conditions and was applied in the SeRANIS (Seamless Radio Access Network for Internet of Space) project. The SeRANIS project is briefly introduced in chapter 2.2.

For this purpose, the paper is structured as follows: In Section 2, the relevant theoretical background is explained, and the state of the art is briefly outlined. Section 3 describes the approach in general and explains the applicated processes. The fourth section contains a conclusion and finally in section 5 a short summary and an outlook on further research activities is given.

2. BACKGROUND

2.1. Foundations & Terms

The following section briefly outlines the terms *method*, *methodology*, *process*, *requirement*, and *requirements engineering*. The term *process* describes the execution of a logical sequence of tasks required to achieve a specific goal [7]. The *process* defines "what" is to be done without specifying "how" a corresponding task is to be performed [ibid]. In contrast, the term *method* is used to describe "how" a specific task is to be performed. A *method* also describes which techniques are used to execute the task. Each method is also itself a process with a sequence of tasks to be performed [ibid]. In other words, the "how" at one level of abstraction becomes the "what" at the corresponding lower level.

A methodology is a combination of different individual methods [8] and can be seen as a kind of "recipe" with which a class of problems is tackled using related processes, methods, and tools [7]. For the development of data, information, and requirements in SRR and PDR, two different methodical procedures are described and compared, whereby different techniques and tools are used respectively, and a super-ordinate process exists. The paper focuses on which tasks and process steps have to be performed in order to answer the research questions. Thus, the emphasis is on "what has to be done". Furthermore, processes are described and compared in the paper without picking up a clearly defined combination of methods. Based on this, the term process is preferred to the terms method and methodology to describe the approach from an ontological point of view.

IEEE defines the term *requirement*, specifying the following three aspects [9]:

- a requirement is a property or capability needed by a user to solve a problem or achieve a goal. User can mean both a person and another system.
- a property or capability that a system or sub-system must meet or possess in order to satisfy a contract, standard, specification, or other formally specified document.
- a documented representation of a property or capability according to (1) or (2).

Rupp [10] considers the term *requirement* as a demand for a performance, which the system must provide as a final result of a development. However, this represents only one aspect of the term, since requirements can be considered on different detailing levels and also apart from the system itself, different further addressees for requirements exist, e.g., test cases and planning documents for the considered system. For the purposes of the elaborated approach in this work, the definition of Lindemann [11] is used: A *requirement* is a demanded function and characteristic of a product. The development project converts the defined requirements into product-features and their characteristics [ibid].

The term *requirements engineering* is explained in Lindemann [11] as an approach, which contains superordinate the two processes *task clarification* and *requirements management*, but is not firmly established in the design theory of mechanical engineering. Thereby the collection, analysis and documentation of the requirements fall under the *task clarification*. The release, versioning, change-management, and traceability is considered under the process of *requirements management* [ibid]. Dick [12] considers the concept of *requirements engineering* as a subset of *systems engineering*, which deals with the key activities of discovering, developing, tracking, analyzing, qualifying, communicating, and managing of requirements. These requirements define the system under consideration at various levels of abstraction. In this context, systems engineering is defined as an interdisciplinary approach that enables the implementation of successful systems, focusing, among other things, on the definition of customer needs and the functionality required in the process [13]. The overall problem definition is thereby always considered. Finally, *requirements engineering* should not be limited to an early phase of the product development process but must be carried out in every phase of the development process, since the analysis of requirements plays an important role in every phase [12]. Figure 1 summarizes the key activities and illustrates the definition of abstraction levels.

