Agile methodologies applied to Integrated Concurrent Engineering for spacecraft design

J. M. Álvarez¹ · E. Roibás-Millán¹

Received: 29 July 2020 / Revised: 23 March 2021 / Accepted: 1 July 2021
© The Author(s) 2021

Abstract

In recent years, space projects have evolved to faster and more variable projects. To adjust the design processes in accordance, new work methodologies arise, as the Concurrent Engineering (CE). This working discipline is characterized by collaborative design and the flux of information being improved by working in a dedicated environment. CE has been recently adopted by space industry for the preliminary design phase of spacecrafts and other space systems. However, this methodology does not envisage tasks prioritization, which is a fundamental aspect to achieve an optimal design solution with an efficient allocation of resources. In this work a variation of CE discipline by applying *Agile* methodologies (in which the aspect of task prioritization is essential), is proposed. *Agile* methodologies allow the proper distribution of the design effort depending on the project priorities, the state of the design and the requirements, in a continuous process to improve the design solution. The general aspects of the proposed method are presented and applied to the design of a space mission, the results being analysed and compared with to the classical CE process in order to outline its differences and similarities with CE and *Agile* methodologies and show its potential for a new environment for space project design.

Keywords Concurrent Engineering · Agile methodologies · Preliminary design · Space project development · Concurrent design facility · Collaborative design

1 Introduction and background

The development of space missions in Europe has been largely linked to the European Space Agency (ESA) [1, 2]. ESA developed the European Cooperation for Space Standardization (ECSS), an initiative established to develop a coherent, single set of user-friendly standards to be used in all European space activities [3]. The ECSS standards divided the space project into seven phases (ECSS-M-ST-10C [4]), providing the number and type of documents that should be released, as well as the milestones and specific reviews to be conducted. System engineering as a discipline is also supported and organized by the ECSS standards [4, 5], being defined as a “*collaborative and multidisciplinary approach that derives, develops and verifies a system solution that is balanced over its life cycle to meet stakeholders needs*” [6]. The traditional system engineering method

applied to space projects is based on a sequential approach, which usually follows a V-model diagram [7, 8] as shown in Fig. 1. The first step is related to the detection of a system solution and the study of its feasibility (phases 0/A of ECSS standards [4]). Therefore, a mission concept is defined through a set of requirements (upper-level system requirements). A conceptual design arises, and several design options are studied by trade-off analyses. Advantages and disadvantages are analyzed and ranked in terms of performance, cost and risk, selecting at the end the option that ranks higher overall [9]. The whole process requires several iterations, a continuous validation with the stakeholders and the need to define verification plans along with the requirements development process.

The sequential methodology is widely used in the general industry and has its own advantages. However, due to the complex nature of space systems, its application usually requires high amounts of time and includes a certain degree of complexity in the decision-making process. The interdependencies between subsystems could cause design conflicts. Usually, changes in one component have an impact (of

✉ J. M. Álvarez
jm.alvarez@upm.es

¹ Microgravity Institute IDR/UPM, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Fig. 1 The V-model of the systems engineering process, from International Council on System Engineering (INCOSSE) [8]

variable magnitude) on others and might produce a potential chain of changes in the whole system design.

In addition to the inherent complexity of space projects, the context in which they are developed highly influence the system engineering method to be used. As the nature of space projects changes, system engineering and management processes must be adapted to new environments and frameworks [10, 11]. Those new projects [12, 13] are more subject to the issues derived from the sequential design scheme, as project budget constitute a limitation in the development time and its associated cost. This has been evident in the evolution of space engineering projects, where transitions from sequential to concurrent design or from hierarchical to parallel processes have been observed in the last decades [14].

One of the methodologies that have been used in recent years is Concurrent Engineering (CE), mainly applied for the pre-design and reliability phases of the project (Phase 0/A) [15]. According to the definition adopted by ESA, CE is “a systematic approach to integrate product development that emphasizes the response to customer expectations. It embodies team values of cooperation, trust, and sharing in such a manner that decision making is by consensus, involving all perspectives in parallel, from the beginning of the product

life-cycle” [16]. CE has established itself as a methodology of great importance in all areas of engineering, its application being studied in other areas such as manufacturing or quality [17].

Although it has a large number of followers [16, 18], CE has not yet been fully implemented in the industry (except in large companies/agencies such as ESA [19] or NASA [20, 21]). However, CE has proven to be a very useful tool in the universities [22], as the requirements of the academic environment include aspects that are shared with CE, such as teamwork or design sessions concentrated in a short period. CE is being applied as a learning tool in a large number of universities [23], such as the Technical University of Delft (FireSat satellite [24]) or *Universidad Politécnica de Madrid* (MEOW Satellite [25]).

While CE is growing in the space industry, software engineering is facing a different problem that also arises from the new nature and environment of current projects. New methodologies based on new enterprise concepts have emerged to adapt the design process to new projects, in which requirements and solutions evolve through all the phases. The *Agile* methodologies [26] are a set of new concepts characterized by iterative development, continuous code integration, and the ability to handle changing business requirements to

develop software projects while allowing risk uncertainties to be managed. *Agile* methodologies have been applied to a large number of companies, having amply demonstrated their ability to improve practically any process, both in engineering or other fields [27].

CE and *Agile* methodologies (in particular, one methodology from the *Agile* family, the SCRUM approach) have certain aspects in common, such as parallel multi-tasking and the continuous design iteration as the development process moves forward. However, CE do not envisage a particular hierarchy among requirements. This means that resources are assigned for each discipline (i.e., orbit analysis, payload, structures, etc.) from start to end of the project, even if at certain times some of them do not have yet enough data to begin (or to continue) with their iteration. In contrast, SCRUM approach puts the requirements into a hierarchy, assigning more resources to the top priority ones.

The aim of this work is to present the evolution and improvements in the application of CE discipline by applying some characteristics of *Agile* methodologies, as the prioritization of tasks that, in turn, has as a consequence for the prioritization of requirements. If requirements are ordered hierarchically, the resources could be allocated on the basis of their priority so an optimization of the design process (in terms of cost and time) is achieved.

To present the method, this paper is organized as follows: the usual process of Concurrent Engineering is described in Sect. 2, which establishes the foundations of the method. A brief review about *Agile* methodologies is presented in Sect. 3 to focus on its main potentials that will be combined with CE in the new methodology proposed in this work. A description of the new methodology (focusing on its application scheme) is presented in Sect. 4. In Sect. 5, the method is applied for the design of a space mission as an illustrative example. Finally, some conclusions are presented in Sect. 6.

2 Concurrent Engineering fundamentals and its application to space systems design

Since the 1990s, CE has been conceived as a discipline to improve design and implementation processes of manufacturing systems. Although several different definition of CE can be found in the literature since its inception, the main fundamentals of CE can be summarized in four basic principles, as defined by Parsaei and Sullivan (eds.) in [28] and Smith [29]. Those principles are the increased role of manufacturing process design in product design decisions, the formation of cross-functional teams to jointly develop new products and processes, a focus on the customer throughout the development process and the use of lead time as a source of competitive advantage.

CE aims to address as many downstream issues as possible early in the design process. It relies on an integrated and parallel design of products and their related processes, including manufacturing, testing and support [30]. By facilitating such integrated design, CE makes it possible to reduce development lead times by executing the tasks simultaneously, without reducing the time required to complete the isolated design task. The reduction in delivery times can be significant and represents a competitive advantage for those companies that can apply it well.

As CE has been applied for over several decades in manufacturing engineering, it is within this field where the majority of CE variants have been developed and implemented. Indeed, it is in that field that differences between the traditional design process (“serial engineering”) and the application of CE are more evaluated. For example, methodologies as Point-Based Concurrent Engineering (PBCE) take the basic principles of serial engineering, but applying continuous evaluation and typical CE design loops. PBCE could be defined as an update from point-based design, but it still does not stray too far from the main drawbacks of point-based design [31]. The overall goal of these families of methodologies is to identify the “optimal” solution as soon as a design problem arises, and to avoid wasting time considering other options. Other methodologies exist, as Set-Based Concurrent Engineering (SBCE) that constitutes an alternative to PBCE where the benefits of working with multiple designs concurrently are included. All members of the design process “reason, develop, and communicate about sets of parallel and relatively independent solutions” [32]. This methodology has a series of advantages over the more traditional approach: it allows generating a reduction in costs both in terms of product and processes by defining multiple alternatives, it produces a reduction in development time by reusing previous knowledge and avoiding delays caused by late changes, and it produces a decrease in the risks, due to the consideration of larger sets and of several simultaneous solutions [33]. SBCE is based on a wide variety of design techniques, the basic notions can be summarized in two fundamental principles: (1) A large number of design alternatives should be considered, which will be discarded as the design progresses, and (2) Specialists must independently review a design, based on their field of expertise, and then look for regions of overlap between those solution sets in a multidisciplinary environment, to develop a final integrated solution.

