Preface

I'd like to express my sincerest gratitude to those who have shown in interest in my paper **"A practical guide to agile mission design and spacecraft development with data-driven systems engineering"**, presented during the European Space Agency's 9th International Systems & Concurrent Engineering for Space Applications Conference (SECESA) 2020.

I wrote this paper during the pandemic from Strasbourg, France, when I joined Valispace as a remote worker. At the time I noticed my peers and organizations in the industry facing technical and social difficulties in adjusting to the new remote working culture, which was contrary to my experience as I started working with Valispace. I discovered there were several factors why my transition to remote working felt natural:

1. Valispace is a software company with a **globally distributed team**; remote working has been embedded in our company since before the pandemic struck.
2. We use a modified **agile process** to avoid a meeting-heavy culture which leads to decision bottlenecks. Sprints enable us to develop solutions rapidly, while keeping our sights on the big picture.
3. We practice what we preach: our culture and software fosters instantaneous communication and feedback because we utilize the principles of **Data-Driven Systems Engineering (DDSE)** in our development, testing, and documentation cycles.

I researched hardware companies which were struggling or faltering to the work restrictions imposed by the COVID-19 pandemic. My subjects were not prepared for this new work culture and were falling behind in their development schedules or finding it difficult to remain productive. It turns out many did not have the resources, processes, or the interpersonal culture to rapidly transition to remote working without disruption.

The solutions I've proposed are designed to be a stepping stone for those who are moving operations partially or completely online, or trying to remain agile in this pandemic. While the five practical guidelines in this paper are geared towards the space hardware industry, I've generalized the principles so they may be modified and adopted by any organization or institution in the technology sector.

Thank you for your interest in this paper. Please contact me at supreet@valispace.com if you have any questions or comments.

Sincerely,
Supreet Kaur

A PRACTICAL GUIDE TO AGILE MISSION DESIGN AND SPACECRAFT DEVELOPMENT WITH DATA-DRIVEN SYSTEMS ENGINEERING

Supreet Kaur [1,2] Stefan Siarov [1], Louise Lindblad [1], Marco Witzmann [1]

(1) Valispace GmbH, Hermann Köhl-Str. 7, 28199 Bremen, Germany – stefan@valispace.com (2) International Space University, 1 Rue Jean-Dominique Cassini, 67400 Illkirch-Graffenstaden, France

ABSTRACT

Traditionally, complex space hardware and mission design has been a sequential document-driven process [1]. However, given the growing complexity of interdisciplinary design in the dynamic global space economy, the hardware design community is looking for methods to optimize workflow given the challenges and limitations of current tools and processes.

The "Introduction" provides some background on space mission and hardware design by identifying the overlap between mission lifecycle and space mission analysis and design. The following section, "Agile Aerospace", provides insight into the benefits of moving away from traditional sequential models and into concurrent and iterative Agile models in space design and development. An example of a successful Agile company and the rise of remote working are discussed. The "Practical Guidelines for Remote Teams in the Agile Space Industry" section provides practical guidelines to benefit from the advantages of agile methodologies, particularly following the Data-Driven Systems Engineering approach, in remote development.

INTRODUCTION

With the increasing complexity of space operations, companies and agencies in the space sector are seeking tools and methods for workflow and development optimization. This is comparable to what happened in the software industry [2,3], where similar problems had to be solved in the past decades. In an era of agile ways of working in the software industry, it is valuable to bring these approaches to hardware design and present the idea of agile space hardware and mission design following a Data-Driven Systems Engineering (DDSE) approach.

This section presents the Mission Lifecycle and the Space Mission Analysis and Design (SMAD) process, which are interlinked and the backbone in realizing space operations.

Mission Lifecycle

The National Aeronautics and Space Administration (NASA) defines Mission Lifecycle stages into discrete phases [4]. The cycle begins with Pre-Phase A, which evaluates a wide range of ideas and mission alternatives to develop initial mission concepts, identify key stakeholders, and define top level system requirements and ConOps. Phase A develops a baseline mission and proposes mission architecture that is both feasible and meets the mission's expectations, requirements, and constraints. In Phase B planning, technical, cost, and schedule of technical and business baselines are developed. This results in prototyping and assessments showing that the system and subsystem requirements, specifications, designs, and verification plans are compliant. Phase C focuses on detailed technology development to realize the final product, which includes unit and integration testing. In Phase D the components are assembled, integrated, verified, and validated, resulting in a system which meets requirements and is ready for operation. In Phase E the mission is executed, and concludes with decommissioning in Phase F.

