

Status Quo of Agile Software Development in the European Institutional Space Flight

Stefan Brüggemann, Christian R. Prause

Abstract

Institutional space flight is increasingly realizing the importance of agile development methods. This goes hand in hand with a shift to NewSpace products and technologies. In this paradigm, short development cycles and flexibility are particularly important. The European (ESA) and German Space Agency (DLR), and their partners, are currently working jointly on two artifacts that lay the foundation for Agile development in institutional spaceflight: (1) An ECSS standard (handbook) on agile software development has passed the public review and is now being finalized. (2) For the first time, the Space Administration of DLR has adopted a set of requirements for suppliers in its product assurance requirements that are geared towards agile development. This paper is the summary of a presentation at this year's German Aerospace Congress. It briefly describes agile development, introduces the two artifacts and shows how agile development can be integrated into the framework institutional spaceflight. The space community approaches the subject both with interest and critical questions on possible consequences (e.g., quality, contract design).

This paper is a translation of the original paper Brüggemann & Prause (2018). The original German version is reprinted in the appendix.

Keywords

Space flight, agile software development, software engineering, NewSpace, ECSS, ECSS-E-HB-40-01A, space agencies, product assurance, quality

1. Introduction

Institutional European space travel is increasingly recognizing the importance of agile development methods. Agile development is about quickly getting products and services to the market, to collect feedback on these, and optimize the products. This goes hand in hand with a change in the product portfolio towards NewSpace products and technologies. In the NewSpace paradigm, it is especially important to have short development cycles with enough flexibility, to adequately take customer requirements and markets into account (Horn, Frischauf, Baumann, & Heinrich, 2016). In the last year, two key artefacts were created that bring together these efforts at DLR and ESA: As part of the ECSS (European Cooperation for Space Standardization), a working group (ECSS-E-HB-40-01A, 2018) with industrial and institutional representatives has been summoned. It pursues the goal of describing how agile software development can be applied in contexts where classic ECSS standards have to be applied. The handbook passed the public review of the space community in 2017. The comments that came from the review are currently worked into the final version of the handbook.

Summoned by the German space agency, the heads of the quality departments of the most important space companies in the National Programme for Space and Innovation meet regularly. The joint working group discusses topics of product assurance. The aim is to specify generic (i.e., independent of an individual project) quality and process requirements, which shall be used in the space agency's procurement projects. In the past year, the most important "quality folks" convened again. On this occasion, the state of the art of all relevant technical disciplines (hardware and software) is revisited. If found relevant, respective regulations are integrated into the generic requirements catalog of the national space program. For the first time, agile software development was regarded as important, and respective requirements were added to the catalog.

This paper discusses the motivation for (Section 2) and properties of (section 3) agile software development, and shows how agile development permeates into the European institutional spaceflight through the new ECSS document (Section 4) and through the product assurance requirements catalog of the German space agency (Section 5). Finally, we critically review open questions related to agile development in spaceflight (Section 6).

2. Why Agile Development?

The world is becoming increasingly complex and thereby less predictable than before. This also shows in new technologies. But not only there: In all areas, including the supposed everyday things, complexity increases. Furthermore, many companies no longer work only locally, but Europe- and world-wide. When entering new markets, they face new tasks and new risks. In institutional space flight, there are traditionally many different parties involved in projects: Space agencies, contracting entities, contractors, various levels of subcontractors, operating organizations, end users such as scientists, providers of technologies, inspection bodies, etc. They often come from different countries. Due to the modern communication tools, speed, range and the level of integration in communication increases. In particular, software is considered as a "sponge" for complexity in technical things (ECSS-E-HB-40A, 2013).

These circumstances also change the role of leaders: they cannot handle everything alone anymore in a just-seize-it-and-decide-alone fashion. Their job becomes rather to serve their team by setting the frame conditions that enable the organization to do the right thing, to coach and to guide. They have to empower people, enable motivation and thereby increase performance, productivity, and efficiency. A new "Fehlerkultur" (error culture) is needed that allows mistakes and paves the way for commitment and responsibility. This is where NewSpace projects come in, and specifically in the field of software, agile development. However, as in classical projects, big failures must be avoided. Therefore, projects are done iteratively and in small steps. The individual steps can be done and finalized. Once done, they are really over. This is when real feedback follows. The small, self-contained units are an added value by themselves; in particular, when the most valuable units are done first. Thereby the time to market gets shorter. Please refer to Figure 1, as well as Bund, Heuser & Tatje (2012); Horn, Frischauf, Baumann, & Heinrich (2016); Kreuzer & Haader (2017) and Pernot (2013).

