17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 608

An Agile Mind-set for Mission Operations in Commercial Space

Edoardo Cocci[a], Colleen Boggs[b], Mark Doyle[c]

[a] *Telespazio Germany GmbH, Europaplatz 5, 64293 Darmstadt, Germany*, edoardo.cocci@telespazio.de
[b] *Telespazio Germany GmbH, Europaplatz 5, 64293 Darmstadt, Germany*
[c] *Telespazio Germany GmbH, Europaplatz 5, 64293 Darmstadt, Germany*, mark.doyle@telespazio.de

Abstract

The commercial space industry can no longer operate in a linear way. We can't efficiently get from point A to B if we must first list all conceivable requirements, carefully map our route and set out on the journey fully equipped. For many commercial missions, point B is a moving target - to reach it, we grab the essentials and move out incrementally, responding to shifting dynamics along the way. Traditional project management methods, such as the predictive Waterfall model, have very limited flexibility against changing requirements, while delays are commonplace due to uncertainties and unforeseen complications. By contrast, adaptive and iterative methodologies such as Agile are well suited for environments where uncertainties reign free and requirements are anything but well defined at the beginning of the project. In this paper, we will outline how Telespazio Germany is applying Agile methodologies for the development of mission operation concepts and ground segment architectures, learning from the experience matured with the agile development of software systems. The iterative approach allows operators to have a usable product at an early stage of the mission design, which incrementally evolves into a solution that satisfies all operational needs. A shift towards an agile mindset and processes can positively influence commercial space projects, minimizing the risk of budget overruns or schedule delays.

Keywords: Agile, fail-fast, Scrum, iterative, complexity, commercial

Acronyms:

CDR	= Critical Design Review
CI/CD	= Continuous Integration and Continuous Deployment
CONOPS	= Concept of Operations
GSaaS	= Ground Station-as-a-Service
IOD	= In-Orbit Demonstration
LEOP	= Launch and Early Operations Phase
MCS	= Mission Control System
MOC	= Mission Operations Concept
UX	= User Experience

1. Introduction

Space missions are not exactly getting easier. Apart from the extensive time and money required, the complexity of space projects stems from the various elements and interdependencies involved between spacecraft, launcher, payload, ground stations, mission control, and stakeholders. Previously, only solidly funded organisations were able to handle the weight of such dynamics, and they did so using traditional methods. However, as technologies advance and space accessibility increases for commercial activity, institutional approaches have given way to competitive strategies. Today's space projects typically involve multiple teams/providers, tight timelines, restricted budgets and uncertainties. Complexity management, therefore, becomes a significant factor in a commercial entity's ability to thrive in the space industry.

The need for increased design flexibility, to align with continuously evolving commercial market demands, drives companies away from sequential approaches. Friction and inefficiencies mount in traditionally managed complex projects where technology and mission requirements have many unknowns. A helpful way to visualize the uncertainty and risk associated with a project is through the Stacey Diagram, also known as the matrix of uncertainty [1]. Commercial space missions are moving towards shorter schedules with dynamic requirements. These factors push most commercial space missions, per the diagram, into the complex area.

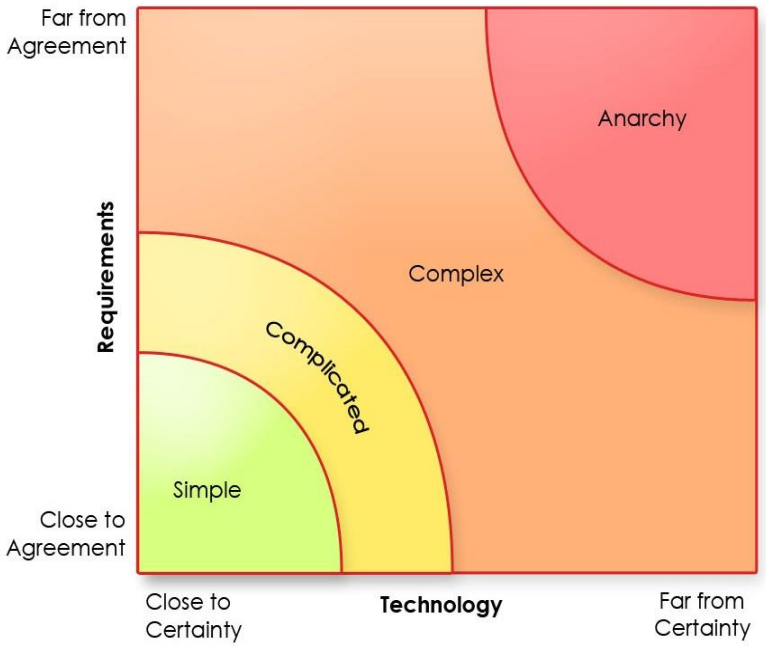

Fig. 1. Stacey Diagram

From the design and testing of systems to their operation, digitalization is a primary means of reducing complexity. Digital tools enhance our ability to collect information, communicate, coordinate, analyse, and automate. In short, they enable us to be more responsive; the same thing we need to battle complexity in space mission project management as well. Several companies have successfully developed new technologies in the highly dynamic and uncertain domain of commercial space using a responsive approach. Most notably, SpaceX and Planet have innovated using an iterative, fail-fast strategy in the launch segment and space segment, respectively [2]. Mission operations have begun to evolve as well with advances in ground segment systems, an area in which Telespazio Germany has led initiatives to enhance responsivity through improved complexity management.

