



Toward a Hybrid Agile Product Development Process

Nicola Garzaniti^(✉), Clément Fortin, and Alessandro Golkar

Skolkovo Institute of Science and Technology, Space Center, Nobel Street 3, 464 Skolkovo,
Moscow 143026, Russia
nicola.garzaniti@skoltech.ru

Abstract. Both startups and traditional space industry are exploring new business opportunities and new engineering approaches in what is becoming known now as New Space.

Organizations started accelerating project schedules and challenging the V-model typically used in space product development. The need of a faster and more adaptive response to changing customer needs within an improved development productivity made Agile process a potential key enabler of New Space sector.

On the other hand, space system projects are typically executed in multi-party consortia. Each organization in a consortium adopts its own product development process and interprets “New Space” differently. For this reason, the implementation of Agile is not seamless as it requires coordination with traditional systems engineering approaches. This setup is what we refer to as “hybrid Agile product development process”.

This paper provides a first definition of the architecture of hybrid product development process targeted toward systems development and lifecycle management of hardware projects developed by multi-party consortia. We consider this discussion in the context of the development of spaceflight hardware in the New Space industry. We identify the main challenges in adopting such a methodology in developing hardware systems. This work identifies opportunities of future work for defining coordination approaches in hybrid product development settings, and improved organization structures of hardware projects in hybrid development contexts.

Keywords: Collaborative product development process · Hybrid agile process · New Space · Agile PLM

1 Introduction

In the last decades we witnessed radical changes in global space activity with greater involvement from the private sector. These changes come together under the definition of “New Space” [1].

Organizations started accelerating project schedules and challenging classical product development processes such as the V-model [2] and the Stage-Gate model [3] typically used in space system developments. The need of a faster and more adaptive response

to changing customer needs within an improved development productivity makes Agile process [4] a potential key enabler of the New Space sector. Nevertheless, the question of whether Agile can truly fulfill its promise in complex hardware developments has not yet been thoroughly validated from a scientific perspective.

Several research projects have been dedicated to extend and tailor the Agile principles to hardware product development [5–7] but most process implementations were unsuccessful from the methodology point of view [7, 8].

The main reason behind those failures is potentially that space system projects are typically executed in multi-party consortia. Each organization in a consortium adopts its own product development process and interprets New Space differently. Therefore, the implementation of Agile is not seamless as it requires coordination with traditional systems engineering approaches.

This paper investigates this concern by providing a definition of Hybrid Product Development Process (Hybrid PDP) targeted toward systems development and lifecycle management of hardware projects developed by multi-party consortia.

We identify the main challenges in adopting such a methodology in developing hardware systems. We propose a coordination approach for hybrid Agile product developments and discuss strengths and limitations based on our initial findings. We limit the scope of the paper to the development of spaceflight hardware in the New Space industry context.

The remainder of the paper is structured as follows. Section 2 illustrates the architecture of hybrid product development. Section 3 discusses our preliminary findings on the implementation of Hybrid PDP on an industrial use case. Section 4 draws conclusions from the research and identifies avenues of future work.

2 Hybrid Product Development Architecture

The goal of our investigation is to define an approach to embed Agile in the traditional stage gate product development process. We analyze from a scientific perspective whether Agile combined with traditional stage-gate can work symbiotically, or whether the two approaches are mutually exclusive or just incompatible.

We provide a first definition of hybrid product development process from a holistic perspective targeted toward systems development and lifecycle management of hardware projects developed by multi-party consortia.

We consider this discussion in the context of the development of spaceflight hardware in the New Space industry.

2.1 Stage-Gate Model: Strengths and Weaknesses

Stage-gate approach [3], also called waterfall, phase gate, toll gate, checkpoint, or structured product development by different authors and practitioners [9] is a well-established product development process. Stage-gate has been designed to help firms to select the right projects, and once selected, to map out the key stages, best practice activities, and roles and responsibilities as part of the project, bringing discipline to “*chaotic*” new product development (NPD) activities [10].