Space Mission Analysis and Design

The overall mission design for space is realized by the Space Mission Analysis and Design (SMAD) process [5]. SMAD articulates the primary and secondary non quantitative objectives of the mission, as well as the quantitative requirements. The primary distinction in SMAD is the trading of requirements, mission elements, and system drivers to find a compromise between what is feasible and the desired outcome. The process starts by defining the objectives, the broad goals, of the mission as well as high level functional requirements, operational requirements, and constraints. Next, the mission is characterized by the mission concept, which includes mission elements such as mission timeline, funding, data handling, communication architecture, scheduling, and control. The mission elements are traded with the system drivers (performance, cost, risk, or schedule) in mind, as alternative mission architectures are explored. This results in the mission utility analysis, which quantifies mission performance as a function of system drivers, ultimately resulting in go/no-go decisions on proceeding with the mission, selection of mission concept, and detailed engineering decisions. This leads to the development, decomposition, and allocation of traceable system requirements into lower levels. While the SMAD process is iterative, it coincides with the Pre-Phase A to Phase B of the mission lifecycle (see Fig. 1).

Figure 1. Mission Lifecycle and Space Mission Analysis & Design

AGILE AEROSPACE

Traditional processes of development are linear and built incrementally, requiring the completion of one stage before moving onto the next [1]. Hence, the standard approach for space hardware development is rigid and document-driven. Consequently, a single delay in one of the stages can impact the entire critical path of the project or mission.

Whereas the Agile method is built for uncertainty and dynamic environments. Agile, which started as a means of management and development for software, has quickly grown to encompass large scale hardware projects in all industries. Given the unpredictable and volatile nature of the present-day customer-driven marketplace, the methodology tackles some the biggest downfalls of the other methods through these main benefits [6]:

1. **Agile leads to a shorter development cycle:** Unlike traditional processes, Agile projects are developed, tested, and managed in discrete units, during sprints, which last 1 - 4 weeks, depending on the team structure [15]. With incremental development occurring throughout an iterative approach, the team can create and deliver a minimal viable product (MVP) to the customer, a base working model, and implement revisions in a short period of time.
2. **Quick and flexible to change:** The shift to agile has resulted in a process that is more streamlined, automated, and enables concurrent engineering. Frequent builds of the model and qualification helps to find and fix defects and bugs quickly and continuously. Agile simply enables teams to thrive and produce by adapting to a continuously changing environment through early identification of defects, ensuring quality and adherence to budget and schedule.
3. **High level of customer involvement:** Another unique aspect of Agile is the high customer involvement throughout the project. This enables accommodation of unexpected changes and revisions even after development has started. Each iteration presents an opportunity to reprioritize product backlog and fine-tune the project's direction to the customer's needs. Agile permits both a dialogue and a negotiation between the engineers designing space hardware and the customers, on topics such as adding or modifying features at the end of every sprint.

The rapid response and stealth adoption of user feedback, combined with ongoing Verification and Validation through the process results in an end product which may be different to the initial idea, but more functional and closer to the customer's needs.

Data-Driven Systems Engineering for Concurrent Engineering

To experience the benefits of being Agile, team members must be able to work concurrently while having simultaneous ease of access to the central database which stores all of the relevant information related to the project [7]. Digital data management is key in the remote agile infrastructure, requiring a Data-Driven Systems Engineering (DDSE) approach.

The database must be consistent, relying on a single source of truth for requirements engineering, early and late design phases, and connectivity to other tools. The correct tool reduces redundant human tasks through automation, provides traceability and transparency throughout the data structure, and optimizes workflow. These benefits scale up by helping the team adhere to the predefined schedule and budget constraints.

Agile in the COVID-19 Pandemic

As the world shifts to prioritize digital business transformation to help manage distributed teams, many are moving away from antiquated systems development methods in favor of Agile.