The aim of this paper is therefore to add momentum to the shifting approach of space project management and provide recommendations for effective application based on the experiences of Telespazio Germany. Section 2 provides an overview of the main methodologies used, highlighting benefits and challenges. The subsequent section will offer practical guidance and lessons learnt from Telespazio Germany's experience employing a responsive approach to develop and deliver end-to-end mission operations systems and services aligning to the needs of the commercial space industry. Finally, the key take-away points will be summarized with a dash of perspective.

2. Waterfall vs Agile Methodology

Traditional, plan-based project management methods in the space industry, such as the Waterfall method, have enabled large teams to move progressively toward a common goal that was well defined in the early development stages. This method functionally arranges tasks, breaks them down for simplicity of understanding, and maintains control based on a clearly defined schedule. It is focused on predefined schedules and milestones aimed at guaranteeing the timely delivery and execution of the space mission.

The Waterfall method is easily understood. In expensive and high-risk space projects, carefully considering all mission aspects before any development work began was considered the prudent approach. Throughout much of the past decades, the most pervasive use of the Waterfall methodology in the space industry could be seen at NASA, where mission critical projects have little tolerance for failure and a high emphasis on mission schedule. Their wealth of mission history gave strength to their understanding of requirements, and with minor technology changes to subsequent missions, the process fit fairly well.

It became difficult, however, for new technology or operations processes to get implemented. The Waterfall methodology's sequential approach is not very adaptive; design changes are made based on results rather than user feedback and issues encountered can return the process to the start. Therefore, the cost of change increases rapidly as the project progresses.

The heavy amount of analysis involved in the early stages of mission development can lead to a bloated list of requirements and extensive documentation. Process overlap is generally avoided, causing inefficiency in work execution and longer mission lead times. Cross-team exchange can suffer as teams are more compartmentalized. In practice, it often generates work silos, compounding project extensions and an excess of time spent deliberating topics due to misalignment rather than taking progressive action. Thus, the Waterfall method can produce a false sense of security wherein the strength of a comprehensive structure and theoretical approach can deteriorate under the reality of engineering, schedule or budget issues.

In comparison, the Agile methodology is essentially focused on responding to change. While change is inevitable in life, not all projects demand this kind of focus (construction of buildings or bridges, for example). In the commercial space age however, this is central to technological advancement. When the Agile manifesto for software development was defined in 2001, it listed the following values:

- Individuals and interactions over process and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan [3]

Having blossomed since then into more than just a software development strategy, Agile has become an approach to organize, learn, and rapidly develop all kinds of products and services. It is customer focused and oriented to manage unpredictable changes in both design and operation. Agile uses iterative learning cycles to continuously improve. Projects are able to pivot and change rapidly, complex technical hurdles are iteratively overcome, and customers gain transparency and influence throughout delivery. It is therefore well suited to handle commercial projects operating at the frontier of space.

Not all are convinced of the suitability of the Agile approach for space projects, however. Some critiques of the method claim reduced quality and safety. While particularly critical aspects of human spaceflight missions, they are less incoherent with the fail-fast approach of unmanned spacecraft and satellites. As a significant mind-set leap from the Waterfall approach, Agile appears to some as a headless process with no clear plan and open-ended scope. Indeed, if executed poorly, Agile projects risk losing focus on goals. A number of issues can present if teams do not fully understand or adopt the process, which can be a steep learning curve. A lack of customer engagement and collaborative friction can also undermine the benefits [4]. The effective application of Agile principles is essential to manage these challenges. In our experience however, the benefit of the approach on space projects far outweighs the implementation effort needed.

3. The Application of Agile

Each space mission has its peculiarities, but at the same time, it is crucial to avoid reinventing the wheel where possible to save resources and time. At Telespazio Germany, we provide operations systems and services that can answer the needs of different operators, with missions that cover a broad spectrum of applications. With the commercialization of space, we recognized that the context and way in which missions were architected began to change. To deliver our services, we needed to experiment with alternative and more flexible approaches to design and provide solutions to the commercial space industry.

Traditional sequential development processes had proven very effective where requirements did not change significantly over time, meaning that the "what" and the "how" were frozen once the design concept consolidated. However, such processes were obviously inappropriate for the commercial space missions in which we were involved. Namely, we saw primarily that:

- Requirements were not very clear or incomplete at the beginning of the project
- Requirements were prone to change throughout the mission
- Launch times were strict

Shorter timelines compared to traditional, institutional scientific missions and evolutive business concepts focused on growth and adaptation in early project phases were the main reasons for requirement instability. In addition, the ground segment and operations were still often an afterthought in commercial space, so the associated requirements were often yet to be consolidated at kick-off.