The ideal stage-gate process proceeds in distinct stages from product planning to product release (Fig. 1). At the end of each phase is a review, or gate, to evaluate whether the previous phase was successfully completed. If the project is reviewed positively, work proceeds to the next phase. If not, then the project iterates within that phase until it can successfully pass the review or the project may be terminated.

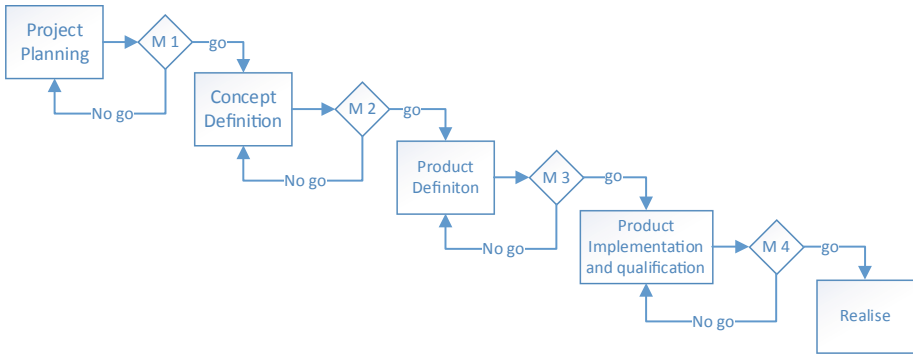


Fig. 1. Stage-Gate model

The major advantage of stage-gate processes is to provide structure to the development by reaching sharp product definitions and specifications early in product development. Technical risk is reduced because narrow iterations and reviews freeze specifications early. Rigid requirements and stable product definitions help to avoid errors by avoiding midstream corrections [9].

The main drawback of this product development process (PDP) is inflexibility. Narrow iterations cannot incorporate feedback from later phases. Failure may occur if early specifications and assumptions are proven wrong by subsequent market research or prototyping.

2.2 The Agile Way of Implementing Projects

Agile is a method that brings flexibility and speed to development projects: It includes micro-planning tools to get a working end-product quickly. This PDP is designed specifically for managing and supporting product developers in developing their system once the development project has been “approved.”

Agile, particularly the Scrum version (Fig. 2) is a methodology that breaks the development process into a series of short, iterative, incremental *sprints* (i.e. development period), each one to four weeks long. The main goal of Scrum is to deliver a *Minimum Viable Product* (MVP) at the end of each sprint (i.e. development period).

The work is decomposed into *stories* related to the development of the product. This decomposition provides structure to the development process. The collection of user stories forms the *product backlog* for the development. Each user story is characterized by a subset of the main high-level tasks (high-level product backlog) that will be divided later on, during the sprint planning, into smaller tasks that could be completed in a one-day time-frame. The goal in Sprint planning is to define a *Sprint backlog* taking tasks out

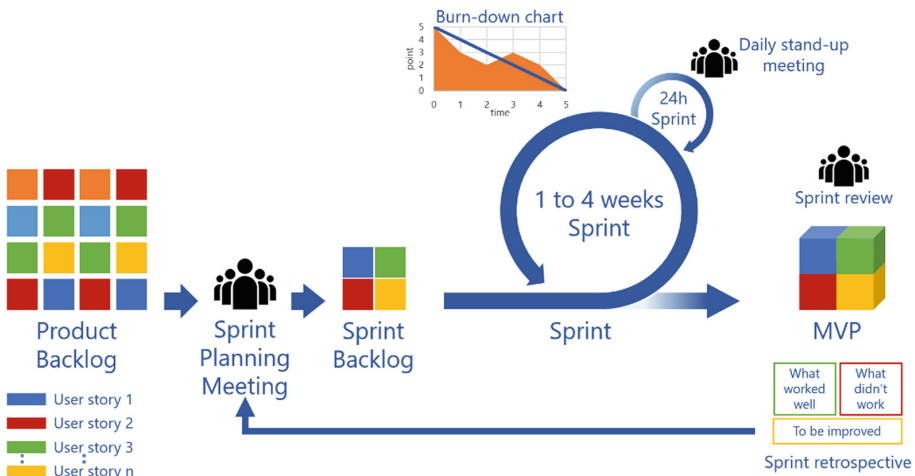


Fig. 2. Scrum process diagram

of the product backlog. Sprint planning is performed collaboratively among the members of the team using a Fibonacci sequence [11] scoring system to evaluate the complexity of the tasks and prioritize them within the one-week time box. A team member is assigned the role of “*Scrum master*,” that is the facilitator in charge of coordinating inputs and running the Agile process. At the end of each Sprint, the team performs a retrospective analysis of the work in order to prepare for the next development iteration.