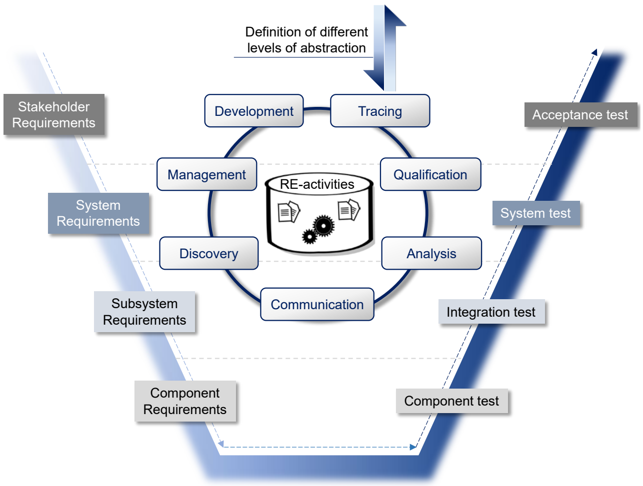
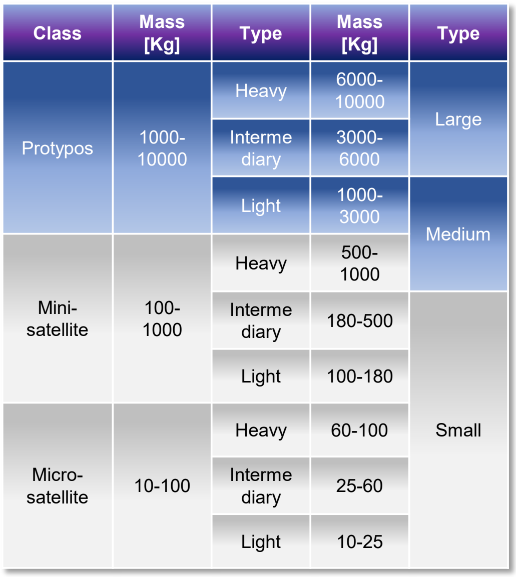


Figure 1: Requirements engineering activities and layers based on Dick [12]

2.2. Small satellite missions & SeRANIS

Traditionally, the access to space was a privilege for those national and international organizations possessing the advanced technical and costly capabilities to develop a space mission. The development of those missions used to stretch over a decade or more, based on a risk-averse development. With this approach, high standards are set for reliability in order to minimize risks, resulting in expensive and complex processes that require huge amount of working force, time, and money. This way of developing a space mission is also often referenced to as *Traditional Space*. This approach has now an alternative. In the last decades, the increasing availability of microelectronics pushed a change in the space sector, enabling the Realization of physically smaller satellite. *Small satellites* gained interest at the end of 90s, beginning of 00s, when they first managed to meet commercial needs. At the beginning of the 00s, the Faster, Better, Cheaper (FBC) approach was developed [14]. This approach consisted of a series of management concepts that revolutionized the implementation of satellite missions [ibid]. Nowadays, the term *small satellite* refers more to a concept rather than a physically small satellite [15]. A unique and generally accepted classification of satellite classes is not available[16]. According to [16], and as reported in Table 1, Small-Satellites can be considered all satellites with a mass below 500kg, belonging to the Nano (1kg-10kg), Micro (10kg-100kg), or Mini (100kg-1000kg) satellite classes. Regardless of pure mass values, the term *small satellite* refers to a series of new management approaches for the development of a satellite mission that make use of commercial-of-the-shelf (COTS) products with the aim of reducing the time and the cost of the mission and increase at the same time its flexibility in projects often being also high-risk compared to the *traditional Space* projects. According to Kochel et al. [17], the term *new Space* started to be used to identify the new spacecraft development approach that emerged thanks to the increasing involvement of private companies and investors in the space industry, and to the increasing availability of microelectronic COTS products. *New Space* is characterized by the usage of new manufacturing method such as additive manufacturing, modularization, and fractionation. This results in reduction of development time and cost and leads to being responsive to changing market requirements and uncertainties due to the capability of reconfiguring rather than redesigning the satellite.