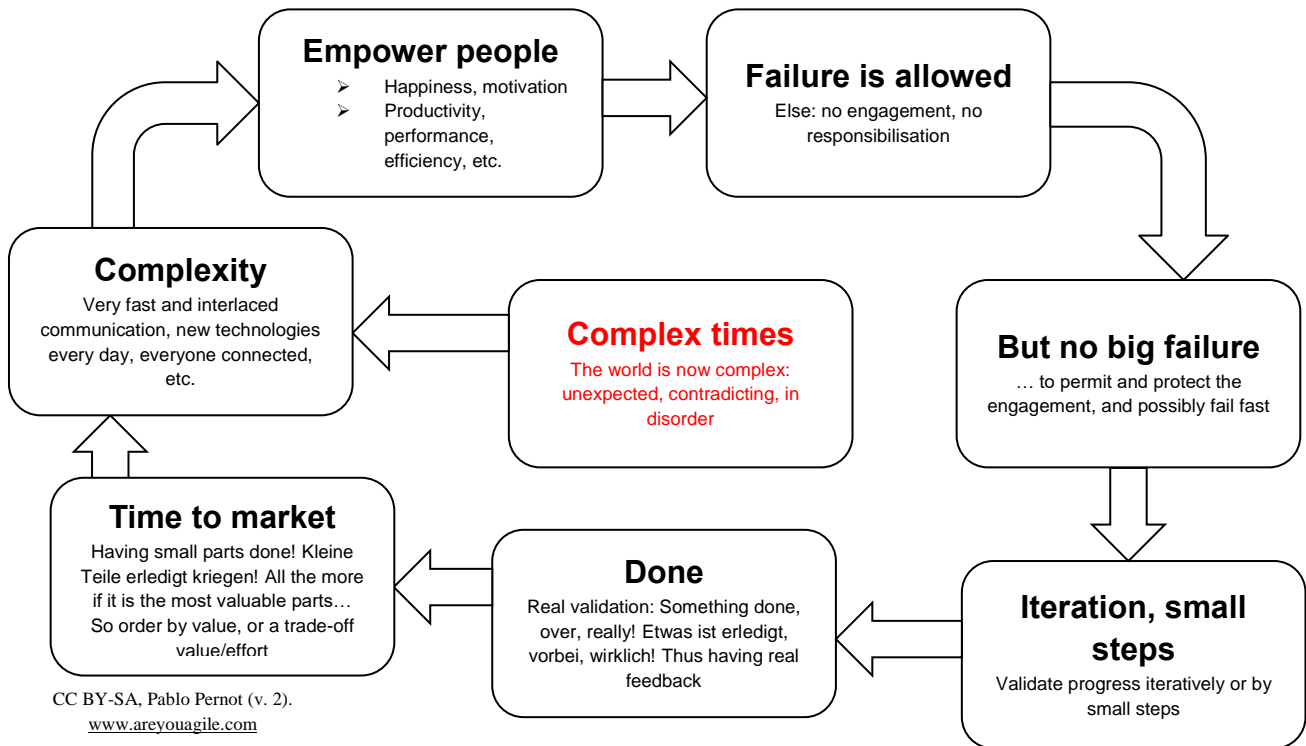


Figure 1: Agile development in complex times (Pernot, 2013)

3. What is Agile Development?

The term "agile software development" goes back to the Agile Manifesto (see Figure 2). It denotes those methods and practices of software development, which build its values and principles. The manifesto was signed in 2001 by the first seventeen signatories (including well-known software developers like Kent Beck, Alistair Cockburn, Ward Cunningham, Martin Fowler or Ron Jeffries.)