The tempo of the Sprint is given by daily 15-min stand up meetings and daily close-out meetings. In order to assess the correct implementation and to constantly monitor the development of the system, the team makes use of a set of tracking technologies to monitor the status of the process (e.g. Atlassian Jira [12]).

The main advantages of Agile are improved communication and coordination, quicker releases, and flexibility to allow quicker responses to changed customer requirements or technical context. However, Scrum presents some challenges for manufacturers, such as a lack of scalability, a proliferation of meetings, and very often a lack of management [13].

2.3 Hybrid Product Development Process for Physical Products

Hybrid Product Development Process embeds the Agile way of working within stage-gate driven product development. Hybrid PDP integrates traditional project management with Agile. The structure of hybrid product development can be described as a three-layer architecture (Fig. 3) composed of multiple project participants, each operating with its own product development process (Agile, stage-gate, or others).

At the top layer of the architecture we have the *consortium layer*. The consortium is the coordinating agent of the PDP; it provides overall management and coordination of the project. It is in charge of the governance of the project. The consortium defines overall mission requirements, the functional system requirements, the interfaces among

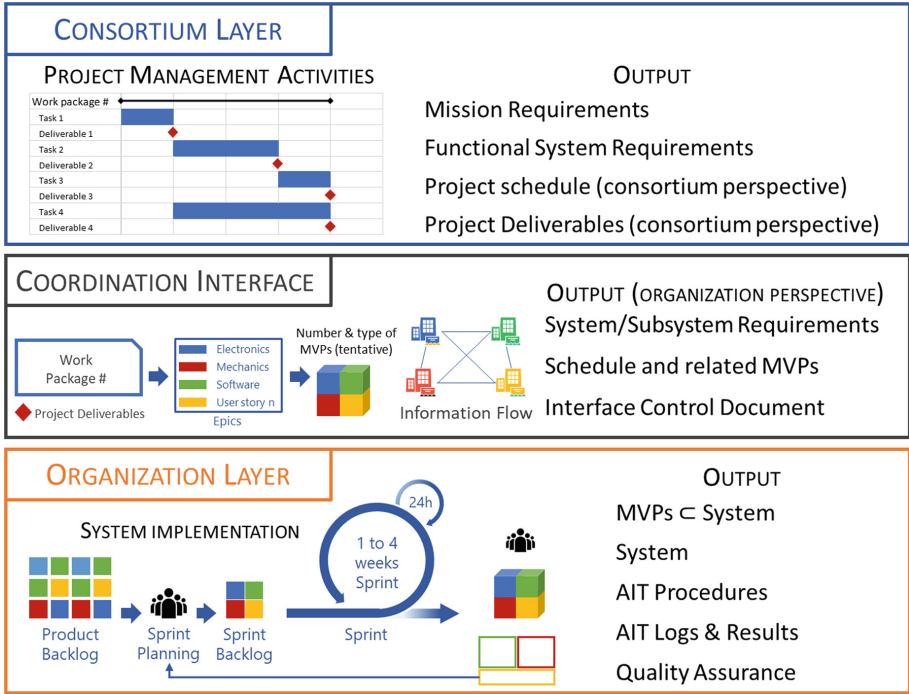


Fig. 3. Hybrid product development process architecture

all parties in the project and it coordinates project reviews and deliverables. The consortium is also in charge of making strategic decisions throughout the lifecycle of the project. Consortia can be operated using either PDP (Agile or stage-gate). In our initial investigation, we focus on consortia operating using traditional stage-gate processes.