2.1 Concurrent Engineering process in the space sector

Tamaskar et al. [34] identify a high level of complexity and uniqueness in aerospace systems: “*aerospace systems are different from other engineering systems as they are*

characterized by large heterogeneity of components, low number of production units, greater reliability and safety concerns, and high-performance requirements. Modern aerospace systems, which are designed to provide high performance and reliability while operating in extreme environments, have exposed the limits of conventional systems engineering tools and practices”.

Indeed, it is mainly due to this intrinsic complexity of space projects, that a modified version of CE has been adopted in the space industry. This version takes those CE principles that may be applicable to the design processes of a space project (e.g. the design of a spacecraft), and ignores or discards those that cannot be applied or are meaningless in this sector. In addition, it should be noted that due to the high reliability demanded to space systems and the high level of complexity in the design requirements, the CE concepts have been applied (at least currently at the time of this work) only to preliminary design stages (evaluation studies, phases 0/A) [6, 15, 16, 18].

Space projects are characterized also by their inherent uniqueness along with strong relationships between their elements. Those attributes mean that the impact of any change must be evaluated through all phases of the project life-cycle [34]. CE has demonstrated that the magnitude of this impact can be immediately identified and assessed. As a result, an early error detection is ensured, reducing the inefficiencies in the design process [25]. However, the use of CE procedures has a problematic side: as the amount of data generated (in terms of technical parameters and design information that must be collected and shared) is large, establishing an environment of data managing (storing, ordering and sharing) becomes a basic need. Efforts have been made in the last decade to create a set of open data exchange standards [18], allowing streamlined communication and data sharing between multidisciplinary design teams and the customer. This practice is known as Integrated Concurrent Engineering (ICE) [35, 36], and it is focused on the optimization of the design process in terms of time, replacing some of the physical and organizational boundaries by direct communication. Other studies have already focused on the complexity of space projects and the need to include collaborative design methods in the space design processes [37], which is not the same as CE, but its an important part of it.

CE concept, as is understood in the space industry, is based on two key elements: the idea of task parallelization (or performing tasks concurrently) and the need for taking into consideration all the elements of a product life-cycle from the beginning of the design phase [14]. CE methodology applied in the space sector maintains the full concept of collaborative design, in which different design experts work on each of one of the different subsystems that are involved in a space mission.

Another well-known property of CE is based on working with multiple designs at the same time. Several solutions are considered and analyzed simultaneously until the very last moment in the design process. Nevertheless, although this concept belongs to CE discipline and other variants such as SBCE, in the space industry it becomes somewhat blurred, due to the high level of complexity in the requirements. It means that, at the mission level, normally there are not many solutions that can meet the requirements without increasing the design risks. However, at subsystem level, an expert performing the subsystem design works with all the valid design options for this subsystem. The multiple solutions are reduced in an iterative process in which all the mission constraints and subsystems (and their multiple solutions) are analyzed and evaluated until convergence.

The application of CE is based on a continuous flow of system information, which is shared as parameters and design values, previously defined as inputs or outputs in a database. This means that all subsystems of a space project, and also the interfaces between them, must be defined as design parameters. It should be highlighted that these design parameters define a *parametric design* environment, which has proven to provide great advantages over other design concepts [38, 39]. CE philosophy implies an iterative design working environment: the subsystems are defined and the information is shared and updated until the mission requirements are fulfilled. Each step of this process is called an *iteration*. Once a closed design is achieved within an iteration, a new iteration arises so that the necessary modifications of design parameters are carried out. This procedure allows achieving an optimization of the previous iteration, whether of size, cost, weight, performance, etc. Once the iterative process has been completed (i.e., all the technical requirements and mission objectives are achieved), the results are studied in terms of feasibility and the strategies for verification are initially defined.

CE has demonstrated significant improvements over the traditional sequential approach [40]. Their application to complex problems allows assessing the impact of changes in the requirements from the beginning through the whole design process. It is well known that the time needed for starting the production of an element increases as the number of design changes increase. This effect is shown in Fig. 2, where a comparison between CE and traditional (or sequential) engineering is presented, making it clear that the application of CE saves time and facilitates the convergence of the solution to an optimized one.

As CE provides a collaborative, co-operative, collective, and simultaneous engineering working environment, their application contributes to avoid interface issues or errors due to design changes not being communicated fast enough between specialists. This advantage directly impacts in maintaining the high quality and reliability standards of the

Fig. 2 Comparison between CE and traditional (or sequential) engineering in terms of number of design changes. Adapted from [41]

space industry. In addition, the time reduction in the information sharing process allows to control, verify and identify inconsistencies of the space project budgets (as mass, power, volume and cost).

CE approach takes advantage of modern information technology, with a team of experts from different disciplines working simultaneously in a properly equipped facility and using a *model-driven* design process, allowing all the design parameters to be dumped into a single model that includes the main properties of the product (e.g. the spacecraft).

2.2 The concurrent design facility

A Concurrent design facility (CDF) is an advanced design room, equipped with networked computers, multimedia devices, and different installed and interconnected software. This properly equipped room allows implementation of the CE methodology in the development of space missions through interaction between design experts, data exchange, and information sharing (projectors, screens, etc.).

As mentioned, the IDR/UPM Institute currently has its own CDF at the UPM Montegancedo campus (Madrid, Spain), developed by an institutional agreement with the European Space Agency (ESA). The CDF consists of a software infrastructure that includes tools for the generation of the different models, integration between them to propagate data in real time, a documentation system, and storage capability. Several calculation models are available for each subsystem specialist to calculate and simulate their respective subsystem variables, which serve as tools to communicate with the database, thus sharing the design parameters,

as well as facilitating the necessary calculations to define each one of the subsystems involved in the design of a space mission. Depending on the characteristics of the mission those calculation modules can be adapted by including new functionalities to be as complex as needed, or combined with commercial engineering software (i.e., NASTRAN/PATRAN®, ESATAN-TMS®, CATIA®, etc).

In addition to the database management system, this facility includes 13 workstations, an audio-visual distribution system, and a videoconference system (see Fig. 3). The CDF is being used and updated since its inception, by including the implementation of several sets of calculation modules: (1) the Open Concurrent Design Tools (OCDT), a set of design modules based on Microsoft®Excel®, developed by ESA [42], and (2) a set of design modules under development, programmed and implemented by IDR/UPM staff and students.

By applying CE in a CDF, the communication between the heads of the different subsystems is highly improved, since they are all in the same place at the same time. The CDF allows for instantly updating any design modification that may affect other subsystems, so it is transmitted to all the computers in order to share this update to the rest of the designers.

3 Overview of the Agile methodologies

The *Agile* methodologies are a set of tasks and procedures aimed to the management of projects in which the result is planned, created, verified and improved. They are those

Fig. 3 Concurrent design facility (CDF) of the IDR/UPM Institute (Madrid, Spain). General design (left) and lateral view (right)

development methods in which both the needs and the solutions evolve over time. This evolution occurs through teamwork of multidisciplinary groups that are characterized by having the following qualities: evolutionary and flexible development, team autonomy, planning and communication.

The principles that sustain *Agile* methodologies are focused on reducing the constraints of development procedures and on the development of quality products and services that respond to customer needs, whose priorities often change during the course of the project. Some key points are the need for self-organizing teams and the reduction of documentation (maximizing and improving its content as the project progresses), to increase speed and flexibility in the development process. These main properties are collected in the *Agile Manifesto* [26], and can be summarized in the following four main values:

- “Individuals and interactions over processes and tools”.
- “Working software over comprehensive documentation”.
- “Customer collaboration over contract negotiation”.
- “Responding to change over following a plan”.

In addition, *agile* methodologies have undergone constant evolution and adaptation as their application has increased in different industries, with different sizes and products. All this evolution and enforcement are collected annually in the “agile status quo” [43], a complete report that includes the main factors and results regarding the application of current *agile* methodologies.