A report published by Digital.ai found that 43% of organizations surveyed have increased their reliance on agile development in the aftermath of the COVID-19 pandemic [8]. Based on two consecutive surveys, conducted in December 2019 and mid-May 2020, the results show agile methodology has helped increase speed to market, improved overall team productivity, and enabled better management of distributed teams [8].

The success of Agile in the space sector is evident through companies including SpaceX, which is recognized as "advanced agile enterprises" [9,10]. SpaceX thrived in both the pre and post pandemic event, and is known for designing, building, and testing multiple types of prototypes of its products often [11], using the Agile methodology.

The customer centric development and adaptable work culture is what sets SpaceX apart from its competitors. Agile iterations cultivate a work environment of rapid production, active learning and testing, and incorporation of lessons learned into the next cycle. Rather than waiting until a product is "perfect", which does not take into account the dynamic nature of the space economy, Agile companies are able to quickly execute and innovate.

However, the digital transformation to agile is not always smooth. The resistance to change and an ill prepared transition to Agile can cause individuals and institutions to quickly revert to old and familiar methods of development. Teams which are accustomed to working in a face-to-face environment, or with less interaction with peers, may find the transition particularly difficult initially.

The following section provides guidelines on adopting the Agile approach for remote teams working within the space industry and beyond.

PRACTICAL GUIDELINES FOR REMOTE TEAMS IN THE AGILE SPACE INDUSTRY

This section provides organisations in the space industry with practical guidelines to benefit from the advantages of agile methodologies following the DDSE approach, in the evolving remote work culture. Five strategies to implement or transition to an Agile or hybrid-Agile approach within interdisciplinary engineering organisations for spacecraft production and mission design are discussed below.

1. Visualize requirements

Mission design and spacecraft development is a complex process with a large number of integrated components. A successful mission relies on a prerequisite of well defined requirements indicating the expected functionality and performance of the system(s), based on the customers' objectives. In an Agile environment the Requirement Engineering (RE) process cannot be confined to just the beginning of the development lifecycle. Instead, requirements can change at various points of the mission development process and lifecycle phases to accommodate for uncertainty in dynamic operations [12].

Today many teams use backlogs or spreadsheets for RE management. However, to truly understand the hierarchy between requirements, there must be a requirement repository with a visual element which facilitates an intuitive understanding of the status of development.

A requirement tree which clearly links all requirements from the highest level to the lowest allows the team to understand the hierarchy, relationships, dependencies between requirements, while providing an instantaneous snapshot of the project status. Visualizing requirements, when connected to project milestones and tasks, provides context, and shows the impact of changing one subcomponent or requirement as a ripple effect on adjacent or dependent requirements, and ultimately the whole planning. The visual model is critical in the development architecture to identify missing or neglected requirements, conflicts, or inconsistencies, and keeping the project on schedule and within budget.

2. Allow Concurrent Access

The development and integration of complex systems and systems of systems requires collaboration of contributors from different domains, including hardware, software, and services. [12]. Rather than a segmented and individualistic design and development process, an innovative process supporting parallel development is vital in creating systems and elements which integrate well.

This is where concurrent engineering (CE) processes and methods come in, enabling collaboration and information exchange between multidisciplinary teams. The goal of concurrent development is to improve quality, reduce product development time, lower costs, and enhance workflows throughout the lifecycle of a mission and spacecraft development [12]. This is achieved through the early involvement of participants, a holistic team approach, and simultaneous work on different systems and phases -- all enabled by concurrent access (CA), the information flow and access between the human capital across various cultural, disciplinary, geographic and temporal boundaries [12].

Simultaneous and rapid accessibility throughout the lifecycle is essential to concurrent access. The team synchronizes through sharing of a database and information, stored on a shared network drive or cloud. This facilitates a functional division of tasks, enables effective remote working, and allows for parallel development of systems and components arising from a single source of truth. The result is users working with the latest and most relevant data.