The above-mentioned challenges pushed forward the need for an adaptable and responsive approach to be applied in the overall development process of our solutions. Taking inspiration from the experience matured within our software department, where the Scrum framework had been used as a reference to cultivating our own Agile practices, we gradually implemented Agile methods into the broader context of mission operations services, namely in our Operations-as-a-Service offering. The portfolio of services we developed included not only software systems but also end-to-end in-orbit operations services, thus the people, interfaces, procedures, and processes involved in the successful execution of a space mission.

Agile methods provide the framework around which each enterprise, with its challenges, can progressively build its own practices rather than prescribe a detailed structure and processes that risk augmenting complexity. In the same spirit, this paper aims to serve as an inspiration for similar use cases. The following sections will discuss the Agile practices and lessons learnt that Telespazio Germany has been implementing throughout the entire lifecycle of commercial missions to alleviate the implementation challenges mentioned above. Our examples are oriented towards small satellite constellation missions, though principles can translate for a variety of other scenarios.

3.1 Participatory Design

From project start, the Agile approach demonstrates its flexibility and customer-oriented nature. Our team can begin solution development through iterative exchanges with minimal mission requirements. There is no heavy burden on the customer side to produce significant packs of documentation to which the provider must ingest and evaluate; developing the mission guidelines and objectives becomes more of a collaborative process. Customers are actively involved in realizing a project and its features, and empowered to shift direction when business needs demand it.

In operations, the Mission Operations Concept (MOC) is a fundamental piece of the mission puzzle; it defines how to execute a mission to accomplish its objectives. Covering the operations of both the space and ground segments, it is essential in defining the mission system architecture and requirements for the various system elements. To combat the need for clarity in its development, we adopted a strategy based on an agnostic baseline solution to be iteratively refined with the customer to make sure it would meet their specific needs.

Table 1. Solution Design Strategy

At project kick-off	Establish a generic operational concept specifically designed for operations of large constellations of small satellites, with automation and scalability as the driving principles. Processes and resources are parametrised based on inputs such as the constellation's size and the space segment's complexity.
After project kick-off	The project enters a “Solution Shaping” phase, where baseline concepts are iteratively tested against the customers' needs and adapted to them. This is achieved through a series of workshops where all possible operational scenarios are investigated and consolidated.

This exercise allows refining “what” the customer requires and which are the service levels. These needs then flow down naturally to the various elements of the system, such as ground segment software components. As described by Carpenter et. al. [5], Agile methods are well suited for knowledge discovery, i.e., for environments where the “what to do” is unknown at first. In contrast, traditional methodologies and standards should be kept in those development contexts characterized by codified knowledge that is already known and repeatable. An Agile approach could be used to help the customer determine what communication protocol would fit best to mission architecture and needs, however, once the protocol is selected the “how to do it” part could be left to well established engineering standards for implementation.

A big weakness in Agile’s collaborative approach is in the availability and receptivity of the customer to participate in these exchanges. Thus, a strong understanding of the customer’s basic needs (directly stated or characterised from

market insights) and a baseline solution will prevent the project from being stalled at the start. Agile processes and teams feed off the momentum of continuous progress, so taking steps to kick-start the project will minimize stumbles and/or stagnation.

3.1.1 Software built 'for' the user and 'with' the user

Agile methods belong to a broader category of incremental and iterative approaches that share the fundamental concept of incorporating continuous or frequent feedback from stakeholders. When developing the systems to be used in the mission operations environment, the tendency is not to consider this a potential innovation area [6]. Since mission operations practices are generally regarded as well established, emphasis is rarely given to user experience (UX) aspects and how we assume operators will execute a mission.

The shift towards employing large constellations of small satellites to achieve objectives similar to a single large platform demands a parallel revolution in how satellites are operated. The same approach to mission operations cannot be blindly applied to missions that consist of hundreds, if not thousands, of satellites. In this endeavour, a leading role is played by software systems, both in the ground and space segment. They can leverage analytics, automation and Machine Learning to minimise the need for human intervention in an operational environment. These are the key elements necessary to materialise concepts such as night-lights-out operations in the space industry [7]. However, the human factor, and in this case, the satellite operator, must be at the centre of the design process and development of all the tools [8]. In our experience, we found it extremely valuable to make room for topics such as usability and ergonomics in developing our Mission Control System (MCS).

The goal is to maximise the level of customer involvement in the design, tailoring and development of the products in order to make sure that all the needs are answered and any deviations captured in a timely and efficient manner. The first best practice essential to achieve the above consists of concentrating all the initial efforts on extracting what the customers need versus what they seem to be asking. There is a very subtle difference between the two that has been discussed far and wide in any industry. In satellite operations, we have found it common to observe strong biases in how operators expect a system to behave or look based on past experience, whereas a more slick and innovative solution would perform better. Thus, it was crucial to learn what questions to ask to drill to the essence of how the customer envisions operations.