Class	Mass [Kg]	Type	Mass [Kg]	Type
Protypos	1000-10000	Heavy	6000-10000	Large
		Intermediary	3000-6000	
		Light	1000-3000	Medium
Mini-satellite	100-1000	Heavy	500-1000	
		Intermediary	180-500	Small
		Light	100-180	
Micro-satellite	10-100	Heavy	60-100	
		Intermediary	25-60	
		Light	10-25	

Table 1: Satellite Classification according to Xavier [16]

It is in this framework that the Seamless Radio Access Network for Internet of Space (SeRANIS) project develops a *small satellite*[1] *mission* established by the Zentrum für Digitalisierungs- und Technologieforschung der Bundeswehr (dtec.bw). The research environment in which this mission is carried out, is the perfect ground to further investigate not only new satellite and ground technologies but also the new management and development processes[2] that are evolving in the New Space era. In general, *small satellite missions* with an experimental focus facilitate the gathering of new knowledge regarding the use of novel space-based technologies and their behavior in a specific operating environment. These findings are an important basis for a later integration of the investigated and demonstrated technologies in a higher-level communication network. The identification of opportunities and constraints for the corresponding payload deployment, but also for the satellite itself, plays an important role in this process [5]. Figure 2 shows a payload technology scope that can be used for this purpose and a possible scope of satellite technology experiments. The needs of all these users must be met, which implies the coordination and participation of many technical domains [18]. The interdisciplinary character of the development of such an experimental small satellite mission becomes evident, which raises the question of the use of suitable approaches and procedures in the development process.

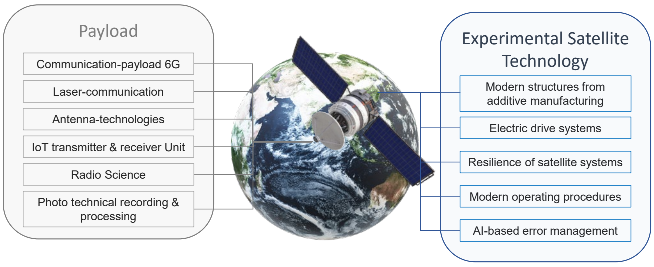


Figure 2: Experimental scope of a small mission satellite according to Gadzo et al. [5]

2.3. ECSS-Standards

As already specified in the introduction, in this paper the requirement engineering approach used in the SeRANIS project focuses on the phase between (and including) SRR and PDR.

This phase is particularly critical with respect to requirements since they drive the preliminary design that eventually will then lead to new requirements or updates of requirements for the payload itself or towards the whole system. This mutual influence and inter-connection of requirements and design, makes this phase extremely interesting and source of studies for requirement engineering approaches.

According to the ECSS Standard, the phase between SRR and PDR, falls in the so-called Phase B, the Preliminary definition (Figure 3). In this phase, the system engineering function shall [6]:

1. Establish the system preliminary definition for the system solution selected at end of Phase A.
2. Demonstrates that the solution meets the technical requirements according to the schedule, the target cost, and the customer requirements.

---

[1] SeRANIS satellite is between 100kg-500kg.

3. Supports the System Requirements Review (SRR) and Preliminary Design Review (PDR) and ensuring implementation of the SRR and PDR actions.
4. Define development approach and plan of engineering activities.

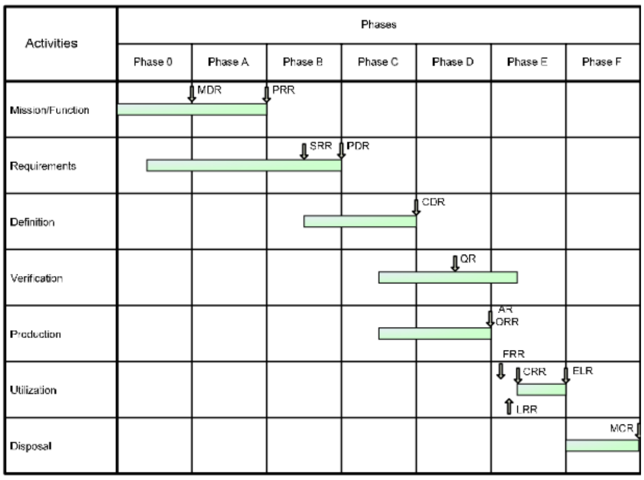


Figure 3: Typical project life cycle according to [19]

From the SRR to the PDR, the sequence of the reviews is *top down*, starting with the top-level customer and his top-level supplier, and continuing down the customer-supplier chain to the lowest level supplier. From the CDR to the AR, the sequence of reviews is reversed to “bottom up”, starting with the lowest level supplier and its customer and continuing up through the customer supplier chain to the 1st level supplier and the top-level customer. This is the so-called *V model*.