A comparison between the main properties of the *Agile* methodologies and traditional project management methodologies are shown in [44]. Both traditional and agile-based software development methods have their own advantages and disadvantages. However, whenever possible, the *agile* approach should be considered, as it provides certain advantages, including the speed at which a functional product is

achieved; and profitability, since making changes is less expensive than with the traditional approach.

3.1 Main *Agile* families: definition and properties

First, it is important to mention that all the variants of the *Agile* methodology have a series of advantages, which are: (1) Improve quality, minimize deliverable errors and improve customer experience and functionality; (2) Increase commitment, improving employee satisfaction and generating team awareness; (3) Speed, as it shortens production cycles and minimizes reaction and decision making times; and, (4) Increase productivity, by better allocating resources and improving production dynamically according to the priorities of the company. Some of the rules of the *Agile* methodologies have emerged as agile teams have learnt how to work together.

However, each rule should be applied in a particular context, so that each team can find the basic rules that allow them to better apply the *Agile* fundamentals. In particular, the following rules allow *Agile* concept application:

1. When there is uncertainty in a certain functionality, very optimistic estimates should be avoided.
2. Always have functionalities on the “bench” that can be developed in case things go better than expected.
3. The priorities of the functionalities shall be very well defined. Thereby, if things go worse than expected, one can be agile and quick identifying what should be left for the next iteration.
4. During the iterations some time shall be reserved for contingencies. It could be a certain percentage of the capacity of the team whose value will vary from iteration to iteration depending of several elements (support to previous deliveries, support to software in production, user training, etc.). If at the end nothing happens, the

functionalities that are in “the bench” could be developed.

5. Give immediate visibility to the client of unforeseen situations and their impact on development. Engaging the customer in daily meetings is key to making agile decisions.
6. Use checklists in everything the team learns and that is essential to ensure predictability.

Although there are different options for the application of *Agile* methodology, from the authors point of view, it should be highlighted the following ones: Scrum, eXtreme Programming (XP), feature driven development (FDD), adaptative software development (ASD), dynamic software development method (DSDM) and Crystal; the six being the most used alternatives found in the literature.

3.1.1 Scrum

Scrum [45] is a process framework based on *Agile* principles, characterized by the use of iterative and incremental life-cycles for product development and a managing philosophy based on roles, responsibilities and meetings. Through SCRUM, the product is released periodically so that the problems that may arise during the project development process are solved as soon as they become evident. This working scheme seeks to apply good collaborative work practices (as a team) facilitating the finding of optimal solutions to the problems that may arise in the project development process. Scrum is a flexible, adaptable, empirical, productive and iterative method that uses the ideas of industrial process control theory for the development of software systems [45].

The development process starts by following a road map that shows the high-level functional objectives and their associated temporal constraints, serving as a guide for task prioritization, so the effort is focused according to these priorities. With Scrum, regular and partial deliveries (sprints) of the final product are made, all of them with a previously established priority that arises according to the benefit they bring to the client. Therefore, the risks that may arise from extremely long developments are minimized. Scrum is specifically indicated for projects in complex environments, where results are needed immediately and where the following aspects are essential: innovation, productivity, flexibility and competitiveness.

3.1.2 eXtreme Programming

Known by its acronym XP (eXtreme Programming), it is a methodology based on a set of rules and good practices for developing software in highly changing environments with imprecise requirements. Therefore, it is focused on continuous feedback between the development team and the client.

XP is characterized by six phases: Exploration, Planning, Iterations to first release, Production, Maintenance and Death [46]. It is focused on enhancing interpersonal relationships as the key to success in software development, promoting teamwork, caring for developer learning, and fostering a good work climate. In addition, the individuals and the interactions among the development team on the process and tools are valued. People are the main success factor of a software project.

For this reason, as the project starts, all the requirements must be defined and then the effort must be invested in handling the changes that occur, thus minimizing the possibility of error. XP is based on simplicity and aims to improve customer satisfaction.

3.1.3 Feature driven development

Feature driven development (FDD) methodology is geared towards large teams (using larger teams than other *Agile* methodologies such as SCRUM). The *Agile* FDD methodology includes the figure of the project manager and an architecture phase.

It is concerned with quality, which is why it includes constant monitoring of the project. It helps to counteract situations such as over budget, program flaws or the fact of delivering less than desired. FDD provides guidelines, tasks, techniques and five sequential processes: Develop an Overall Model, Build a Feature List, Plan by Feature, Design by Feature and Build by Feature [46]. FDD software development focuses on iterative design and construction of the software product [46].

3.1.4 Adaptative software development

Adaptative software development (ASD) grew out from a rapid application development methodology powered by Highsmith [47]. ASD incorporates the principle of continuous adaptation, providing a work framework that encompasses concepts, guidelines and practices in order to encourage rapid, incremental and iterative development of complex systems with constant prototyping. In other words, its principle is to adapt to change instead of fighting against it.

3.1.5 Dynamic software development method

The dynamic software development method (DSDM) *Agile* Project Framework is conceived for its application in a wide range of projects, from small software developments to large projects. Born in a business environment, it allows applying an *Agile* approach in corporations accustomed to working with more traditional projects. The basic principle of DSDM is that the resources and timeframe are adjusted and then the

Fig. 4 Scheme of the design process of a space mission in a CDF [2]

goals and the required functionality (that is not fixed) are adjusted accordingly [48].

3.1.6 Crystal

Crystal is an *Agile* software “family” of methodologies. It is subdivided into several types of methodologies depending on the number of people who are going to conform the project. However, all of them have a “genetic code” in common, since it shares a human orientation but done in a different way as the other *Agile* methodologies [49].

It puts a lot of weight on the revisions at the end of the iteration, encouraging the process to apply continuous improvement techniques automatically. Its assertion is that iterative development is to find problems early and then allow them to be corrected by placing more emphasis on people monitoring their process and improving it as development progresses.

One of its main characteristics is the vital importance given to the developers that make up the working group, which is why its study points are aimed at: Human aspect of the team, Team size, Communication between developers, Policies to follow and Physical workspace.

4 The proposed method: *Agile*-based CE methodology

As aforementioned in Sect. 2.1, all space projects have some points in common, being the main characteristics the uniqueness and the complex interdependence between the different disciplines involved (i.e., between subsystem technical specialists, scientific teams, quality and product assurance, project management, and others). When a pure Integrated Concurrent Engineering (ICE) approach is applied for the assessment phase within a CDF, the situation presented is similar to the one illustrated in Fig. 4. Starting from the definition of the technical requirements and constraints, and using the available schedule and resources, the design process arises, mainly focused on the subsystem definition and the relationships among them. Inter-dependencies are treated as design parameters and variables, which are shared between each participant or specialist.

When a result study (*iteration*) is obtained, a description of the spacecraft (S/C) design, its configuration, and constraints (in terms of cost, risk, and programmatic aspects) is provided. The refinement of the design is achieved through successive iterations of the same process.

One of the weaknesses of the ICE process applied in a CDF is the relative isolation of subsystem parameters, from the point of view of the specialist working module. While the process is fully coupled (i.e., all subsystems are inter-related among them), the specialists tend to have a partial view of the design, only knowing in full detail their own subsystem design. Indeed, only partial information about other subsystems designs is provided. This drawback is caused by the main design philosophy of the CE process, in which the information about other disciplines is only provided in terms of numerical values (the parameters inputs and outputs of the subsystem designs). This issue may be amplified when each subsystem development have different critical paths depending on the specific technical requirements of the project.

The proposal of this work is to develop a solution to solve this particular aspect of ICE application, by proposing an *Agile*-based methodology but focused on the CE. In particular, ICE methodology will be applied and, at the same time, some aspects of *Agile* philosophy will be integrated. The new method presented is therefore an attempt to merge and unite the potentials of both methods from the point of view of the space sector needs.

Within *Agile*-based techniques, a Scrum scheme will be implemented and combined with ICE. This decision is based in the fact that Scrum has been demonstrated as a good methodology for managing small and specialized teams where finding the problem solution based on a set of requirements is the goal [45]. In addition, Scrum framework is applicable to any project with aggressive deadlines, complex requirements and a certain degree of uniqueness, making it attractive for space industry where reducing development times have a significant impact on the development cost reduction.