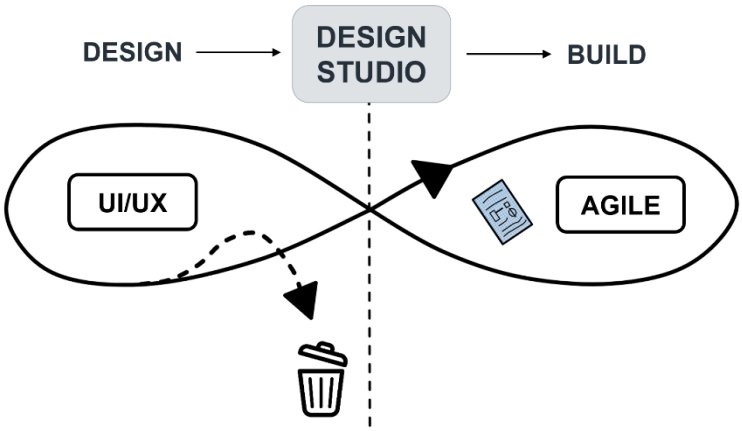

Fig. 2. From design to build

Second, in the systems development domain, we have established a standard workflow to effectively translate customer needs into meaningful products. The approach draws from the Lean UX philosophy that lets human factors and ergonomics carry on the design process focusing on rapid iterative cycles as done in Agile development. To ensure that we develop intelligent systems with high user-relatability and ease of use, we introduced UX Design Studios. They consist of workshops that bring together all the possible stakeholders, such as frontend, backend and test developers, and the final users, in this case, the satellite operators. In a very condensed manner, involving divergent and convergent thinking, these workshops allow exploring a wide array of different ideas that ultimately will enrich the final design solution. The overall process is usually time limited to 1.5 to 2 hours, and it consists of the steps outlined in Table 2.

Table 2. UX Design Studios Workshops

Individual session	Consisting of an initial creative part and time-boxed to 15 minutes, it is where each participant draws on paper 1 to 3 ideas of how they envisage the user problem to be solved. Then each participant has 3 minutes to show and describe his or her ideas to the group. The group has some time to comment and provide feedback.
Group session	Similar to the previous part, but in this case, the participants are assigned to different breakout groups of 2 to 3 people. Each group has then to converge internally and present a single idea. Each group has 3 minutes to show and describe the concepts to the other groups. The whole group has some time to comment and provide feedback.
Final convergence	Based on the ideas presented in the previous step, the whole group has to discuss and converge on a final design. The outcome of the process will be a solution enriched by the contributions of all participants.

A UX Design Studio requires participants to be equipped with just some paper, a thick black marker, great ideas and a spirit of collaboration. Because of the simplicity of setting them up, such workshops can also be held in a virtual environment, with the participants distributed in remote locations. Figure 3 illustrates the steps described above as carried out during one of the numerous design studios that we used to define, together with the customer, what was the best graphical interface to accomplish a specific function.