As per [19], the main review objectives for the SRR are:

- Release of updated technical requirements specifications.
- Assessment of the preliminary design definition.
- Assessment of the preliminary verification program.

The same way [ibid], for the PDR, the main review objectives are:

- Verification of the preliminary design of the selected concept and technical solutions against project and system requirements.
- Release of final management, engineering, and product assurance plans.
- Release of product tree, work breakdown structure and specification tree.
- Release of the verification plan (including model philosophy).

An ECSS standard [20] is existing for the organization and conduct of the reviews. It provides indications on how the review information flow shall be established (Figure 4), how to form the review bodies, the documentation needed for each review, what meeting are necessary, how the RID process works and how are action items issued.

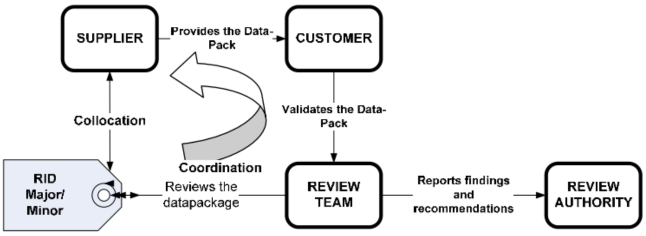


Figure 4: Review information flow according to [20]

In the SeRANIS project, even if the ECSS Standard [20] has always been used as a reference, two new processes have been developed and applied, respectively to SRR and to PDR.

3. GENERAL APPROACH

Figure 5 provides an overview of the implemented processes for requirements engineering in the development phase B of the product life cycle under consideration. The SRR-RID driven process is based on the fundamentals of the described ECSS standards and was developed according to the needs of a small satellite mission and thus follows the idea according to Incose [13], that the purpose of the tailoring process is to adapt standardized processes to satisfy particular circumstances or factors. This process refers to the first research question (RQ1) and is used between the two milestones SRR and PDR and ends with the closure of all documented RIDs.

The PDR AI-driven process refers to the second research question (RQ2) and starts with the PDR event. It represents a novel and innovative approach to the development of data, information, and requirements. The experimental and scientific nature of SeRANIS also applies to the management, organization, and project execution aspects. In order to develop the most effective and efficient approach in this regard, the Preliminary Design Review has been performed in a new format that differs appreciably from the traditional way of conducting reviews.

A special characteristic of the SeRANIS project is that no platform design was yet available in the development phase B. For this reason, in the rest of the paper, we will refer to the PDR as *Experiment* PDR, since the use of the developed process is focused on the payloads of the small satellite mission. This underlines an additional strength of the approach: the flexible implementation.

The development progress driven process takes place throughout this development phase and represents the approach to develop requirements as a function of new development artifacts.

In the SeRANIS project, a web-based data management tool (Valispace [21]) is used to store and manage data, information, and requirements. For project management and the coordination and documentation of tasks, the web-based tool Jira [22] is used.

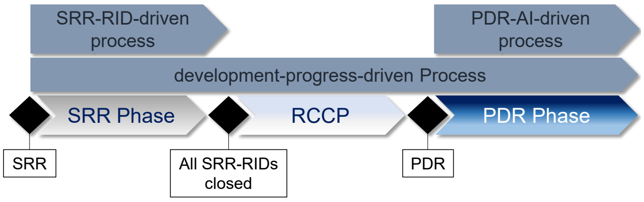


Figure 5: Requirements engineering process overview

3.1. SRR Phase

The SeRANIS SRR phase indicates that phase of the project starting with the announcement of the SRR and concluding with the closure of all the RIDs coming from it. This review has been conducted following a tailored version of the procedure described in the ECSS standards [20].
The review was conducted with different objectives: confirmation of SeRANIS system requirements (meeting mission requirements), confirmation of the payload requirements, confirmation of preliminary system design (functional baseline), confirmation that all experiment external interfaces are identified and characterized, compatibility of schedule with the project, confirmation of the preliminary verification program, confirmation that a PA Plan is available and is adequate for the SeRANIS project, confirmation of the maturity of the project for continuing with the preliminary design phase.