Agile principles include customer satisfaction, delivery of new versions of functional software at short intervals, almost daily cooperation between experts and developers, creating a motivating environment, maintaining a steady pace of work, technical excellence and good design, self-organization of the teams and self-reflection (Laplante, 2012). The principles are supported through agile methods such as pair programming, test-driven development or continuous refactoring, and implemented through agile processes like Extreme Programming, Feature Driven Development or

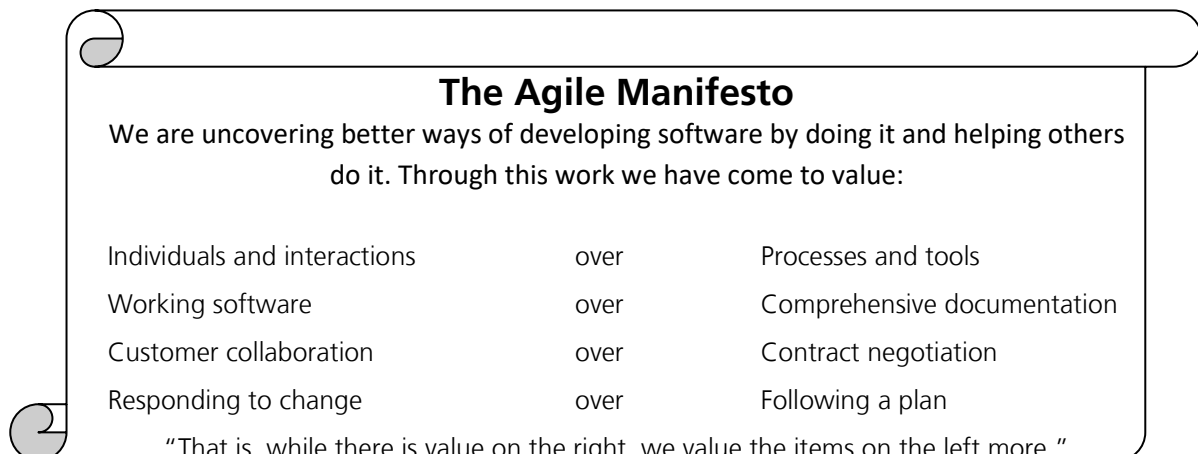


Figure 2: The Agile Manifesto

Scrum. In the past two decades, agile software development has become mainstream in more and more areas (see Kuhrmann et al., 2018).

As part of their investigation on NewSpace, Horn, Frischauf, Baumann, & Heinrich (2016) get to the conclusion that agile software development is an approach that is "increasingly transferred to spaceflight. In these procedures, development is not planned in advance in all details and then done accordingly. Instead, the development is iterated cyclically. This makes it possible to change requirements during the project duration and to continuously make adjustments." [Quote translated from German.]

4. ECSS Agile Software Development Handbook

The ECSS was founded in 1993 as an independent association supported by ESA, national European space agencies, and the interest group of the industry Eurospace. It creates common standards for project management, development, construction and operation of space systems. The members of ECSS are (at the same time) developers and users of the standards. The standards concentrate on project work and processes: on development, production and operation of technical systems in space (e.g., satellites or space probes) and on ground (e.g., send, receive and control centers). The ECSS is divided into the following sections Space Project Management (standards in ECSS M-Branch), Space Engineering (E-Branch), Space Product Assurance (Q-Branch), and Space Sustainability (U-Branch)