3. **Use Scalable & Connected Tools**

To facilitate concurrent access and streamline the engineering process of space hardware projects, the appropriate collaboration tool is needed. According to the European Space Agency (ESA), when it comes to streamlining the engineering process, collaboration tools are the backbone for every software project today [13]. Unfortunately, the go-to tool, spreadsheets, falls short when it comes to data structure and automation for engineering projects, leading to human error [14]:

- **Lack of data structure:** Without active planning and maintenance, spreadsheets can become a data jungle, making it difficult to retrieve the essential pieces of data
- **Automation is not thorough**: Due to a disconnect between numerical and non numerical values, its it difficult to visualize the overall impact of a change in values

For rapid prototyping and design iterations of mission and spacecraft design, the following functions are essential in the central platform the team will utilize for collaboration:

- Store and secure data regarding design specifications and requirements
- Exchange data and communication
  - Assignment and ownership of tasks and components to individuals or groups
  - Automatic design reviews, updates, and verification
  - Document management
  - Standardized work and test procedures
  - Analysis generation of engineering budgets

A central knowledge hub, using a single source of truth, is key in monitoring technological progress, managing risk, and preventing delays - all of which are a costly setback in today's competitive market. A collaborative platform which captures the history and tracks the status of each component, task, requirement, etc. enables control of changes. This makes it easier to identify the subsequent work items affected and minimizes overhead costs and time lost.

Whereas a typical spreadsheet is managed and controlled by a user, a competent tool alerts the users of changes through user-defined notifications. Fully automated systems issue warnings when a design does not meet the bounds of requirement, instead of the other way round. Thus, a user has a snapshot of adherence at any time as well as an overview of the critical path to completion.

4. **Ensure Bidirectional Traceability**

Successful bidirectional traceability allows everyone to understand the interconnectivity of different types of requirements at various levels of hierarchy. The traditional document-based systems make it difficult to achieve complete traceability. This is because the links between requirements, design, engineering analysis and testing are often missing, leading to a lack of information [12]. Thus, the lack of traceability results in low transparency.

Complete bidirectional traceability insures there are no

- **High level "childless" requirements:** every system element, subsystem, and component is linked to a business, stakeholder, or system requirement [12].
- **Low level "orphan: requirements:** lower level requirements which are not linked to a higher level requirement and are out of scope of the

project [12].

Without ownership of traceability, orphan requirements may go unnoticed whereas childless requirements may not be addressed at the appropriate time, both leading to unanticipated costs or extending the critical path to completion. Platforms with automated relationships between calculations and simulations take requirements monitoring to another level. The correct tool and strategy permit control, alerting users when constraints are in violation, and management through a Requirement Traceability Matrix (RTM) [4]. This shares and extends the comprehension of the requirement decomposition and derivation.

Furthermore, an automated RTM identifies imperfect requirements or designs and is programmed to fix the core defects instead of treating the symptoms. A smart web-based work platform results in robust requirements management, instinctively performing further system wide analysis to identify and correct other impacted items affected by defective requirements. Rapid reaction to unintended or unanticipated changes is key to the success of Agile Spacecraft development.

5. Establish & Fortify Communication

At the end of the day, the success of a mission is attributed to the people who enabled the technology, not just the technology itself. Team members must be in sync with work methodology and technology use to experience the full impact of Agile. This requires a twofold approach: 1) selecting the right process for the project at hand, and 2) selecting the right team.

The concurrent engineering approach requires both a social and technological approach to product and process development [12]. Agile relies on more interaction and exchange between the stakeholder and the developers, as well as the subdivisions and individuals within the organization. Its crucial team members have consistent communication. For example, some companies implement a 15-minute mandatory daily scrum meeting (sometimes referred to as the daily stand up meeting) which requires active participation from everyone on the team to answer the following topics [15]:

1. Progress made since the last meeting
2. Progress to be made in the current period
3. Obstacles which may hinder the current objective

These stand up meetings allow the team to identify hazards and allocate extra support and resources where needed. This is especially important for the growing number of remote teams, which are missing face-to-face interpersonal and social interaction, to identify the support that can be provided for project success and boost **REFERENCES** morale. In addition to meetings, clear and concise communication is key for data-driven engineering projects on web-based platforms.

A new culture of collaboration, customs, and language must be developed to facilitate transmission of information between individuals and subdivisions within a team. The work culture, methodology, acceptance criteria, and translation of requirements across the different domains of a company must be developed to prevent knowledge gaps, conflicts, confusion, and a lack of understanding [12].