Of particular interest for the topic of this paper are the system requirements, payload requirements and interfaces. For them, the conditions to be fulfilled in order to consider the review successful were set as:

- The system requirements are complete and correctly formulated in a way to fulfill mission objectives.
- All payload requirements are correctly formulated and correctly describe specifications towards the platform.
- All the major interfaces space-to-ground, interfaces on the ground, interfaces between the platform and the payloads and among the payloads are correctly identified and described at a functional level.

For the SeRANIS project a data management tool is used for storing and managing Payload and System requirements. From such a tool, documents containing the requirements have been created for the SRR.

In order to confirm the objectives previously mentioned, the conduct of the review was organized following a standard sequence, using ECSS Standards, tailored for the SeRANIS mission, characterized by:

- Delivery of SRR data package
- Review of data package and generation of RIDs. Generation and tracking of RIDs done by using an appositely set environment in Jira.
- Generation and provision of responses to RIDs
- Check of responses and preparation to Colocation-Meeting
- Colocation-Meeting
    - Discussion of all RIDs which could not be closed with the supplier response.
    - Definition of action items needed for closure of RIDs.
- Release of the SRR report with decision on SRR success.

Through this review it is possible to assess the maturity of the requirements of the system and to bring them to the desired state through RIDs and Action Items when needed. In the SeRANIS project, the many experiments have a direct impact on the system. This requires that the System Engineering (SE) team always follows the evolution of payload requirements and their effects on the overall system, in a process summarized by the Figure 6, where the experimenters providing the payloads are generalized as Domain Specific Developer (DSD).

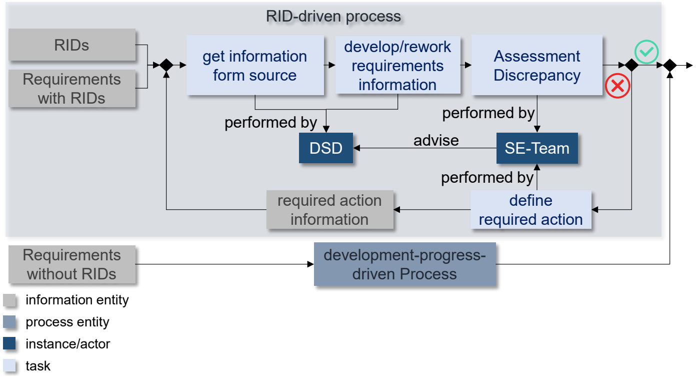


Figure 6: RID-driven Process flow

3.2. Requirements change control phase

The time frame between the SRR Phase and the PDR Phase is indicated as Requirements Change Control Phase (Figure 5).
Once the SRR Phase is concluded, all payload and system requirements are assessed and accepted in the project and used as baseline. This means that any change from that moment, has to be checked and accepted to make sure that it will not be a source of problem for the system in a later stage of the project. This phase is naturally characterized also by advancement in the design. This has often a direct impact on the requirements (especially performance requirements), that change as the design develops. It results then clear the need of a control process on the requirements during this phase, especially for the ones having an effect on the overall system, in order to ensure a proper development of the system. This process is schematized in Figure 7, where it is to highlight the function of the SE-Team as constant reviewer of the modified/new requirements that have an impact on the system. Also, the SE-Team has to coordinate with the Change Control Board for the

approval of the change. In SeRANIS CCB and SE-Team coincide, but in larger projects, they might be two separate teams.