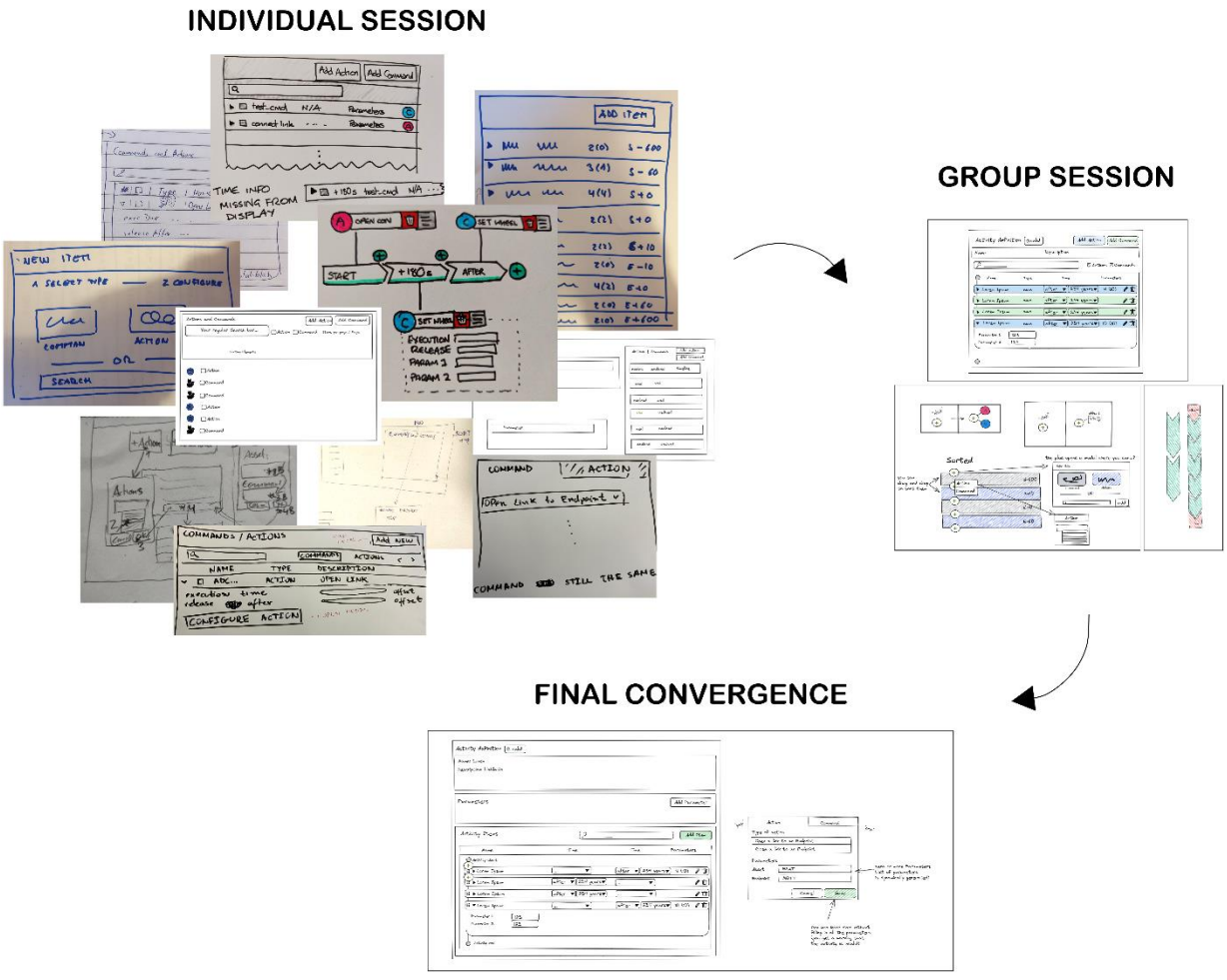

Fig. 3. UX Design Studio iterations example

The observed benefits of utilising such an approach are:

- *Fast problem solving* - Rapid prototyping allows to complete the conceptual design in a few hours, removing the need for lengthy exchanges between stakeholders that can also lead to fatal misunderstanding.
- *Cross-functional communication* - Bringing together users, developers, testers, and the customer, the improved collaboration between stakeholders promotes transparency and understanding about the solution.
- *Three brains are better than one* - The heterogeneity of backgrounds and areas of expertise of the participants foster a great diversity of ideas and thinking.

Including the customer, and thus the final user, in the creative part of the development augments customer satisfaction by allowing them to contribute to the design actively. In addition, this ensures that all their needs are addressed in the design and embodied in the final product.

3.2 Fit for Use

In most projects, change leads to delays and increased cost. As a project progresses, the impact of change on schedule and budget becomes greater. For example, in the traditional testing phase, the cost of corrective action is much higher than in the requirements definition stage. In projects that struggle to accept change, there is a risk of creating something unsuitable for the mission. The core principle of Agile, however, embraces change by accepting it as not only unavoidable or necessary, but useful.

As defined by Heeager and Nielsen [9], an iterative method is one where work is carried out in a cyclical manner; tasks are executed based upon project priorities. In an incremental process, system functionalities are delivered in a series of phases, rather than as one final delivery, with the goal of evolving the system based on customer and user feedback. Agile methods combine both iterative and incremental methodology, as they aim to deliver value constantly throughout the entire project while improving the solution at every iteration. Adopting an iterative-incremental approach during the implementation phase of a system and operational processes has brought us several benefits.

- *Adaptability to a continuously changing environment* - In the commercial world, the time from the contract signature to launch can be short, sometimes less than a year. This is a very short time to develop a mission. During this timeframe, we must build the ground system components without concrete interfaces, protocols, or even a complete CONOPS. This is only possible using an iterative and incremental method. If the project is unknown and changing, how is it possible to design everything up-front? The best we can do is make predictions, but predictions are not a solid plan. The iterative-incremental method which brought us success is Scrum. As defined by Schwaber et al. [10], the approach spreads the effort across a series of iterative cycles called sprints, allowing teams to respond to changes in the scope of work and stay on target. Teams work only on tasks that will bring value to the mission and therefore develop a system that is fit for use.
- *Empowered customers* - Customers have the chance to be tightly involved during the implementation phase. They also have the opportunity to participate in prioritisation activities and fine-tuning of the project direction [11]. In the race to get their product or service to market, having transparency in the project progress enables companies to re-prioritize and pivot as needed to meet their schedule and business needs.
- *Test early and often* - Another unique characteristic of the iterative nature of Agile methods is frequent testing and validation. Tests are fundamental to verify that a system was built according to the specifications and overall stakeholder needs. Haskin et al. [12] showed how the cost of corrective actions can increase drastically as a project advances, reaching values of 21 to 78 times higher when issues are identified only in the testing phase compared to what it would have cost to make the change during the requirements phase. Agile methodologies do not postpone testing activities, they are employed at end of each iteration cycle. This approach improves quality persistence and ensures that the project's pitfalls are identified as early as possible.