CONCLUSION

Engineering projects are moving away from traditional document-driven models to the concurrent Agile approach which emphasizes data-driven engineering. This transition, which was already gaining momentum due to the competitive nature of the New Space industry, has accelerated to accommodate the new work culture in the COVID-19 pandemic period [8]. Companies which were able to adapt to mandated remote working fared better and were able to maintain operations with minimal interruptions.

While the benefits of working in an agile iterative process have been discussed, the transition to agile and remote working are not intuitive for pre-existing space organizations. Both the technological and social structures within organization, as well as external partnerships, need to be assessed and modified to thrive in this new environment. The guidelines emphasize using visual elements to foster an instinctive understanding of the systems specifications and development. For successful fully remote operations, team members need to be able to collaborate concurrently using a single source of truth, which requires investing in the appropriate tools for the tasks at hand. To ensure accountability, requirements must have bidirectional traceability, increasing transparency in the workflow. Lastly, and perhaps most importantly, fostering a culture of clear and concise communication within the team helps avoid the pitfalls of ambiguity and false or outdated information. While this paper addressed the space industry, the practical guidelines can be implemented in a wide variety of interdisciplinary engineering organisations adapting to remote working.

[1] Mosher, T., Kolozs, J., Colegrove, C., & Wilder, E. (2018). Agile Hardware Development Approaches Applied to Space Hardware. In 2018 AIAA SPACE and Astronautics Forum and Exposition, pp1-3. AIAA.

[2]Wortman, K., Duncan, B., & Melin, E. (2017, March). Agile methodology for spacecraft ground software development: A cultural shift. In 2017 IEEE Aerospace Conference pp. 1-8. IEEE.

[3] Huang, P. M., Darrin, A. G., & Knuth, A. A. (2012, March). Agile hardware and software system engineering for innovation. In 2012 IEEE Aerospace Conference pp. 1-10. IEEE.

[4] NASA (2007). *NASA Systems Engineering Handbook.* NASA Headquarters, Washington, DC, pp 16-33.

[5] Larson, W.J. & Wertz, R.W. (2005). *Space Mission Analysis and Design.* Microcosm Press, E1 Segundo, CA, USA, and Kluwer Academic Publishers, Dordrecht, The Netherlands, pp 1-93.

[6] Trimble, J. & Webster, C. (2012). Agile Methods for Space Operations. NASA Headquarters, Washington, DC, pp1-8.

[7] Lindblad, L., Witzman, M., Bussche, S.V. (2018). Data-driven Systems Engineering: Turning MBSE into Industrial Reality. In International Systems & Concurrent Engineering for Space Applications Workshop (SECESA).

[8] Walker, K. (2020). 14th Annual State of Agile Report Shows 60% of Respondents Have Increased Speed to Market and 55% Are Planning to Implement Value Stream Management. Online at https://www.businesswire.com/news/home/202005280 05097/en/14th-Annual-State-Agile-Report-Shows-60 (as of 30 July, 2020).

[9] Nesta (2020). Scaling 'agile' management approaches to entire organisations and new sectors. Online at https://www.nesta.org.uk/feature/innovation squared/scaling-agile-management-approaches-entire organisations-and-new-sectors/ (as of 30 July, 2020).

[10] Noble, A., Rigby, D.K., & Sutherland, J. (May - June 2018) Agile at Scale. Harvard Business Review, pp 88–96.

[11] Denning, S. (2020). Why Hardware Must Embrace Agile Principles. Forbes. Online at https://www.forbes.com/sites/stevedenning/2020/06/21/why-hardware-must-embrace-agile principles/#6cec7f4875c8 (as of 30 July, 2020).

[12] Stjepandić, J., Verhage, W.J.C., (2015). *Concurrent Engineering in the 21st Century.* Springer International Publishing, Switzerland, pp 2-830.

[13] ESA (2019). Valispace: the smart collaboration platform for engineers. Online at https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Technology_Transfer/Valispace_the_smart_collaboration_platform_for_engineers (as of 30 July, 2020).

[14] Bussche, S.V. (2018). Why you need more than a spreadsheet for your engineering data. Online at https://www.valispace.com/beyond-spreadsheets/ (as of 30 July, 2020).

[15] Sutherland, J.J. & Sutherland, J., (2014). *Scrum: The Art of Doing Twice the Work in Half the Time*. Penguin Random House Company, New York, USA