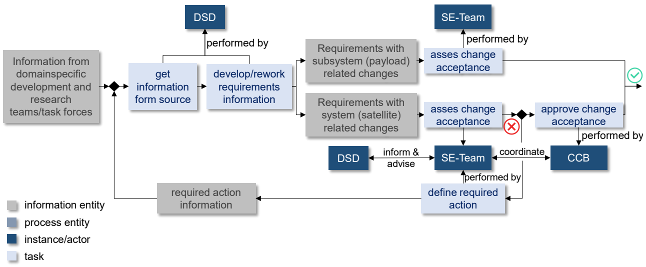


Figure 7: Development-progress driven process flow

3.3. Experiment PDR Phase

As already discussed at the beginning of chapter 3, the PDR has been conducted for the payloads only, since the platform design was not defined yet at that stage of the SeRANIS project, hence the reference to PDR as Experiment PDR.
To address the second research question, the Experiment PDR for SeRANIS has been split into three different phases:

- LE- Loading Phase
- PDR Review Phase
- PDR Board

LE Loading Phase. At the beginning of this phase, all the labs have been provided with a list of "PDR topics", containing all the topics and the level of information required for the PDR. This list has been made tailoring the ECSS Standards ([19] & [20]) to the SeRANIS mission. At the same time, on the web-based platform Valispace, a template based on the ESA EID-B for organizing and structuring information has been created for each different payload. During this phase, the DSDs have filled the information required for the PDR in the template in Valispace. Valispace is used within this project also for the requirements management and for technical budget management. The addition of also a section containing all the structured payload information allows the system engineering team to have only one source of information. Furthermore, this way, DSDs are not required to create any document anymore, and the only task they have to comply with is to always keep the information up to date on Valispace. This approach also solves the issues of inconsistencies and versioning, which is pretty common in document-based approaches. The system engineering team anyways sees the need to generate documents, especially for external outreach (for example platform procurement). For this reason, the SE team has developed a tool which is able to create automatically documents containing the information available on Valispace. This way, it is possible to overcome the disadvantage of inconsistencies and versioning issues, but still being able to create documents on request.

PDR Review Phase. For the review phase, 3 tech boards have been established in order to conduct detailed evaluation of the different payload's design in the specific areas of competence.
The boards have been divided this way:

- *Data Handling and Conops*. This board focuses on topics related to the concept of the experiment operation and data handling, within the experiment and with the platform, payload OBC or with other experiments. Software concept is also covered by this board.
- *EMT (Electrical, Mechanical, Thermal)*. This board focuses on the Electrical, Mechanical and Thermal interfaces and design, and on relative technical budgets.
- *PA, AIT*. This board covers product assurance, assembly integration and testing topics.

Individual payload designs were reviewed by the tech board in in-person meetings/workshops. The DSDs have presented the information required by using their EID-Bs on Valispace as source. During the meetings the information is explained, reviewed, and solutions are eventually discussed.
For keeping track of discussions and any open point during the technical meeting, a Jira Board has been set up for the PDR. During the meetings itself, open points were discussed between the teach board and the DSDs. Once agreed on a common solution, an action item was defined on Jira for it, in order to keep track of the future progress and state regarding the action.
This immediately highlights the benefits of this process. In the traditional RID-based way of conducting reviews, many documents are read by technical reviewers, issuing RIDs according to what is contained in the documents. These RIDs are then passed over to the document's author, for each RID a reply/discussion is needed. Afterwards the documents are eventually updated and the RID's author closes the RID after reviewing the updated documents.
It is evident that in this process, there is a back and forth between document author and reviewer that can be in some cases really inefficient and time consuming, especially if no direct communication is established.
The approach we followed for SeRANIS PDR solved this issue, allowing to resolve many points even on the spot, directly modifying the information in Valispace during the meeting, that would have otherwise (in the classical RID based review described above) resulted in the issuing of many RIDs. Not only this, the agreement of the action items together with the DSDs, ensures that the experimenter deeply understand the issue/missing information and know what is actually required and meant by the action item agreed on. This transparency highly increases the quality of the review, and even shortens the time

required for it by a large factor. The diagram (Figure 8) summarizes the phases and the actions taken in this proposed Review approach.