Due to its structured nature, stage-gate provides natural means of coordination through decision gates and milestones, and it has been proven to work with very large, complex organizations. Future research, however, will investigate the possibility for having consortia operating using Agile, and investigating the conditions by which such process brings advantages (or disadvantages) compared to a more traditional stage-gate approach.

The *organization layer* sits at the bottom layer of the architecture. This layer groups together all the organizations participating the project. Each project participant operates its own product development process, and coordinates with each other through interfaces with the consortium layer.

The coordination is implemented through *coordination interface* defined at organization level. Each organization participating to the project defines the work packages or the minimum viable products (MVPs) to meet the deliverables defined in the top layer depending on the PDP they are operating. This layer can also implement information coordination through direct interfaces between the participating organizations. In the current setting we neglect the latter, and focus on the main structured means of coordination.

The key challenge in hybrid PDP architectures is reconciling coordination between Agile and stage-gate, that are designed under different operating principles. While the former aims at building incremental minimum viable products towards the target performance, the latter has the goal of achieving the target with no intermediate deliverables.

In particular organizations implementing Agile need to creating a project development backlog (for their own product) for roughing out a high-level tentative development plan that meet the project milestones. Then they need to ensure that the backlogs developed for sprints are consistent with the product definition approved at the gate.

3 Preliminary Findings

We report preliminary findings on the analysis of a hybrid development architecture such as the one described in Sect. 2.3 on a use case of development of a New Space product.

The product is an engineering system developed in a multi-party consortium including different entities: universities, small and medium enterprises, supervised by an institutional partner.

At the beginning of the project, all the participants met together to define overall mission requirements. Mission requirements have been consolidated in a Mission Requirements Document (MRD). Then each organization has broken down the requirements of the MRD into system requirements for their specific contribution to the project. Those requirements have been formalized into a System Requirements Document (SRD).

Based on those two documents (MRD and SRD), each organization has defined its own Product Development Process (PDP). The prime contractor structures the PDP at consortium level, defining in a Gantt chart the high-level activities and the main milestones. Traditional Space Flight Project Life Cycle (Fig. 5) has been adopted [14].

At the time of writing of this paper, the consortium has completed the path up to final system delivery.

In our investigation, we focus on one project participant that adopted Agile methodology to deal with fixed schedule and cost constraints. We analyze the metadata concerning the project the organization has provided. To ensure delivery of the system on a schedule of 12 months, they structured the development following an Agile approach (Scrum development process).

The organization moved directly from the *consortium layer* to the *organization layer* without implementing *coordination interfaces*. Because of this initial weakness in hybrid product development process several issues were encountered.

3.1 Work Breakdown

The organization implementing agile started decomposing the work into *user stories* related to the development of the main subsystems of the product, shaping the so called *product backlog* (Fig. 4). Within this activity the team did not set up *a priori* the Minimum Viable Products (MVPs) needed to meet the deliverable foreseen in the main schedule.

The lack of clear objective for all the MVPs and a clear view of the *big picture* yielded the rescheduling of several activities (a percentage close to the $35\% \pm 6\%$ of the total

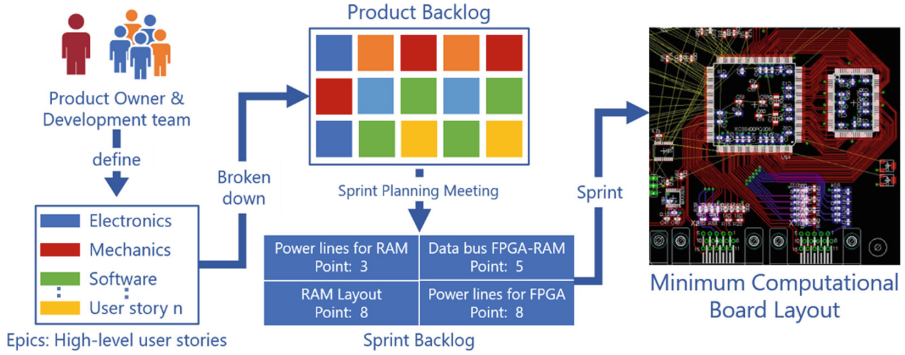


Fig. 4. Example of Agile implementation to realize one MVP

points foreseen for a given Sprint). That caused delays (in Fig. 5) and non-recurring engineering efforts (NREs). Furthermore, without a high-level tentative development plan, it was not possible to address early programmatic risks and develop a mitigation strategy.