From our experience developing ground software systems, we have observed that iteration cycles of 2 weeks provide the right balance between having a cadence that is high enough to incorporate changes and having enough time to develop features that are more complex.

The iterative-incremental nature of Agile methods can also be of great help after the implementation phase. Indeed as operations are kicked off, and thus the execution phase commences, the wheel of iterative development and continuous improvement does not stop. Space operations are characterised by an evolving environment and

consequently operational tools and processes need to be adapted to guarantee safe and optimal operations. For this reason, an Agile mind-set can bring many benefits during the execution phase of a mission, it brings greater levels of flexibility and response speed when urgent changes or upgrades are necessary.

- Specific elements of the mission operations systems, such as automation, strongly rely on the knowledge of how a system behaves when operational. The foundation of an autonomous concept for monitoring and control of large constellations of satellites shall be available before the execution phase starts. However, the systems and the development methodology must be flexible and agile enough to build up automation capabilities incrementally as operations progress.
- Satellite operators become the users of continuously developed systems. The operators therefore can contribute ideas for feature improvements, critical upgrades or bug fixes to the backlog at any time. Then, if the priority is deemed high enough, the developer's team will pick the item in the following sprint iteration. Using the same approach as mentioned in Section 3.1, the developers and operators will concurrently work to design solutions that keep improving the system.

Success Story #1

To support a customer in their first In-Orbit Demonstration (IOD) mission, our team was requested to integrate the ground station owned by the customer into the MCS platform to allow them to use it to exchange telemetry and telecommands with their satellite once in orbit. However, during the Launch and Early Operations Phase (LEOP), the customer experienced difficulties identifying their satellite after separation. The performance of their antenna was the root cause of this. To speed up the process of connecting to their satellite, we decided to use the ground station network provided by an external Ground Station-as-a-Service (GSaaS). This meant that the telemetry and telecommand decoding/encoding chains of the MCS were required to be adapted to the new customer needs. Thanks to the flexibility enabled by the Agile working scheme, the team could reprioritise the backlog of activities being carried out and address the new pressing customer issue. The customer environment was ready to support the data connectivity with a GSaaS provider in less than a week. This was a great example of how the operational requirements of a mission can suddenly change and to what extent an agile mindset and organisation can support this.

3.3 End-to-End Functionality

Technological advancements in the launch segment have resulted in an increasing launch cadence, giving more opportunity to companies (particularly new players in the industry) to accelerate their time-to-orbit. The faster pace also means that the development lifecycle of operational systems and processes need to be able to answer operator needs in a shorter timeframe. This means the process we apply has to produce systems that are fit for the mission without any surprises. In order to satisfy this requirement, we have had success using the Scrum concepts of sprints and sprint planning. Our user stories are defined in terms of value for the user and follow the Scrum recommendation of having a definition of "ready", and a definition of "done". This process is carried out in collaboration with the client (which for some systems may be two internal teams e.g., operations and software) and the team developing the system, including the product owner. This gives all of our stakeholders a shared understanding of what is expected. As part of the sprints, we use Continuous Integration, Continuous Deployment (CI/CD) that forces us to deliver systems into operations once they pass testing. We automate testing as much as possible to provide confidence against regression and speed for deployment. Through effective prioritization, transparency and continuous testing and integration, Agile methods reinforce that by the launch date, the customer is equipped with what is needed to succeed.

- *Effective Prioritization* - Scrum employs a prioritized product backlog, which we use to track all tasks necessary for project completion. The important thing to note is that the backlog is a living document; its tasks will change and it will grow and shrink throughout the project. The product owner prioritizes the backlog with input from the entire team and other stakeholders, such as the customer. This collaboration fosters engagement through buy-in and makes sure the project is moving towards the most important feature. The prioritization mechanism is enacted at the beginning of each sprint, and considerations about dependencies and risk reduction are at the basis of the decisional process that determines what shall be addressed next. The ability to review the backlog with a high frequency allows reducing risk greatly, and making sure that core tasks are tackled promptly before it becomes too expensive. The overall process is also instrumental to guarantee a clear understanding of the project goal and direction.
- *Transparency* - As mentioned above, Agile ceremonies do not involve only the development team, but also any other relevant stakeholder. Customer feedback is encouraged at any step of the process to make sure that

it is incorporated in the product and services. Additionally, the backlog is available to the customer allowing them to keep an overview on the development progress and what tasks are upcoming. Transparency is a core tenant of Agile and one we have found to be very valuable. We can make decisions which reduce risk and provide a better product. All this has made fixed launch dates less of a worry, no longer do we have "integration" bubbles.

- *Continuous Testing and Integration* - Traditional methods might seem more rigorous in comparison to Agile with regards to safety and reliability requirements and thus better suited for certain space projects. However, making sure that each user story has well-defined verification and validation criteria is one of the many best practices that can answer such safety concerns. Agile processes include adopting continuous and iterative testing, while, in a typical Waterfall project, the verification is an end-loaded process [13]. This "test at the end" focus often uses test cases that have failed to evolve with changes in the project. In contrast, Agile methods help identify issues with requirements much earlier, for example, wrong, missing, or unrealistic requirements [14].