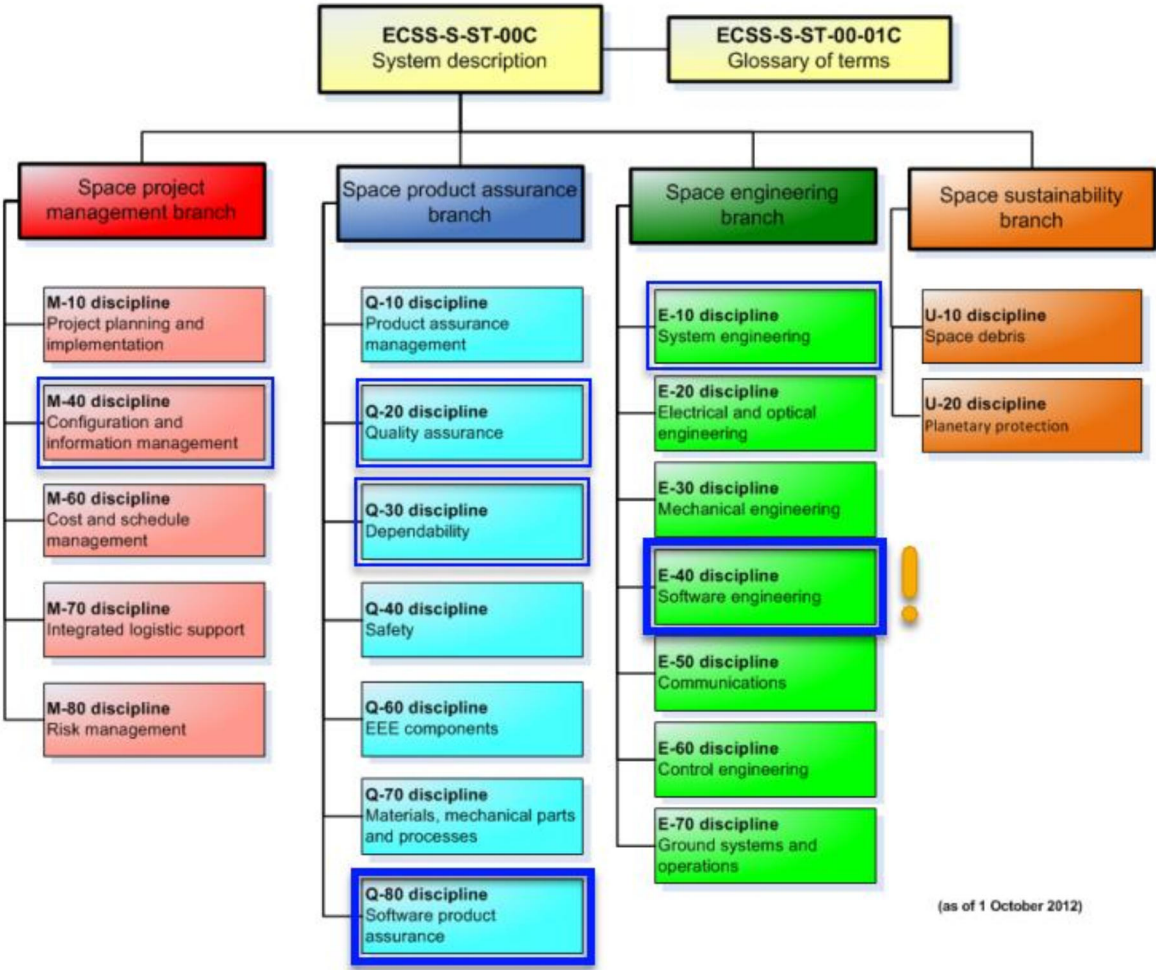


Figure 3: Overview of the main standards of the ECSS (2012)

(Schiller & Heinemann, 2014). The two main standards for software development are ECSS-E-ST-40C “Software Engineering” (in the E-Branch) and ECSS-Q-ST-80C “Software Product Assurance” (in the Q-Branch). Additionally, various more general standards like ECSS-M-ST-40C (Configuration Management) or ECSS-Q-ST-30C (Dependability) apply to software development. The new handbook on agile software development ECSS-E-HB-40-01A is subordinate to the main standard ECSS-E-ST-40C (see Figure 3).

The working group to develop the new ECSS Handbook is composed of industrial and institutional representatives from ESA, DLR, CNES, GMV, Thales Alenia Space, Airbus DS, Telespazio, Intecs and SpaceBel. The handbook has gone through the public review of the European space community from April 2017 to January 2018, and the comments resulting from it are currently being incorporated into the final version of the handbook.

The aim of the handbook is to describe how agile software development works when classical and established ECSS standards are applied. The handbook presents a guideline for the application of agile development approaches in compliance with ECSS-E-ST-40C and ECSS-Q-ST-80C. It covers all areas addressed by these two standards, including flight software, ground software and simulators. Not covered are e.g. topics of risk-based tailoring of the ECSS standards or contractual aspects of this particular development approach (cf. ECSS-E-HB-40-01A, 2018).

Specifically, a handbook which introduces into agile software development has been developed. It addresses topics such as characteristics of agile development, lean management, and agile methods. One chapter discusses general considerations and questions that arise long before agile methods are implemented. The next chapter contains guidelines for the selection of an agile life cycle model. For example, key factors of the customer, supplier, project and team context are discussed. The core part of the document introduces the Scrum process and how it can be adapted to space project requirements. The Scrum process is one of the most widely used agile methods. After the