Scrum principles were written by software engineers, but in the author's opinion, they can be adapted to space engineering. The main difference between software and space projects is the nature of the goal on which these projects are focused. Space projects are focused on the development of a physical system (the spacecraft or the space equipment) that requires both hardware and software. This fact completely changes the paradigm, as hardware is not as revisable as software. Therefore, tasks such as manufacturing and testing cannot be programmed as quickly and frequently as the *Agile* method suggests. In addition, as *Agile* methodologies have grown, and as they have been implemented in a greater number of companies apart from the IT sector, different studies analyzing the difficulties of their implementation have emerged over the years [50, 51]. Nevertheless, in the first phases of a space project, in which no hardware is involved and *only* a design is pursued, are susceptible to being managed with the Scrum approach by forming cross-functional teams focusing on the priorities of the on-going phase.

Before presenting the new methodology (merging both ICE and Scrum), a brief summary of Scrum characteristics is provided. As aforementioned in Sect. 3, Scrum is based in a prioritization of the requirements, to focus the design effort to achieve the requirements that affect directly the needs of the customers. In addition, Scrum facilitates the search of optimal solutions to problems that may arise in the development process.

Scrum proposes regular and partial deliveries (called *sprints*) of the final product, all of them as a result of a previously established priority, which is determined according to the customer needs. It also includes daily meetings regardless of whether the prioritized requirements are met (*sprints*), called *daily sprints*. This structure allows to minimize the risks that may arise from extremely long developments. Scrum is specifically indicated for projects in complex environments, where results are needed immediately and aspects as innovation, productivity, and flexibility are essential. The objective is that, in each one of those *sprints*, all the improvements are put in common, so that all the experts have a global knowledge of the whole project. This knowledge is important to optimize the design and facilitates the decision-making processes.

The Scrum concept is outlined in Fig. 5 and it consists of two main phases. The first one is where the requirements to be fulfilled in the current iteration are selected from the whole requirement list (or *requirements backlog*), defining the *sprint backlog*. In the second phase two steps are held, one with a very low time-cycle (the *short cycle*) and one with a higher time-cycle, the *sprint* itself. In the first step, the state of the project in the current iteration is shared, including the definition of problems that may have arisen from previous iterations. The actions needed to solve them are discussed here. The *sprint* is performed once the tasks are precisely defined and all stakeholders have a clear vision on the main activities required in the current phase. As results of the *sprint*, a solution to the problem (the upgraded design or product) is released. A new iteration could be necessary to proceed with the remaining requirements from the *requirements backlog*.

If approaches from ICE and Scrum are compared (see Figs. 4, 5), one may realize that the main difference is the allocation of resources during the course of the project. If all requirements are treated with the same level of priority (as performed in CE for space projects) and the resources are distributed between disciplines, this distribution may not be optimal. In certain circumstances, the mission requirements list can be organized to focus the initial effort on the most restrictive ones. Indeed, during the preliminary design of a space mission, some subsystems usually require certain parameters (as inputs) from other subsystems that are not defined at that time at the required level. This situation typically leads to a waste of time for the specialists assigned to

Fig. 5 Development scheme of the Scrum *Agile* methodology

subsystems requiring a high level of definition to set their inputs.

The methodology presented in this paper aims to solve this issue, exploiting the advantages of Scrum into ICE to assign resources and prioritize tasks. Space design, especially in space platform and satellite design, is characterized by fully focusing the design effort on requirements. Indeed, the ECSS define the design phases around the requirements. First phases of space projects are focused on the definition of the Mission Requirement Document (MRD), where the technical high-level requirements are defined and extracted from scientific needs. Those requirements define the mission in a broad aspect, as the orbital regime, the size of the system, the budgets -mass, power, cost-, etc. After MRD definition, the next phase is focused on the specification of the System Requirement Document (SRD). At this step, the requirements of the different subsystems are generated from the high-level requirements, marking the starting point for the system preliminary design.

ICE is usually used for preliminary design because allows classified the design process in several simultaneous small designs at subsystem or discipline level. For each discipline (i.e., structures and mechanism, attitude determination and control, electrical power, on-board electronics, communications, payload, etc.) several solutions are studied and compared by trade-off analysis. From this study, one or several subsystem solutions are developed and shared to find an optimal solution for the complete mission in terms of mass, power, cost and reliability. This process, although considerably faster than traditional design, often requires several iterations to identify incompatibilities between subsystems and their interfaces, due to the partial isolation between subsystem specialists (only a few set of parameters is shared between disciplines).

The methodology proposed here is based on the following principles:

- The mission requirements are prioritized, establishing an hierarchical order for subsystem design, depending on the number and the complexity of their inputs. This classification shall be performed retaining the versatility of the CE design scheme from Fig. 4.
- The available resources are distributed considering the requirements priority. In practice, this means that more resources will be allocated to fulfill primary requirements.
- The design process is broken down into small increments and tasks, instead of approaching the problem with large work packages. In each short iteration all members of the team work in the areas of subsystem design, analysis, and testing/review/quality, will update and share their progress with the rest of the team, by scheduling daily quick meetings. This process is similar to the first steps in a CE process, but in this case every member of the team works in the discipline or subsystem with a higher priority in the current iteration, so everyone gets a global view of the final solution.

To clarify the method, a possible design sequence for the development of a generic spacecraft is included in Fig. 6. It should be noted that the hierarchy between disciplines presented in the scheme of Fig. 6 refers to the definition of the subsystem *inputs/outputs* and their connections and, therefore, marks the start of the design stage for each of them. Additionally, it should be said that some activities might be performed in parallel (i.e., it could be possible to start the orbit definition before the end of the payload design).

Fig. 6 Scheme of the design process of a space mission in a CDF prioritizing the subsystem that needs a low level of development to define their inputs

The first step defined in Fig. 6 belongs to the payload definition and, therefore, to the main objectives of the mission. The payload requirements usually define the mission orbit. Without the knowledge of at least the main parameters of the orbit the power subsystem design cannot start, as the uncertainty in the orbital parameters leads to the absence of essential inputs. Likewise, defining a possible configuration for the spacecraft without the characterization of the subsystems (at least in the most relevant parameters, as mass, size, or number of elements) is complicated. It could lead to a high number of different configurations that shall be studied in later stages.

While ICE proposes to assign an expert (or a team, if required) to each subsystem, therefore performing the design of all subsystems at the same time, the proposed approach suggests the reallocation of the project resources at every step of the process, to the design experts who are involved and working in the design. Therefore, in the first stages within each iteration, the principal resources of the project are distributed based on the priorities. In practice, this means that a greater number of design experts will be assigned to the disciplines that have priority in the design (due to their design parameters). This allows increasing the capacities to study different possible design options for the most critical subsystems for the development of the mission. This way, as the project progresses and the iteration is being completed, the human resources are reallocated to more backward subsystems in the scheme (i.e., subsystems that require a greater level of detail in the design to establish their inputs).

Using this work concept, the design team has knowledge of the whole project (not only by the meetings that take place to inform about the design situation or the input parameters of each design module), as they have been part

of that design and have participated in the proposed solutions. However, it could facilitate the process if every discipline has always assigned one design responsible, to analyze the potential implications of the decisions in the subsystem level design.

A scheme of the proposed methodology is shown in Fig. 7. At the first step, the requirements and constraints lists (objectives, environment, lifetime, payload, reliability, schedule, technology, and budgets) in addition to the study requirements lists (budget, study level, planning, and resources), are used to define the mission requirements list.

These requirements define the mission concept and they are the primary foundation for the S/C design. In each iteration (*sprint*), the System Scrum manager (a combination of the System Engineer and the Scrum Master), along with the ICE Team and the Product Owner (The customer, typically the science team), define the mission requirements prioritization list depending on the particular properties of the mission and the design process scheme (which may be different for each mission). The design sequence and the engineering efforts inside of a sprint are prioritized, as explained in Fig. 5.

In each *sprint*, there also exist various progress meetings (the short cycles in SCRUM terminology) to share the state of each discipline, including all the information relevant for the whole project to anticipate potential risks. At the end of each sprint, the mission requirements prioritization list is revised and new requirements are included, re-distributing the human resources to the new scenario. At the end of the design process, when all the requirements are included and discussed, the optimized solution is obtained (S/C design, S/C configuration, launcher definition, risk analysis, cost analysis, simulations, etc.).