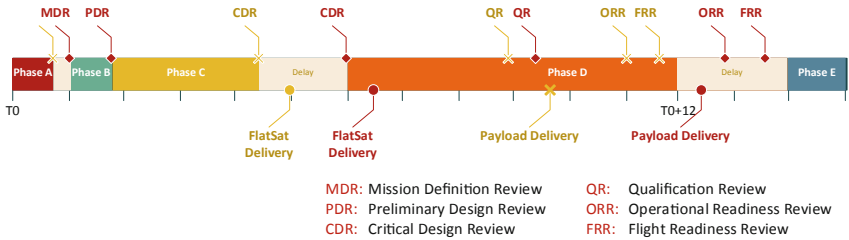


Fig. 5. Use case Project Life Cycle - we marked in yellow the foreseen milestones and in red the actual milestones (Color figure online)

3.2 Coordination with Partners

In the traditional Scrum approach all the development depends on the team. In Scrum, the development is entirely managed using Agile. Within a hybrid development architecture each organization in the consortium needs input from the other project participants and has to provide input to others.

The interfaces among all parties in the project have been defined at the consortium level. Technical interfaces have been formalized in the SRD (interface sections) and in the Interface Control Document (ICD). Due to the nature of the Agile process, most of the interfaces were not yet defined at the time of the consolidation of SRD and ICD (the system was at the concept level). Therefore, in order to deliver the document according to the schedule the development team has to make assumptions that sometimes have proven to be wrong in the later stages. This caused *requests for deviation* that impacted all the consortium.

3.3 Finding

Based on this experience we highlight the challenge they face, and use it as a lesson learned for a future implementation of a Hybrid Agile Product Development Process. We summarize the preliminary finding using a SWOT analysis framework (Strengths, Weaknesses, Opportunities, and Threats).

Table 1. SWOT analysis of Hybrid PDP

<p><i>Strengths</i></p> <ul style="list-style-type: none"> • Combining the ability to better adapt to changing requirements due to Agile, while keeping structured pace in the development through staged-gate processes • Allowing collaboration between organizations adopting heterogeneous product development processes • Accelerating project schedule by learning and adapting through rapid engineering iterations • Assigning clear responsibility and accountability of user stories to each team member • Providing clear traceability of work history through backlog review 	<p><i>Weaknesses</i></p> <ul style="list-style-type: none"> • Lack of strong coordination between layers may lead to suboptimal results due to asynchronous pace of development • Lack of rigorous coordination methods requires tailoring of hybrid PDPs to the specific characteristics of each product development • Challenges in implementing rigorous verification and validation due to potentially diverging integration readiness of interdependent system components • Challenges in reconciling incremental minimum viable products with intermediate system development stages typical of stage-gated processes
<p><i>Opportunities</i></p> <ul style="list-style-type: none"> • Delivering the advantages of Agile to complex collaborative development projects, while ensuring interoperability with traditional PDPs • Accelerating speed of execution and product delivery to the market • Improved alignment with rapidly evolving competitive environments due to the inherent adaptability of hybrid PDPs • Potentially reducing development costs by shifting risk posture in complex hardware development projects 	<p><i>Threats</i></p> <ul style="list-style-type: none"> • Lack of coordination potentially leading to interface mismatches or failures between system elements developed by different organizations • Lack of understanding between project participants or by the coordinating consortiums of the fundamental differences between stage-gated and Agile development processes • Lack of understanding of the motivations underlying the implementation of a hybrid PDP does not allow for efficient exploitation of benefits of Agile developments

4 Conclusion

A hybrid product development process that integrates elements of both Agile and Stage-Gate can help companies capitalize on the strengths of both.

Stages and gates remain an important part of this hybrid model. Stages provide a high-level overview of the project’s main phases and a guide to required or recommended

activities and expected deliverables for each stage. Gates provide vital go/kill decision points providing focus in the development pipeline.