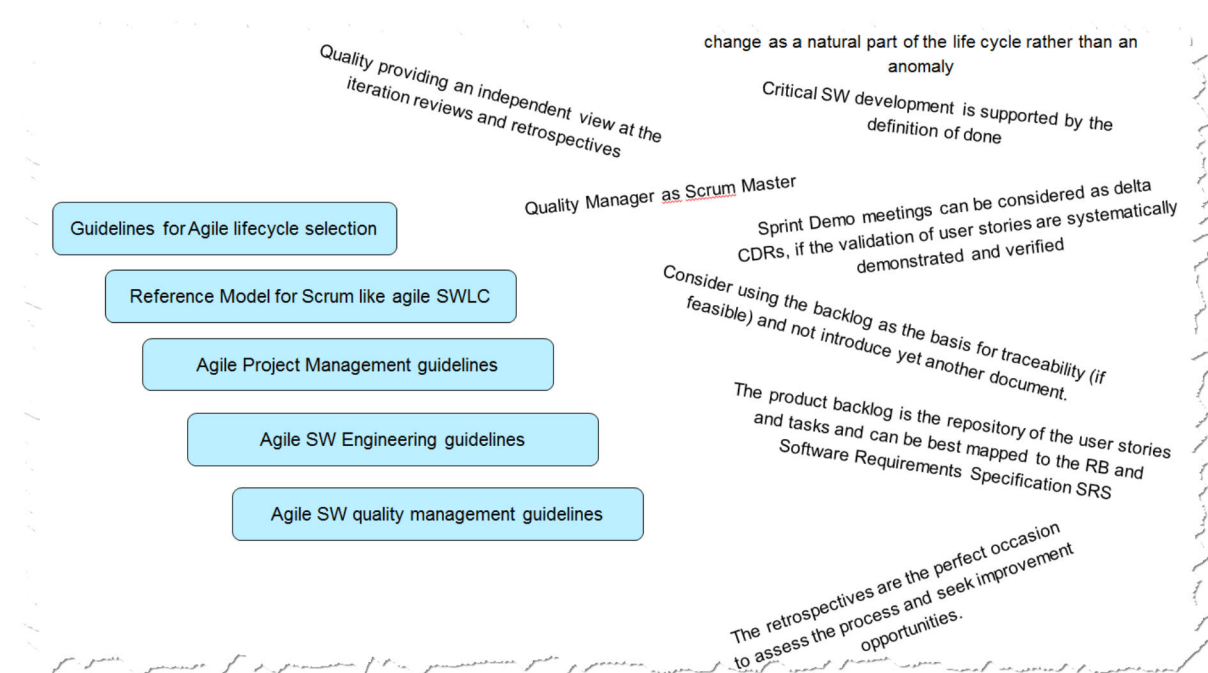


Abbildung 4: Themen des und Zitate dem Handbuch ECSS-E-HB-40-01A

introduction, the handbook describes for all central disciplines such as project management, software development, configuration management and quality management in detail and how these can be designed in an agile project scenario. The handbook concludes with various models of how classic ECSS reviews can be combined with agile methods.

Project characteristics	model 1	model 2	model 3	model 4
Strong coordination with higher level team via reviews	Green	Green	Yellow	Yellow
Strong coordination with higher level team via agile	Red	Red	Green	Green
Formal endorsement required	Green	Yellow	Green	Yellow
Requirements need to be flexible	Red	Red	Yellow	Green

Figure 5: Project characteristics of life cycle models compatible with ECSS. Source: ECSS-E-HB-40-01A (2018)

4.1 Example: ECSS Reviews with Agile Development

Four different models were developed to implement agile development in ECSS projects (see Figure 5): There are review-driven software lifecycles that are closer to ECSS, and sprint-driven software lifecycles that are more like Scrum. A table in the handbook can be used to compare key project characteristics with recommendations for or against one or the other of the four models.