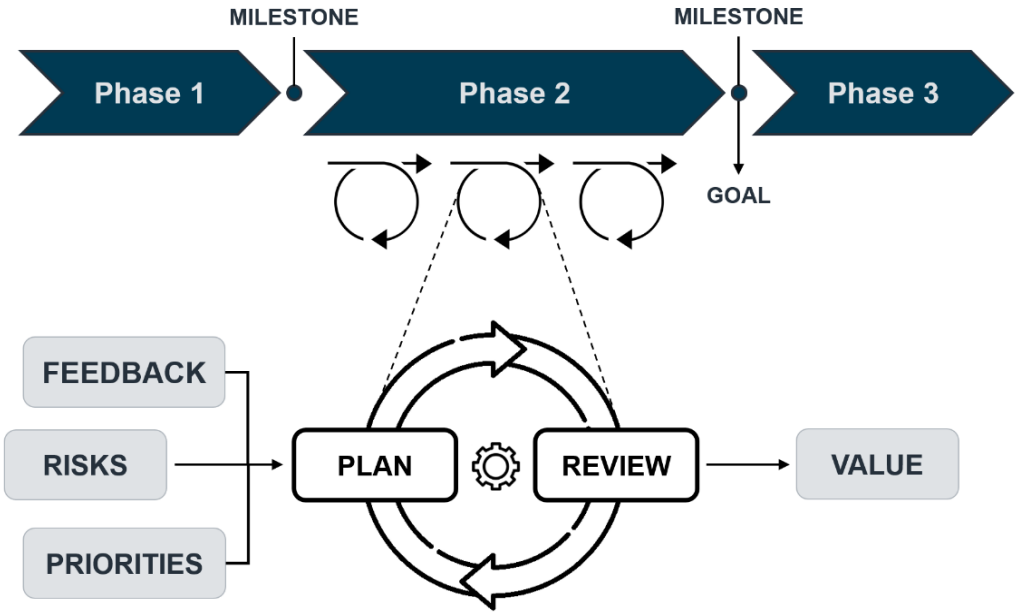

Fig. 4. Agile approach as part of a phased project

One of the most popular misconceptions of Agile methodologies is that they can conflict with deadlines, thus, are not suited for complex projects that traditionally are tackled through an ordered sequence of progress phases and milestones. However, Agile does not preclude the presence of deadlines, but it suggests a different way to get to the delivery date with a product that meets all customers' needs. A typical Waterfall model, or gate-phased process, outlines all the steps lying in between two milestones with an extraordinary level of granularity. While this is meant to mitigate risk, it only creates a false sense of security of solution progress, according to [15], forcing premature design decisions and "false-positive" feasibility. As illustrated in Figure 4, an Agile team would instead consider milestones as boundaries and focus on maximising the value delivery by measuring progress and incorporating customer feedback, iteration after iteration. In addition, a roadmap is established to help in the process of planning and prioritising activities, however, it shall not be intended as a fixed plan that must be followed to the letter. The roadmap provides the goals that drive the prioritisation of the backlog of activities, and it is likely to change and mutate throughout the lifecycle as the team starts implementing and learning more about how to meet the customer and business needs best.

Success Story #2

On another occasion, an operator requested our support developing a MCS that could operate their soon to be launched mission. The challenge was that the lead time until the launch of their satellite was less than one month. Rather than concentrating the effort in following a very elaborate and accelerated plan to the letter, the team shifted its focus towards a new goal: ensuring the customer would be in the position to successfully run its mission enabling

monitoring and control capabilities. Under the new shared roadmap vision, the team started working incrementally, together with the customer, to reach the launch with a fully functioning, tested and validated solution for their needs. In two 2-week sprints, the team was able to integrate the satellite's telemetry and telecommand protocols, the ground stations' interfaces and protocols and ultimately test and validate them with a flatsat. This demonstrated that a common vision and a shared goal in some cases can be more effective than a very detailed plan.

4. Conclusion

While mission complexity remains high, the commercialization of space demands new approaches to project management. Competitive strategies which lower cost, increase innovation and move more swiftly towards launch are better suited for the majority of commercial space missions. Focusing on customer needs, products and services that follow iterative development cycles offer greater flexibility, efficiency and user-centered design.

Agile methodologies are well oriented to this approach. They can help teams manage the complexity of space projects by providing a framework for breaking down tasks into smaller, manageable chunks, and prioritizing critical systems. It permits more rapid response to changing requirements or unexpected challenges, while maintaining a means to efficiently coordinate among a number of interdependent mission elements and stakeholders. Across mission planning, design, and testing, Agile offers a number of advantages over traditional sequential methods such as Waterfall.

The application of Agile methodologies is not without challenges, however, and safety-critical space projects may require a different approach. Therefore, organizations may adapt the methodology to fit the specific needs of their projects, rather than trying to fit the project into a standard Agile process. In Telespazio Germany's experience, the Agile method was particularly useful to address weak requirements, shifting needs, and shortened project timescales.