Fig. 7 Scheme of the SCRUM method applied to ICE for spacecraft design

Within ICE process, another advantage of this proposal is the reduction in the number of parameters that must be used simultaneously, reducing the amount of data to be shared and actualized. This might result in a less demanding system in terms of data environment, widening ICE to missions of all sizes and characteristics.

5 Application case: results and discussion

The IDR/UPM Institute has developed over the years a considerable expertise in conducting feasibility design sessions at its CDF, in both academic and industrial projects. Among these projects, it could be highlighted the phase 0/A design for various micro-satellites (50 kg-class), as the UNION LianHé [52] or the UPMSat-2 (successfully launched on September 2nd 2020, in the VV-16 (SSMS-POC) Vega flight), whose missions are devoted to technological demonstration of equipment and payloads developed by the space industry. Among other projects, it could be emphasized the NANOSTAR project¹. It is articulated around an international consortium (European universities, aerospace clusters,

and ESA Business incubation centers), whose objective is to support emergent technologies in nanosatellites, using them as a learning tool of first level. Within this project, several CE studies have been performed, including challenges between teams of young aerospace engineers or students from several universities to compete for the creation of innovative designs for space missions with multiple purposes: CubeSats moon-flyby missions, missions dedicated to studying the survival in space of marine photosymbiotic worms, satellites formation flight mission at L2, etc. Another milestone at the IDR/UPM Institute's CDF was the participation in the 1st ESA Academy Concurrent Engineering Challenge, organized by ESA Academy's Training and Learning Centre. IDR/UPM system engineers, along with a team of Master students, developed a 300-kg satellite to identify feasible areas for a future human base in the Moon surface [25].

In the IDR/UPM institute, and in most of the facilities where ICE is applied in the space sector, the CE concepts are mainly used for the preliminary phases of the space mission life-cycle. In those phases, space engineers must study the system as a whole, in order to find a solution that complies with the mission top-level requirements. Several design options are investigated, both at the design level of the sub-system itself, where multiple options are studied; as well as at the level of the complete platform, where the different

¹ <https://nanostarproject.eu/>

Table 1 Mission-level (top-level) requirements list for CDF sessions (*groups 1 and 2*)

ID	Definition
RQ-001	The S/C shall be launched from the ISS
RQ-002	The S/C shall include an imaging system with 600x400 pixels resolution for Earth observation (Mount Fuji)
RQ-003	The S/C-ISS separation shall be imaged with a resolution of at least 0.5 m at a distance of 500 m
RQ-004	The S/C De-Orbit Mechanism (DOM) shall be treated as a black box with 0.3 kg mass and $0.1 \times 0.1 \times 0.05 \text{ m}^3$.
RQ-005	The S/C DOM shall be positioned faced to the velocity vector
RQ-006	The S/C shall accomplish with the CubeSat design specifications

options achieved are compared depending on the fulfillment of the requirements. Therefore, as the mission is more subjected to changes, the advantages of applying CE are more accentuated, due to aspects as fast reaction to changes and time savings. It is in this context where the application of SCRUM approaches within the ICE methodology shows its further benefits.

In the following subsection, a case study performed in the IDR/UPM CDF is presented. The main objective is to illustrate comparison between the application of the traditional ICE procedure (the used for the aforementioned space projects) and the new proposal of integrating Agile methodologies within ICE. This application example aims to show the possible improvements that have been noticed as a result of carrying out several studies in the CDF. Sessions were guided for professional system engineers. The design experts responsible for each activity were students from the M.Sc. in Space Systems (MUSE) of UPM [53]. Therefore, the profiles of designers in this experimental session are space engineer graduates (i.e., early career professionals) with specialized training in ICE design in CDFs [54].

5.1 Application case: comparative CDF sessions

As mentioned, a comparison between the solutions achieved by using usual ICE and by using the proposed methodology, applied to the preliminary design of a space mission, is presented in this subsection. The design case was carried out by a group of 15 aerospace engineers (MUSE students when the study took place) and 2 system engineers, IDR/UPM professors specialized in system engineering, orbital analysis, power generation and distribution, etc, and who have directed several CDF session.

Two express CDF sessions were performed (3 days), implementing the two design methodologies: (i) the first session, called *Group 1*, managed the session as is usual in ICE (Fig. 4); (ii) the second session, *Group 2*, implemented the modified *Agile*-based CE methodology proposed in this paper (Fig. 7).

The CDF infrastructure and calculation modules used in both sessions were usually used by IDR/UPM staff for space missions design. Therefore, the parameters shared among

subsystems, the variables of a dependent nature, and the causality criteria employed are already well established. For example, structural parameters depend on payload size and launcher selection, the power subsystem depends on payload consumption and orbit parameters, etc. The type of components used for a particular subsystem (for example, the type of actuators and sensors that compose the Attitude Determination and Control Subsystem) and the possible design options were studied and compared by the specialist in charge of the subsystem design, and advised by the System SCRUM Manager (system engineers).

Both sessions entailed the phase 0/A design of a CubeSat to image the Earth's surface. In particular, the main objective was to study the thaw on Mount Fuji (Japan). The mission was presented to the two groups through a technical report, obtained after a meeting with the project manager and the science team. Some of the more complex requirements were simplified, so that they could be met in a reduced session of 3 days (e.g., the thermal requirements were simplified to a margin for the entire S/C). Although the mission was adapted to the level of knowledge of the students, it was intended to be comparable to an usual CDF mission design session for more complex missions and systems.

The mission-level (top-level) requirements were the followings: the S/C shall be launched from the ISS; the payload shall include a system to obtain and process images of Mount Fuji and at the time of separation between the satellite and the ISS; and a de-orbit system is required at the end of the mission. The entire mission requirements list is shown in Table 1. In addition, the design was carried out by complying with the CubeSat design specifications (standard) and minimizing mass and cost.

The main steps followed to obtain a preliminary design complying with the mission requirements were similar in both approaches. First, a study of the mission (high-level) requirements was carried out, to derive the corresponding technical (system-level) requirements for the subsystems. It should be noted that, in any space project, high-level and system-level (or low-level) requirements shall be distinguished. High-level requirements are related with the overall mission, in terms of its performances and constraints. On the other hand, system-level requirements are related with

the particular performances of the subsystems, in terms of purpose (i.e., conditions under which the subsystem is used), operational constraints, resource utilization and operational modes (i.e., nominal, contingency, etc). The combination of both high- and system-level requirements provides a requirement tree (or function tree) showing their inter-dependencies. At last, a technology matrix is developed listing the system requirements/functions and for each, a potential technology or technological element. Some information is also delivered as the maturity of the technology (technology readiness level); proof of company's maturity concerning the knowledge and master-ship of the technology, including a description of the necessary technology acquisition activities, and the identification of potential risks (e.g., technology availability, programmatic and financial aspects).

Once the system-level requirements are extracted and studied and the technology matrix is developed, the design process could begin. In fact, *Group 1* started the design as usual in ICE, with the study of similar past missions, the assignation of responsibilities and the initial search for solutions. However, in the case of the *Agile*-based ICE proposed here, an intermediate action is needed. The system-level requirements are hierarchically ordered so that a prioritized requirement/function tree is extracted. This action defines the coherent order for the subsystems design and marks a prioritization of tasks.

It should be also appointed that, due to the academic definition of the mission, in the *Group 2* design sessions carried out the roles of System Scrum Manager and Product Owner (in the terms defined in Scrum methodology) were unified in a single person jointly undertaking both tasks.

After those initial steps, as previous designs are essential for CE processes, the missions database was checked by both *Group 1* and *Group 2* to find past missions with a certain level of similarity. This action is essential to study previous solutions and implement lessons learnt from past missions. Later, the design phase began, where solutions at subsystem level were identified and checked with the technology matrix. As usual in ICE, within each subsystem several solutions complying system-level requirements were proposed. Trade-off analyses were subsequently carried out in a search for low-risk solutions with reduced mass and costs budgets.

At that point, an iterative process started for both session groups. For *Group 1* design, one iteration implies that a complete solution for the overall mission is achieved since all system requirements are retained (subsequent steps will be dedicated to cross check mission design and requirements and starting a new iteration). The duration of an iteration is very variable (depends on the complexity of the mission) but, for this application, only two iterations were performed in the three days session.

For *Group 2* the process is significantly different. Since system-level requirements have been prioritized, a reduced

number of conditions and constraints are presented in the process. Each iteration, called sprint in this case, is faster and allows the study of a big set of possible solutions. After each sprint, possible solutions are analyzed and compared in a search for the optimum. The duration of the sprints is conditioned by the number of requirements included but, for this application, about three sprints by session day were needed.