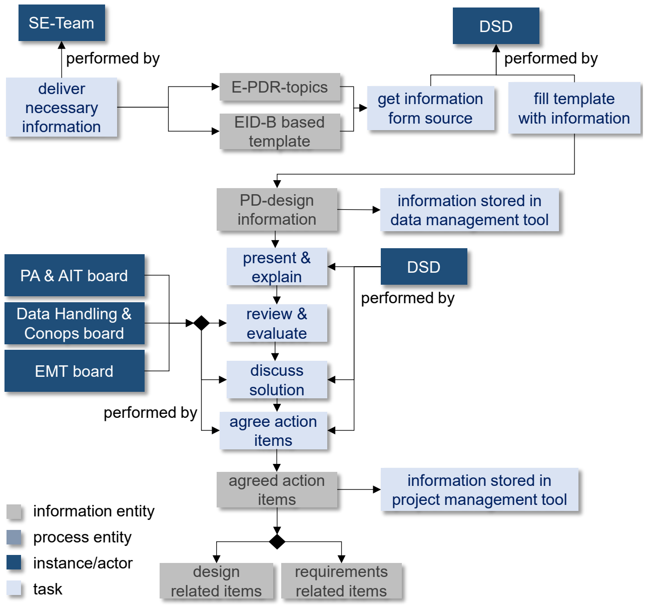


Figure 8: PDR AI-driven process flow

PDR Board. The PDR Board is formed by member of the management and project leaders that have the required knowledge and expertise in the relevant topics of the review.

Tech boards report to the PDR Board during a one-day meeting where the maturity of each experiment is discussed and assessed. After this, individual feedbacks are issued to the different labs, assessing the maturity and quality of the preliminary design, and the way forward in the project.

4. CONCLUSION

The implementation of different processes for requirements engineering in a small satellite mission with a scientific background has shown the different advantages and disadvantages of each process. For the SRR phase, a methodological approach was chosen that follows a standardized procedure. The ECSS-standards were used as a basis for developing process through their tailoring, considering the constraints of a scientific small satellite mission. This approach has proven to be very reliable and robust, leading to a consistent improvement of the maturity of requirements. Its implementation also involves less effort at the beginning of the process due to the information already available from the ECSS-standard work. At the same time, this process is more time-consuming in terms of processing the information. Limited human resources represent a constraint that has a notable impact on requirements engineering and its processes and methods for a small satellite mission with scientific background such as SeRANIS. Seamless and continuous coordination and communication between the reviewers and the domain-specific developers can therefore only be implemented to a limited extent within a tightly limited time frame. Therefore, when using this process, a higher time requirement must be considered.

The process used in the PDR phase, on the other hand, differs appreciably from the process used in the SRR phase and can be regarded as a corresponding alternative. The process proves very efficient since it does not involve documents in the review and instead makes direct communication within a defined methodological framework the central element. The shortened information processing and a reduction in processing loops provide a process that is particularly suitable for small satellite missions, which are characterized by limited resources (personnel and budget) and tight schedules for development and adaptation. Such development projects require efficient and time-saving approaches to requirements engineering. A disadvantage is that no standardized framework such as the ECSS standard is available as a basis, like in the standardized process in the SRR phase. In conclusion, it can be said that for the requirements engineering of the SeRANIS project, both processes and also the described RCCP were successfully implemented and used, thus enabling the answering of the research questions and for this purpose the elaborated comparison described in this paper.

5. SUMMARY & OUTLOOK

This paper describes the processes developed to implement requirements engineering in a small satellite mission with a scientific background. On the one hand, a requirement engineering approach is presented that fulfills both, the requirements defined in the ECSS standards and the project constraints and boundary conditions of a small satellite mission. In addition, a second systematic approach is presented that represents an innovative alternative to this and focuses exclusively on the project constraints of a resource-limited small satellite mission. In the context of the SeRANIS small satellite mission, which is being developed in a scientific environment, the two approaches have been successfully implemented and used in different development phases (SRR phase & PDR phase). Furthermore, the developed requirements engineering process that accompanied the project development activities throughout these two phases is described. Future work will focus on model-based support of the described processes to address gaps and inconsistencies in data, information, and requirements. Furthermore, a focus will be put on the further development, standardization, and scalability of the described innovative process in order to make requirements engineering and the development of data and information more efficient and target-oriented also in earlier project phases and to explore the applicability in larger projects.

ACKNOWLEDGEMENT

This research work was carried out as part of the SeRANIS - Seamless Radio Access Networks in the Internet of Space project. The project is funded by the dtec.bw - Research Center for Digitization and Technology of the German Armed Forces, with the grant number 150009910.