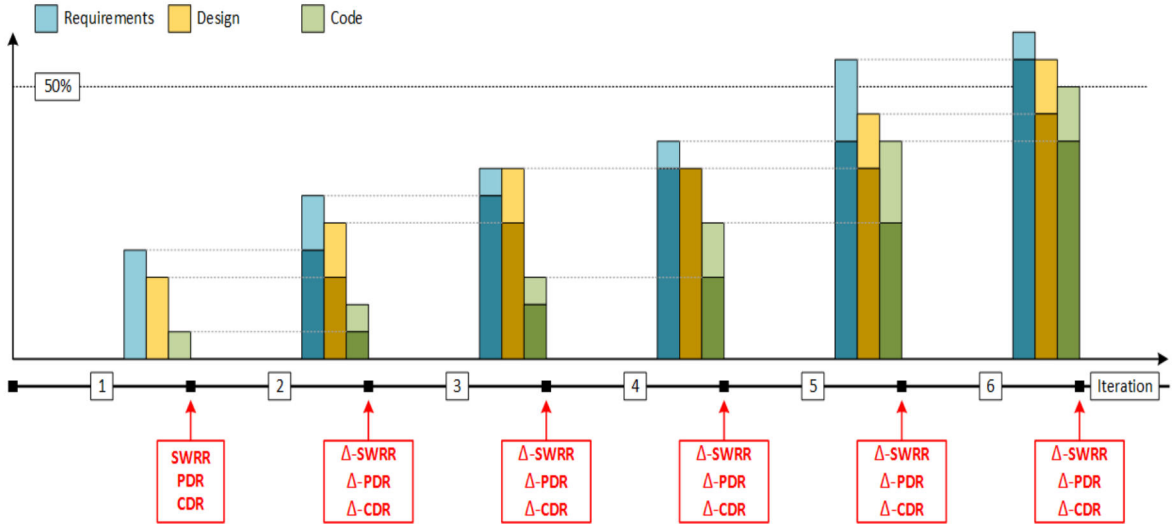


Figure 6: Scrum sprints in between reviews (ECSS, model 4). Source: Buzgan (2017)

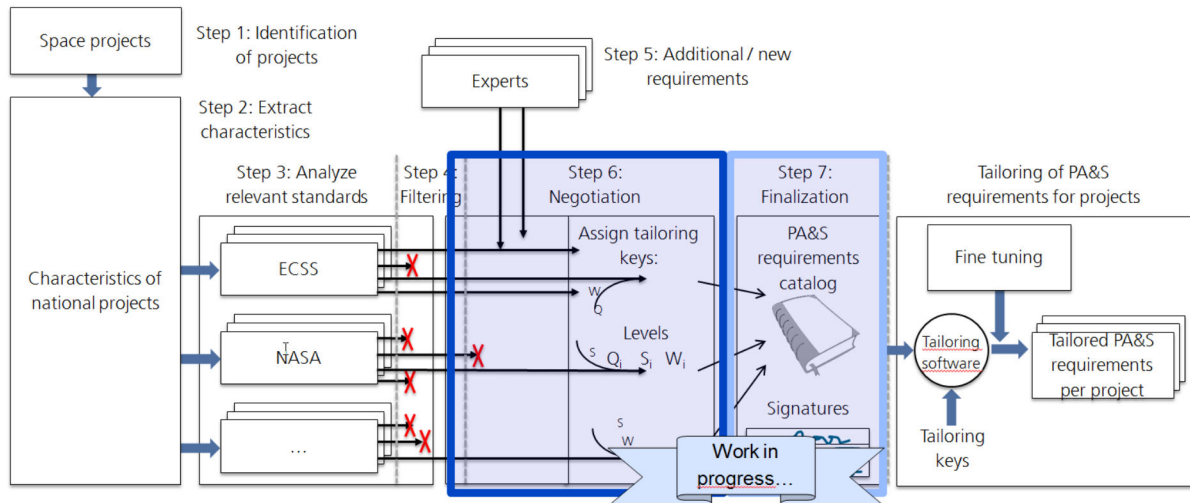


Figure 7: Conception and integration of requirements for agile development into product assurance and safety requirements into the requirements tailoring catalog (Prause et al., 2016)

For example, one could choose a sprint-driven lifecycle with formalisms. In this model, delta reviews of the milestone reviews could be performed after each iteration/sprint (see Figure 6).

5. National Space and DLR Product Assurance Requirements

Over the years, the Product Assurance (PS) department of DLR Space Management has repeatedly invited the heads of the quality departments of the major national space companies to working groups on the subject of "Product Assurance & Safety Requirements for DLR Space Projects". Last year, quality experts from important partners in the National Programme for Space and Innovation met again to examine all requirements in the ECSS Q standards, and decide which ones to carry over to the national requirements catalog (Figure 7, step 3). Also, it may be necessary to include further requirements, where, from a national perspective, ECSS has gaps. Once the catalog is approved by the heads of the national agency and the major industries, adapted requirements are incorporated into the respective project contracts by means of tailoring. To date, the working group has largely completed its work but fine-tuning is still taking place, e.g. the definition of tailoring keys. These keys serve later to define the requirements applicable to the projects in a semi-automated way. (cf. Prause, Bibus, Dietrich, & Jobi, 2016)

In the course of the working group, the current state of the art for all disciplines is reviewed and evaluated for applicability in the national programme. If this results in the need to apply more stringent demands on product development and manufacturing processes, corresponding requirements are integrated into the catalogue (Figure, Step 5). Among the new topics that the expert working group has classified as important for the national space program is agile software development. Specific requirements have been discussed, defined and included in the catalogue.