In the following subsections, the work carried out and the goals achieved in the two CDF sessions, *Group 1* and *Group 2*, are described. To show a better comparison in terms of the gradual evolution of the design process, the description is organized day by day.

5.1.1 CDF session, Day 1

Group 1 was divided into all disciplines according to the different subsystems (one or two engineers per subsystem depending on its complexity). After the definition of the roles, a detailed study of the imposed requirements was carried out and the system-level requirements were derived. Once the mission requirements were defined and understood, the design of all the subsystems began.

In the case of *Group 2*, the definition and study of the requirements includes one additional step: the prioritization, directed by one of the system engineers that acted as System SCRUM Manager and Product Owner (see Fig. 7). Once the requirements were sorted according to their priority, a subgroup of them was selected and the mission design began, focused on the fulfilment of the selected requirements. As a consequence, the subsystems also were prioritized, according to the mission requirements and the defined design sequences (see Fig. 6). The engineers were distributed according to the subsystems and the requirements that had been selected for the current sprint. Specifically, at the first step, they were divided between mission analysis, payload, and power subsystem. This prioritization will be usual in many space missions, where payload performances and constraints define the objective orbit and the power required by the system and, accordingly, the power subsystem design is significantly constrained by the selected orbit.

The subsystems distribution, both for *Group 1* and *Group 2*, is collected in Table 2. Prioritization has been designated by a letter, from higher priority (A) to lower (C), both for the mission requirements and the subsystems. The prioritization of the subsystems was carried out according to the requirements hierarchy.

At the end of the day, *Group 1* was able to advance in the design of several subsystems, typically orbital analysis and payload. However, at this time, some subsystems (as communications, ADCS, or OBDH) do not have enough information (parameters or inputs) to begin their design. As usual in ICE, the initial inputs of those subsystems are estimated using statistical data from previous designs (those

Table 2 Subsystems definition and prioritization for the two CDF sessions conducted at the first day

Subsystem	Group 1		Group 2		
	Teams size	Prioritization	Team size	People increment	Prioritization
Mission analysis	2	None	8	–	A
Power	2	None	3	–	A
Communications	1	None	0	–	B
Ground Station	1	None	0	–	C
Payload	1	None	4	–	A
ADCS	2	None	0	–	B
Propulsion	2	None	0	–	B
Thermal	1	None	0	–	C
OBDH	1	None	0	–	C
Structure/Configuration	2	None	0	–	B

from the missions database of the CDF that were found to have similarities with the proposed missions). However, the effectiveness and utility of the estimated parameters will depend on the extent of the available database, as well as the particularities of the mission being carried out. As usually occurs, such data are more subjected to changes in the forthcoming iterations.

On the other hand, *Group 2* presented a slower start. Indeed, a fraction of the session time was needed to define the prioritized requirements and subsystems from the requirements tree. Two teams of two engineers were included in the payload, four teams of two engineers were assigned to mission analysis and three engineers were in charge of the power subsystem design. Several possible solutions were analyzed and studied to relate the orbit properties with aspects as image capturing quality and ground station contacts. As the design progress (after several design sprints), several valid solutions were achieved. At the end of the session, since a large amount of resources were assigned to the prioritized subsystems (especially mission analysis), they achieved a higher level of definition than in the case of *Group 1*, although, on the contrary, some subsystems remained undefined (Table 2).

5.1.2 CDF session, Day 2

At the beginning of day two, *Group 1* continued with the design of all subsystems from their previous state. The design study and information data of previous session was shared and actualized in the form of design parameters. This actualization usually causes that the number of possible design solutions at subsystem level is reduced considerably. As all the subsystems are further defined, more information is shared and cross checked, so that the intersections between the solutions of the different subsystems rapidly decline. After two iterations, and at the end of the day, a first design option meeting the requirements was achieved.

Henceforth, the design was frozen and the process restarted to refine the design and to study another design solutions, to minimize the budgets of the mission (in terms of mass, power, and cost).

Group 2 started the day with the definition of the new sprint priorities, including new ones added from the requirements tree. Therefore, a re-allocation of resources was needed to start the design of new subsystems. While part of the engineers continued with the iterations on the subsystems already defined, both to achieve a better solution and evaluate the inherent risks, or to find and study new design solutions; others moved to the new subsystems, as shown in Table 3.

It should be noted that a fraction of the engineers assigned to one subsystem are not experts in that field, but they are experts in some of the subsequent subsystems, which come later in the design sequence (Fig. 6). This approach has proven to be effective to provide different points of view in the design process. Indeed, it is a common practice in the aerospace sector to create multidisciplinary groups of experts that give advice to the design team in dedicated meetings. The proposal here is to include this multidisciplinary character in the design process itself (not only through meetings), increasing the agility in the decision-making process.

Throughout the day, and once convergence in the solution was achieved, human resources were transferred from one subsystem to another, preferably between the ones with a direct relationship (see in Table 3 the increments of participants in each team).

5.1.3 CDF session, final day

On the last day of the sessions, *Group 1* was able to select an optimal solution for the complete preliminary design among all the design options studied and refined. This

Table 3 Subsystems definition and prioritization for the two CDF sessions conducted at the second day

Subsystem	Group 1		Group 2		
	Teams Size	Prioritization	Team Size	People Increment	Prioritization
Mission analysis	2	None	2	−6	A
Power	2	None	3	0	A
Communications	1	None	2	+2	B
Ground Station	1	None	0	0	C
Payload	1	None	2	−2	A
ADCS	2	None	2	+2	B
Propulsion	2	None	2	+2	B
Thermal	1	None	0	0	C
OBDH	1	None	0	0	C
Structure/Configuration	2	None	2	+2	B

Table 4 Subsystems definition and prioritization for the two CDF sessions conducted at the third day

Subsystem	Group 1		Group 2		
	Teams Size	Prioritization	Team Size	People Increment	Prioritization
Mission analysis	2	None	1	−1	A
Power	2	None	1	−2	A
Communications	1	None	1	−1	B
Ground Station	1	None	2	+2	C
Payload	1	None	0	−2	A
ADCS	2	None	1	−1	B
Propulsion	2	None	1	−1	B
Thermal	1	None	2	+2	C
OBDH	1	None	2	+2	C
Structure/Configuration	2	None	4	+2	B

Table 5 Comparison of final properties of *group 1* and *group 2*

Property	Units	Value	
		Group 1	Group 2
Volume		3U	2U
Mass	(kg)	4	2.7
Average power consumption	(W)	5.5	4.8
Lifetime	(months)	8	10
Mount Fuji access	(number)	100	120
Images number	(number)	54	61
Images quality	(m per pixel)	60	68
Orbit control and maintenance	(−)	No	Yes

design allowed to fix the mass, power and cost budgets (including margins).

On the other hand, *Group 2* reorganized the student group to complete all the subsystems design, as it is shown in Table 4, ending also the day with a preliminary design. The final design for both groups is presented in Table 5.

The preliminary design solution for *Group 1* was a 3U CubeSat of 4 kg mass and 5.5 W of average power consumption, and an estimated lifetime of 8 months. The solution achieved by *Group 2* was a 2U CubeSat of 2.7 kg mass, 4.8 W of average power consumption and 10 months of estimated lifetime. Differences between both designs are substantial in global terms (size, mass, power, and lifetime) because the level of refinement was higher for *Group 2*. In addition to the global differences in the design, the fulfilment of the initial requirements was also different for both groups. The design proposed by *Group 1* was able to complete a total of 100 passes over Mount Fuji, with a total number of 54 images, due to light conditions (imaging system can only operate in sunlight conditions). As the software simulations demonstrate, these images were obtained with a quality of around 60 m per pixel (with a 600×400 of pixel resolution). The capacity of the communications subsystem designed allowed downloading all the images in each contact (in terms of data rate), and the on-board computer had enough memory capacity for the storage of the acquired data in case a maximum of three failed contacts occur. *Group 2*

was able to improve the results of *Group 1* in terms of orbit stabilization. Due to the high level of prioritization in the mission analysis discipline (see Table 2), a detailed analysis of the decrease ratio of orbit altitude was performed at the first stage of the design (day 1). As conclusion, *Group 2* ADCS and propulsion disciplines start their design on day 2 with an additional input for orbit control and maintenance. This was not the case for *Group 1*, so no orbit maintenance was defined at this stage. Therefore, the design of *Group 2* was able to pass over Mount Fuji 120 times, increasing the number of images to 61 until mission finished due to orbital decay. The imaging quality was also improved up to 68 m per pixel. Like *Group 1*, the capacity of the communications subsystem allowed downloading all images in each contact (data rate). However, the on-board computer memory capability was reduced (to reduce its mass and size), been able to store the data of a maximum of two failed contacts.