REFERENCES

[1] M. Hofmann, “Der_Funkmast_im_All,” *NET 01-02/21*, 2021. [Online]. Available: www.net-im-web.de

[2] S. Husung, C. Weber, A. Mahboob, and S. Kleiner, “Using Model-based Systems Engineering for need-based and sonsistent support of the design process,” *Proc. Des. Soc.*, vol. 1, pp. 3369–3378, 2021, doi: 10.1017/pds.2021.598.

[3] H. Hick, M. Bajzek, and C. Faustmann, “Definition of a system model for model-based development,” *SN Appl. Sci.*, vol. 1, no. 9, 2019, doi: 10.1007/s42452-019-1069-0.

[4] J. Feldhusen and K.-H. Grote, *Pahl/Beitz Konstruktionslehre*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[5] Gadzo E. and Mehlstäubl J., “Modellbasierte Umsetzung des Anforderungsmanagements zur Unterstützung der Entwicklung eines Experimentalsatelliten,” in *DS 111: Proceedings of the 32nd Symposium Design for X*, 27 and Sep. 2021.

[6] *Space Engineering: System engineering general requirements*, ECSS-E-ST-10C Rev. 1, European Cooperation for Space standardization, Noordwijk, The Netherlands, Feb. 2017.

[7] J. Estefan, *Survey of Model-Based Systems Engineering (MBSE) Methodologies*.

[8] U. Lindemann, *Methodische Entwicklung technischer Produkte*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[9] *IEEE standard glossary of software engineering terminology - IEEE Std 610.12-1990*, 610.121990, The Institute of Electrical and Electronics Engineers, Sep. 1990.

[10] C. Rupp, *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil,* 6th ed. München: Hanser, 2014.

[11] U. Lindemann, *Handbuch Produktentwicklung*. München: Hanser, 2016. [Online]. Available: http://www.hanser-elibrary.com/doi/book/10.3139/9783446445819

[12] J. Dick, E. Hull, and K. Jackson, *Requirements Engineering*. Cham: Springer International Publishing, 2017.

[13] David D. Walden, Garry J. Roedler, Kevin J. Forsberg, and R. Douglas Hamelin and Thomas M. Shortell, “INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities,”

[14] J. van der Ha, “Small Satellites: Continuous Improvements in Project Management,” in *Advances in the Astronautical Sciences, 01.2001*.

[15] M. N. Sweeting, “Modern Small Satellites-Changing the Economics of Space,” *Proc. IEEE*, vol. 106, no. 3, pp. 343–361, 2018, doi: 10.1109/JPROC.2018.2806218.

[16] R. C. B. A. S. and A. L. Xavier, “A Unified Satellite Taxonomy Proposal Based on Mass and Size,” *AAST*, vol. 04, no. 04, pp. 57–73, 2019, doi: 10.4236/aast.2019.44005.

[17] S. Koechel and M. Langer, “New Space: Impacts of Innovative Concepts in Satellite Development on the Space Industry,” in *69th International Astronautical Congress IAC, 1-5 october 2018*.

[18] S. Gao, W. Cao, L. Fan, and J. Liu, “MBSE for Satellite Communication System Architecting,” *IEEE Access*, vol. 7, pp. 164051–164067, 2019, doi: 10.1109/ACCESS.2019.2952889.

[19] *Space project management: Project planning and implementation*, ECSS-M-ST-10C Rev.1, European Cooperation for Space standardization, Noordwijk, The Netherlands, Mar. 2009.

[20] *Space management: Organization and conduct of reviews*, ECSS-M-ST-10-01C, European Cooperation for Space standardization, Noordwijk, The Netherlands, Nov. 2008.

[21] Valispace, *Home | Valispace.* [Online]. Available: https://www.valispace.com/ (accessed: Sep. 5 2022).

[22] Atlassian, *Jira | Issue & Project Tracking Software | Atlassian.* [Online]. Available: https://www.atlassian.com/software/jira (accessed: Sep. 5 2022).