The majority of requirements in the catalogue of requirements originate from ECSS Q-standards. Regarding agile development, however, the situation is somewhat different: As the handbook's code "ECSS-E-HB-40-01A" suggests, it belongs to the E-branch of ECSS. Although topics of product safety and security are given their place in the handbook, it is still subordinate to ECSS-E-ST-40C, and hence not a Q standard in the narrower sense. Furthermore, the handbook is an "ECSS handbook" ("HB" in its code), i.e. explanations and practical recommendations are given, but no normative requirements.

As a result, the group of experts first had to develop suitable requirements for agile development in the catalogue from scratch. Since so far there is little experience with agile development in projects of the national programme, it is unclear whether there may be points of friction with the traditional processes. The respective requirements therefore generally formulate the proviso that an agile life cycle is already contractually agreed in the project.

6. Future Developments and Open Questions

In 2015, the Technical Authority, i.e. the governing body of the ECSS, made the conscious decision to include agile software development only at handbook level, i.e., subordinate to the existing standards and regulations. The rationale was to not possibly counteract established and proven requirements of the two software standards ECSS-E-ST-40C and ECSS-Q-ST-80C. Too many critical questions regarding agile development for space projects are still open:

A system such as a satellite or a probe must be able to survive autonomously in space for many years under harsh conditions, without anyone being able to intervene or press the reset button if necessary. One would hardly dare to install a partially finished software with a version number less than 1.0 on such a system. However, it may be possible to continue development after the first fully functional software (e.g. the SpaceX Falcon rockets) has been completed. For development of less critical but more software-intensive systems like information and planning systems on the ground, agile development could certainly play out its strengths better. The ECSS-E-ST-40C and ECSS-Q-ST-80C standards are designed and claim to cover space and ground systems alike. In the coming years, it might be necessary to renegotiate the roles of traditional and agile development; perhaps also in the direction of hybrid process models (cf. Kuhrmann et al., 2018). In practice, there may be a tendency to use more classical process models for highly critical applications, and agile development for less critical applications (Klünder et al., 2018). However, other issues related to agile development also raise questions for institutional spaceflight. For example:

- How can invitations to tender be organized? They are typically based on elaborately pre-specified descriptions of work.
- Iterative procedures with customer participation often already represent a partial acceptance in the legal sense. This has an impact on the issues of responsibility and liability, as responsibility is transferred from the contractor to the client (cf. Gennen, 2016 and Horn, Frischauf, Baumann, & Heinrich, 2016).
- Not least, agile development increases the communication effort between customer and supplier, which might cause a corresponding additional effort on the part of the institutional client. This may run against goals of a lean public administration.
- The use of classic contract models is probably not possible either but rather "agile contracts with a strong individualisation are to be drawn up" (translated from Horn, Frischauf, Baumann, & Heinrich, 2016).
- Agile development may pose a particular challenge for stable and bureaucratic organizations (Lappi & Aaltonen, 2017).

The tension in the European space community also showed itself at the "ESA SW Product Assurance and Engineering Workshop 2017". There were several presentations on agile development. In his pointed closing talk, the head of the software product assurance department at ESA stressed that agile development sounds sexy, that people want to be part of the game, but that "hope" is not a new product assurance method. It may be that the new role of the Scrum Master is just a new name for the same old functions. He asked one of the speakers on agile development, who was present in

the audience, if he would start a company that would develop flight software in an agile way. The speaker answered “no”. He continued to ask others similar questions, and kept getting analogous responses. Finally, he asked one of speakers, who was the representative of a large space company, whether he as prime contractor would subcontract a small, agile company that could not produce a successful maturity assessment. The respondent affirmed and substantiated: Since the company was following the agile principles, he knew that it would be doing continuous process improvement.

7. Conclusion

This paper motivates agile development in institutional European space flight. In particular, two artefacts – the new ECSS handbook on agile development and a chapter on agile software development in the national catalogue of requirements for product assurance and safety – are presented. With these two artefacts, institutional European spaceflight is taking a clear step towards NewSpace and agile development. However, some questions remain unanswered. In order to be able to answer them satisfactorily in the future and to further develop the project frameworks, engineers will now have the opportunity to try out agile software development and gain experience over the next years. Thus, the transformation of software development in spaceflight itself can follow the agile principles of iterations and small steps.