5.2 Discussion

First of all, it was noted that the level of optimization reached by *Group 2* was higher than the one obtained by *Group 1*. This fact was not only observed by the results, but by the development process itself, since the authors took on the role of System Engineers (for *Group 1* and System SCRUM Manager (for *Group 2*) and reviewed the whole design process. On the one hand, the level of fulfillment of the requirements was different. By obtaining a longer operating time and a better resolution, *Group 2* managed to improve the operation of the S/C. In addition, by obtaining a smaller S/C, a reduction was also achieved in terms of mass and consumption, which results in an important decrease in the total cost of the mission.

Beyond the solutions achieved that, in fact, could be affected by external factors, an important aspect is extracted from the study about the applied methodologies. As is usual in ICE, the optimization phase occurs once a valid design is achieved [55], by means of successive iterations throughout the design. However, by the method proposed here, the optimization phase is embedded in the design phase itself. Indeed, as more resources were assigned to mission analysis and payload, a higher number of possible solutions were studied (including some time for the finding of innovative solutions), as well as the inherent risks of all of them. Then, the final decision results more robust and a database of different solutions can be created for subsequent stages.

In this way, characteristics of methods as SBCE are recovered [31–33], improving them with a greater allocation of resources, which generates that different subgroups have the possibility of studying different design options.

These improvements are mainly produced by the two main concepts that have been achieved by combining an *agile* methodology such as Scrum with CE: the task

prioritization and the re-allocation of resources during the design process.

In the space sector, an inherent prioritization of tasks is always performed since the beginning of the design process: the requirements are classified, as has been aforementioned, into high (mission) and low (subsystem)-level requirements [55–58]. With the proposed methodology, the prioritization is embedded into the design process itself, so that more resources are allocated for the most critical actions, which often means being able to study more design solutions.

However, for prioritization to be able to obtain its maximum potential in this new methodology, the second mentioned improvement must be implemented: the re-allocation of resources. As already mentioned, from the first moment in the design, the prioritization of requirements means that not all the subsystems start the design at once. This is beneficial in the first place, since many of these subsystems require that some others have a certain level of development, in order to obtain a necessary minimum number of inputs for their definition [15, 18, 55]. However, it is important to note that the criteria followed for this re-allocation is essential. The re-allocation should be performed between subsystems related to each other, so that the information already analysed can be useful for their design. Although the design responsible for each subsystem has to be an expert in this field, other designers are able to give their point of view (as experts of other disciplines directly related). For example, an engineer in charge of defining the communications or power subsystem should already have knowledge of the orbit definition for his own subsystem. Including him/her in the mission analysis design generates a different perspective away from the main objectives, since his/her decision criteria are based on his own subsystem. Nevertheless, this implies that there must also be a number of experts in each of the subsystems to be defined, which will depend on the size of the team itself.

6 Conclusions

Space missions have evolved from big missions with high development times and cost to small missions with more specific objectives (reducing size and cost). This tendency confirms the space mission design discipline as a dynamic process, in terms of adaptability and flexibility to the use of new technologies. Therefore, smaller development times are required and, as a consequence, an optimization of tasks is needed. However, in spite of this change of tendency, space mission design is still settled in the traditional concept of huge missions with high development times and cost.

CE applied to space sector emerged two decades over the past as a method to adapt space mission design to the *new space* paradigm. However, nowadays CE is not yet the usual design technique for most space missions and projects.

The method has remained static from an innovation point of view, still maintaining a traditional approach in their basis in spite of its potential. Other areas, as software development, have evolved in the same manner in terms of their environment, but an adaptation has been carried out affecting not only to the project concept itself but also to the design processes.

This work takes some of these new techniques (called *Agile* methodologies) used for software design discipline, but adapting them to be compatible with CE and, therefore, applied to space mission design. In particular, this work adapts the managing concept used in *Agile* methods to the CE usual scheme.

Both, CE and *Agile* methodologies have revealed as effective in their own application fields, as have been proved by their implementation in a significant number of companies and projects. As the method is based on these two design concepts, their basis and foundations are solid and well established.

The application of the new methodology shows an optimization in the development of the design, in terms of a better distribution of resources and a better structuring of the design phases. It allows, in the same development time, to achieve a more optimal solution, since waiting times associated with the lack of definition in certain fundamental subsystems are avoided. This improvement is accomplished due to the continuous optimization of the design during the whole preliminary design process.

Using the method proposed here, and as a result of the prioritization of requirements, the combination of CE and *Agile* methodologies in an application case within a CDF achieved an optimal pre-design for each subsystem at the end of each iteration (or sprint, if *Agile* nomenclature is used). As a result, no refinement of the design was required at the end of phase 0/A analysis.

On the contrary, for the application case using traditional CE, the iteration ends with a solution that does not necessarily correspond with an optimum. Indeed, as a result of the parallelization of tasks in the design, subsequent refinements and trend-off analysis must be performed by means of successive iterations of the overall design. As each iteration involves changes in at least one parameter (and this could, in turn, cause a chain of changes in several input or output parameters), a new loop in the design is required to ensure the convergence of the design solution.

During the design sessions carried out combining CE and *Agile* method, multiple small optimizations were made at the subsystem level, since several groups worked on them looking for different solutions. In this way, a greater optimization is achieved from early stages of the mission preliminary design. In addition, the waste of resources that usually occurs in CE process for subsystems that require a considerably high level of detail in their inputs, is minimized.

The higher optimization level from the CE plus *Agile* session is also caused by the flow of information associated with the project principal resource, the engineers. For instance, an engineer who defined the mission analysis will know perfectly its status and the main parameters that affects it. When this engineer is transferred to the communications subsystem design, he will have a greater vision of the properties of this subsystem, mainly affected by the mission analysis (orbital period, height, orbit type,...). If another engineer coming from the payload definition joins this group, the communication team will know first-hand the main inputs that affects the subsystem design.

Usual CE development times for long-term projects are of a few weeks instead of 6-9 months for sequential (or traditional) design [18]. The application of the method proposed in this paper could, in turn, shorten even more the process. Indeed, the main advantages of the application of *Agile* method for the Concurrent Design of space systems (a substantial time reduction and a high level of optimization at the first stages of the preliminary design phase) and their effects should be amplified if they are applied to (standard) long projects rather than a short session for preliminary design.

Acknowledgements Authors are indebted to the students of the Master on Space System, of the Technical University of Madrid (IDR/UPM), for their enthusiasm and hard work. Authors are also truly grateful for the support of the IDR/UPM staff during the development of this work.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Pacros A, Minster O (2007) The ESA contribution to space research on two-phase systems. *Microgravity Sci Technol* 19(3):9
2. Reuter K, Oberlechner J (2004) The ESA history project. *ESA Bull* 119:48