Agile is used to structure execution within the development teams. In this way the deliverables specified for each gate are leaner, more flexible and less granular than in the classical model, and they are physical prototypes rather than reports or slide presentations.

The key element of the hybrid PDP is the *coordination interface*. It is crucial to reconcile incremental minimum viable products with intermediate system development stages typical of stage-gate processes.

The definition of high-level tentative development plan at the *coordination interface level* is needed to address early programmatic risks in the project and develop a mitigation strategy (it is not a common practice in pure Agile approach).

In this paper we described a use case of a hybrid product development architecture where the consortium did not implement a coordination layer. For this reason, coordination misalignments emerged, such as schedule and cost overrun, NREs and quality assurance issues related to the risk mitigation strategy and product verification and validation. We used the challenge faced by the project participants as a lesson learned to shape a preliminary implementation of a Hybrid Agile Product Development Process.

By combining Agile with Stage-gate, the hybrid PDP aims at accelerating project schedules and overcoming organizational limitations in New Space project development (or highly innovative projects in general). There are still challenges in reconciling the two approaches (as shown in Table 1). This work intended to contribute to this future and this vision shaping a strategy to enable the implementation of a hybrid PDP.

References

1. Paikowsky, D.: What Is new space? The changing ecosystem of global space activity. *New Space* **5**, 84–88 (2017). <https://doi.org/10.1089/space.2016.0027>
2. Lee, J.S., Miller, L.E.: *INCOSE - Systems Engineering Handbook* (2007). <http://dl.acm.org/citation.cfm?id=552393>
3. Cooper, R.G.: Stage-gate systems: a new tool for managing new products. *Bus. Horiz.* **33**, 44–54 (1990). [https://doi.org/10.1016/0007-6813\(90\)90040-I](https://doi.org/10.1016/0007-6813(90)90040-I)
4. Beck, K., et al.: *Manifesto for Agile Software Development* (2001). <http://agilemanifesto.org/>
5. Thomke, S., Reinertsen, D.: Agile product development: managing development flexibility in uncertain environments. *California Manag. Rev.* **41**, 8–30 (1998). <https://doi.org/10.2307/41165973>
6. Huang, P.M., Darrin, A.G., Knuth, A.A.: Agile hardware and software system engineering for innovation. In: 2012 IEEE Aerospace Conference, pp. 1–10. IEEE (2012). <https://doi.org/10.1109/aero.2012.6187425>
7. Garzaniti, N., Briatore, S., Fortin, C., Golkar, A.: Effectiveness of the scrum methodology for agile development of space hardware. In: 2019 IEEE Aerospace Conference (2019). <https://doi.org/10.1109/AERO.2019.8741892>
8. Stepanova, D., Ballheimer, W., Van Ginkel, G., Joos, C., Skyrpnyk, I.: Scrum methodology in aerospace projects. In: 69rd International Astronautical Congress, Bremen, Germany (2018)
9. Unger, D.W., Eppinger, S.D.: Comparing product development processes and managing risk. *Int. J. Prod. Dev.* **8**, 382 (2009). <https://doi.org/10.1504/IJPD.2009.025253>

10. Sethi, R., Iqbal, Z.: Stage-gate controls, learning failure, and adverse effect on novel new products. *J. Market.* **72**, 118–134 (2008). <https://doi.org/10.1509/jmkg.72.1.118>
11. Falcon, S.: The k-Fibonacci difference sequences. *Chaos, Solitons Fractals* **87**, 153–157 (2016). <https://doi.org/10.1016/j.chaos.2016.03.038>
12. Jira | Issue & Project Tracking Software | Atlassian, (n.d.). <https://www.atlassian.com/software/jira>
13. Gregory, P., Barroca, L., Taylor, K., Salah, D., Sharp, H.: Agile challenges in practice: a thematic analysis. In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) *XP 2015. LNBP*, vol. 212, pp. 64–80. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18612-2_6
14. NASA, NPR 7120.8 Nasa Procedural Requirements (2008). <https://nodis3.gsfc.nasa.gov/displayDir.cfm?t=NPR&c=7120&s=5E>