To leverage the strengths of Agile, several best practices are recommended. An understanding of essential needs and a baseline solution can prevent a sluggish start to the mission design phase, when the availability of a customer to iteratively build the solution is lacking. In other cases, where capturing the essence of user needs is the focus, workshops may be used to elicit ideas on individual and group levels to engage the customer in the creative process. In either case, the participatory design process aims at ensuring specific needs are well defined and incorporated into the product design.

Space missions and technologies are constantly evolving to improve space utilization, but when this happens in the course of a traditionally managed project, the cost of change grows exponentially higher with time. Agile, by contrast, embraces such changes. It empowers customers to fine-tune project direction throughout the process, and continuously verify alignment with frequent testing. In our experiences, commercial projects can encounter significant course corrections and therefore benefit greatly from the ability to reprioritize to maintain the product or service is fit for use.

Finally, under the pressure of tight deadlines, Agile can move much quicker to an operational phase than traditional project methods. Through greater transparency and engagement for the customer, Agile teams maximize value by adapting planning and reprioritizing activities. A roadmap helps by setting goals for backlog execution, a different approach to ensuring launch dates are met.

Agile methodologies are not one-size-fits-all. They can be adapted to the project and integrated with other project management methodologies to ensure that the project stays on schedule, within budget and delivers high-quality results. Unknowns intrinsically dominate the space environment. Pushing the boundaries of space exploration and utilization will keep bringing us to new uncharted territories where we need to have the right approach deal with it. NASA, for example, did not always follow Waterfall methods. During the Apollo program, teams iterated, discovered, and improvised much more than later programs. The teams tasked with finding the way to the Moon did so incrementally - gradually stepping up mission complexity, making mistakes and learning from them. The end goal was the most fixed aspect – to get to the Moon, quickly. This approach was less predictive than it was adaptive, and demonstrates that commercial projects are not the only ones to benefit from an Agile approach in the space industry.

References

[1] H. Bodner, Why Agile? – The Stacey Complexity Model, 17 February 2016, https://www.scrum-tips.com/agile/stacey-complexity-model/.

[2] B. Howard, What is Agile Aerospace? Learn Planet's Approach, 16 October 2019, https://www.planet.com/pulse/what-is-agile-aerospace-learn-planets-approach/#:~:text=Learn%20Planet%27s%20Approach,-Ben%20Howard%20%7C%20October&text=Agile%20aerospace%20is%20a%20philosophy,one%20on%20the%20first%20try..

[3] K. Beck, et al., Manifesto for Agile Software Development, 2001, http://agilemanifesto.org/, (accessed 20.01.23).

[4] G. J. Miller, Agile problems, challenges, & failures, PMI Global Congress - North America, New Orleans, USA, 2013, 29 October.

[5] S. E. Carpenter, A. Dagnino, Is Agile too fragile for space-based system engineering?, 10.1109/SMC-IT.2014.13, 2014 IEEE International Conference on Space Mission Challenges for Information Technology, MD, USA, 2014, 24 – 26 September.

[6] J. Trimble, M. Shirley, S. Hobart, Agile: From Software to Mission System, 10.2514/6.2016-2477, SpaceOps 2016 Conference, Daejeon, Korea, 2016, 16 – 20 May.

[7] D. Doan, R. Zimmerman, L. Leung, J. Mason, N. Parsons, K. Shahid, Commissioning the World's Largest Satellite Constellation, 31st Small Satellites Conference, Logan, USA, 2017, 5 – 10 August.

[8] M. Hollender, T. G. Grave, J. Partini, P. Scharing, Making sense of complexity, ABB Review (2018).

[9] L. T. Heeager, P. A. Nielsen, A conceptual model of agile software development in a safety-critical context: A systematic literature review, Information and Software Technology 103 (2018), 22–39.

[10] K. Schwaber, J. Sutherland, The Scrum Guide, November 2020, https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf.

[11] S. Kaur, S. Siarov, L. Lindblad, M. Witzmann, A practical guide to agile mission design and spacecraft development with data-driven systems engineering, Small Satellite Systems and Services Symposium, Portugal, 2021, March 3.

[12] B. Haskins, J. Stecklein, B. Dick, G. Moroney, R. Lovell, J. Dabney, Error Cost Escalation Through the Project Life Cycle, 14.1723-1737, INCOSE International Symposium, Toulouse, France, 2004, June 20 – 24.

[13] R. Rasmussen, T. Hughes, J. Jenks, J. Skach, Adopting agile in an FDA regulated environment, 10.1109/AGILE.2009.50, 2009 Agile Conference, Chicago, USA, 2009, August 24 – 28.

[14] J . P. Notander, P. Runeson, M. Höst, SimPal: a design study on a framework for flexible safety-critical software development, ACM SIGAPP Applied Computing Review 13 (2013), 17–29.

[15] D. P. Oosterwal, The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development, Amacom, 2010.