3. Jones M, Gomez E, Mantineo A, Mortensen U (2002) Introducing ECSS software-engineering standards within ESA. *ESA Bull* 11:132–139
4. ESA/ESTEC Requirements and Standards Division (2017) ECSS-M-ST-10C Space project management. In: Project planning and implementation, Noordwijk, The Netherlands
5. ESA/ESTEC Requirements and Standards Division (2017) ECSS-E-ST-10C space engineering. In: System engineering, Noordwijk, The Netherlands
6. Loureiro G, Panades W, Silva A (2018) Lessons learned in 20 years of application of Systems Concurrent Engineering to space products. *Acta Astronaut* 151:44
7. Hoffman M, Beaumont T (1997) Application development: managing the project life cycle. Mc Press, Boise
8. Walden D, Roedler G, Forsberg K, Hamelin R, Shortell TM (2015) Systems engineering handbook: a guide for system life cycle processes and activities. Wiley, New York
9. Eickhoff J (2009) Simulating spacecraft systems. Springer Science & Business Media, New York
10. Ridolfi G, Mooij E, Cardile D, Corpino S, Ferrari G (2012) A methodology for system-of-systems design in support of the engineering team. *Acta Astronaut* 73:88
11. Leaney GLP (2003) A systems and concurrent engineering framework for the integrated development of space products. *Acta Astronaut* 53(12):945
12. Creasey R, Whitcomb G (1999) Proceedings of the Institution of Mechanical Engineers, part G. *J Aeronaut Eng* 213:201–212
13. Capova K (2016) The New Space Age in the making: Emergence of exo-mining, exo-burials and exo-marketing. *Int J Astrobiol* 15(4):307
14. Casani E, Metzger R (1995) Reengineering the project design process. *Acta Astronaut* 35:681
15. Wall S (2000) Use of concurrent engineering in space mission design. In: Proceedings of European Systems Engineering conference, vol 1144, pp 1–6
16. Braukhane A, Quantius D (2011) Interactions in space systems design within a Concurrent Engineering facility. In: The IEEE international conference on collaboration technologies and systems (CTS), pp 1–8
17. Ye B, Salustri F (2003) Simultaneous tolerance synthesis for manufacturing and quality. *Res Eng Des* 14(2):98
18. Bandecchi M, Melton B, Ongaro F (1999) Concurrent engineering applied to space mission assessment and design. *ESA Bulletin* 99:34–40.
19. Bandecchi M, Melton B, Gardini B, Ongaro F (2000) The ESA/ESTEC concurrent design facility. In: Proceedings of EuSEC, vol 9, pp 1–8
20. Karpati G, Martin J, Steiner M, Reinhardt K (2002) The integrated mission design center (IMDC) at NASA goddard space flight center, vol 8–3657, pp 1–11
21. Smith J (1998) Concurrent engineering in the jet propulsion laboratory project design center. In: SAE transactions, pp 1106–1118
22. Bermejo J, Álvarez J, Arcenillas P, Roibás-Millán E, Cubas J, Pindado S (2018) CDF as a tool for space engineering master's student collaboration and concurrent design learning. In: Eighth international systems and Concurrent Engineering for space applications conference, pp 1–5
23. Ivanov A, Masson L, Belloni F (2016) Operation of a concurrent design facility for university projects. In: IEEE Aerospace Conference, pp 1–9
24. Guo J, Matthysen A, Fijneman M (2014) Concurrent Engineering in student course Practice Space Systems Engineering. In: Sixth international conference on systems and Concurrent Engineering for space applications, pp 8–10
25. Roibás-Millán E, Sorribes-Palmer F, Chimeno-Manguán M (2018) The MEOW lunar project for education and science based on concurrent engineering approach. *Acta Astronaut* 148:111
26. Manifesto A (2001) The Agile manifesto. www.agilemanifesto.org
27. Reich Y, Konda S, Subrahmanian E, Cunningham D, Dutoit A, Patrick R, Thomas M, Westerberg AW et al (1999) Building agility for developing agile design information systems. *Res Eng Des* 11(2):67
28. Parsaei HR, Sullivan WG (2012) Concurrent engineering: contemporary issues and modern design tools. Springer Science & Business Media, New York
29. Smith RP (1997) The historical roots of concurrent engineering fundamentals. *IEEE Trans Eng Manag* 44(1):67
30. Hauptman O, Hirji KK (1996) The influence of process concurrency on project outcomes in product development: an empirical study of cross-functional teams. *IEEE Trans Eng Manag* 43(2):153
31. Sobek DK II, Ward AC, Liker JK (1999) Toyota's principles of set-based concurrent engineering. *MIT Sloan Manag Rev* 40(2):67
32. Sobek D (1998) Principles that shape product development systems: a Toyota-Chrysler comparison. *Sloan Manag Rev*
33. Kerga E, Rossi M, Taisch M, Terzi S (2014) A serious game for introducing set-based concurrent engineering in industrial practices. *Concurr Eng* 22(4):333
34. Tamaskar S, Neema K, DeLaurentis D (2014) Framework for measuring complexity of aerospace systems. *Res Eng Des* 25(2):125
35. Avnet M, Weige A (2010) An application of the design structure matrix to integrated concurrent engineering. *Acta Astronaut* 66(5):937
36. Loureiro G, Leaney P (2003) A systems and concurrent engineering framework for the integrated development of space products. *Acta Astronaut* 53(12):945
37. Crabtree RA, Fox MS, Baid NK (1997) Case studies of coordination activities and problems in collaborative design. *Res Eng Des* 9(2):70
38. Jabí W, Soe S, Theobald P, Aish R, Lannon S (2017) Enhancing parametric design through non-manifold topology. *Des Stud* 52:96
39. Brown NC (2020) Design performance and designer preference in an interactive, data-driven conceptual building design scenario. *Des Stud* 68:1–33
40. Molina A, Al-Ashaab AH, Ellis TI, Young RI, Bell R (1995) A review of computer-aided simultaneous engineering systems. *Res Eng Des* 7(1):38
41. Biçici B, Cangeliç C (2012) Collaborative digital data management for design and production. *Proc CIRP* 3:585
42. Braukhane A (2015) Evolution of the European concurrent design tool for space-based system-of-systems studies. In: AIAA SPACE 2015 conference and exposition, p 4438
43. Komus M, Ayelty K (2020) Status quo agile 2019/2020. In: 4th international survey benefits and challenges of (scaled) agile approaches
44. Livermore JA (2007) Factors that impact implementing an agile software development methodology. In: Proceedings 2007 IEEE SoutheastCon (IEEE), pp 82–86
45. Schwaber K, Beedle M (2002) Agile software development with Scrum, vol 1. Prentice Hall, Upper Saddle River
46. Beck K (2000) Extreme programming explained: embrace change. Addison-Wesley Professional, Boston
47. Highsmith J (2013) Adaptive software development: a collaborative approach to managing complex systems. Addison-Wesley, New York
48. Abrahamsson P, Warsta J, Siponen MT, Ronkainen J (2003) New directions on agile methods: a comparative analysis. In:

- Proceedings of 25th international conference on software engineering, 2003 (IEEE), pp 244–254
49. Cockburn A (2004) Crystal clear: a human-powered methodology for small teams. In: A human-powered methodology for small teams. Pearson Education, Chennai
 50. Atzberger A, Paetzold K (2019) Current challenges of agile hardware development: what are still the pain points nowadays?. In: Proceedings of the Design Society: international conference on Engineering Design, vol 1. Cambridge University Press, Cambridge, pp 2209–2218
 51. Ovesen N, Dowlen C (2012) The challenges of becoming agile—experiences from new product development in industry and design education. In: International conference on engineering and product design education, London South Bank University
 52. García A, Torres B, Roibás-Millán E, Torralbo I, Cubas J, Sorribes F, Piqueras J, Alonso G, Sanz-Andrés A, Pérez-Grande I, Franchini S, Chamorro A, Gonzalez M (2016) Conceptual design of the UNION Lian-Hé University satellite using IDR/UPM CDF. In: Seventh international conference on systems and concurrent engineering for space applications, SECESA, pp 1–8
 53. Pindado S, Sanz A, Franchini S, Pérez-Grande I, Alonso G, Pérez J, Sorribes F, Cubas J, Roibás E, Fernández A (2016) MUSE (Master in Space Systems), an Advanced Master's Degree in Space Engineering. In: ATINER's COnference Paper Series, vol ENGEDU2016-1953, pp 1–16
 54. Roibás-Millán E, Sorribes-Palmer F, Cubas J, Pindado S, Chimento M, Alonso G, Sanz-Andrés A, Pérez-Álvarez J, Franchini S, Pérez-Grande I (2016) Implementation of Concurrent Engineering approach in MUSE (Master in Space Systems) Master's Degree in Space Engineering. In: ATINER's Conference Paper Series, vol ENGEDU2018-2537, pp 1–19
 55. Knoll D, Golkar A (2018) A coordination method for concurrent design and a collaboration tool for parametric system models. *Concurr Eng* 26(1):5
 56. ESA/ESTEC Requirements and Standards Division (2017) ECSS-E-ST-10-06C Space engineering. System engineering. Technical requirement specification, Noordwijk, The Netherlands
 57. Asundi SA, Fitz-Coy NG (2013) CubeSat mission design based on a systems engineering approach. In: 2013 IEEE Aerospace Conference (IEEE), pp 1–9
 58. Viscio MA, Viola N, Fusaro R, Basso V (2015) Methodology for requirements definition of complex space missions and systems. *Acta Astronaut* 114:79

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.