References

Brüggemann, S. & Prause, CR. (2018). Status Quo Agiler Software-Entwicklung in der Europäischen Institutionellen Raumfahrt. Deutscher Luft- und Raumfahrtkongress (DLRK), 2018

Bund, K., Heuser, U. J., & Tatje, C. (2012). Super-Männchen statt Alpha-Tiere. WirtschaftsWoche (online).

Buzgan, A. (September 2017). Methods of Harmonizing Agile Software Development with ECSS (presentation). ESA SW Product Assurance and Engineering Workshop. Darmstadt.

ECSS. (2012). Release Note for ECSS standards. Noordwijk, The Netherlands: ESA-ESTEC.

ECSS-E-HB-40-01A (2018). Agile Software Development (Draft). Noordwijk, The Netherlands: ECSS Secretariat, ESA.

ECSS-E-HB-40A (2013). Space engineering – Software engineering handbook. Noordwijk, The Netherlands: ECSS Secretariat. ESA.

ECSS-E-ST-40C (2009). Space engineering - Software. Noordwijk, The Netherlands: ECSS Secretariat, ESA.

ECSS-Q-ST-80C (2009). Space product assurance - Software product assurance. Noordwijk, The Netherlands: ECSS Secretariat, ESA.

Gennen, K. (2016). Auswirkungen hybrider Projektvorgehensmethoden auf den Softwareerstellungsvorgang. In M. Engstler, M. Fazal-Baqaie, E. Hanser, O. Linssen, M. Mikusz, & A. Volland, Projektmanagement und Vorgehensmodelle (S. 37-48). Bonn: Gesellschaft für Informatik e.V.

Horn, R., Frischauf, N., Baumann, I., & Heinrich, O. (2016). NewSpace: Geschäftsmodelle an der Schnittstelle von Raumfahrt und digitaler Wirtschaft. Bundesministerium für Wirtschaft und Energie.

Klünder, J., Hohl, P., Küpper, S., Krusche, S., Lous, P., Fazal-Baqaie, M., et al. (2018). Towards Understanding the Motivation of German Organizations to Apply Certain Software Development Methods. In International Conference on Product-Focused Software Process Improvement (PROFES). Springer.

Kreuzer, B., & Haader, M. (Januar 2017). Zukunft im Fokus. Audi Magazin, S. 24-27. Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Trektere, K., McCaffery, F., et al. (Jan 2018, Early Access). Hybrid Software Development

Approaches in Practice: A European Perspective. IEEE Software. Laplante, P. A. (2012). Real-Time Systems Design and Analysis. Hoboken, New Jersey: John Wiley & Sons.

Lappi, T., & Aaltonen, K. (2017). Project governance in public sector agile software projects. International Journal of Managing Projects in Business, S. 263-294.

Pernot, P. (2013). La Horde Agile. «Qui ne transitionne pas se fossilise.», haiku préhistorique. Online: <https://pablopernot.fr/pdf/lahordeagile.pdf>.

Prause, C., Bibus, M., Dietrich, C., & Jobi, W. (2016). Managing Software Process Evolution for Spacecraft from a Customer's Perspective. In M. Kuhrmann, J. Münch, I. Richardson, A. Rausch, & Z. H., Managing Software Process Evolution. Springer.

Schiller, D., & Heinemann, J. (2014). 20 Years of Collaboration for European Spaceflight. DLR Newsletter Countdown, 24.

APPENDIX Original Publication: Status Quo Agiler Software-Entwicklung in der Europäischen Raumfahrt

STATUS QUO AGILER SOFTWARE-ENTWICKLUNG IN DER EUROPÄISCHEN INSTITUTIONELLEN RAUMFAHRT

Dr. Stefan Brüggemann, Airbus DS GmbH, Airbus-Allee 1, 28199 Bremen, Deutschland

Dr. Christian Prause, DLR Raumfahrtmanagement, 53227 Bonn, Deutschland

ZUSAMMENFASSUNG

Die institutionelle Raumfahrt nimmt zunehmend die Bedeutung agiler Entwicklungsmethoden wahr. Dies geht einher mit einer Veränderung hin zu NewSpace Produkten und -Technologien. Hier sind insbesondere kurze Entwicklungszyklen und Flexibilität wichtig. Zusammen mit Industriepartnern verabschieden sowohl die europäische (ESA) als auch die deutsche Raumfahrtagentur (DLR) derzeit zwei Artefakte, welche einen Grundstein für agile Entwicklung in der institutionellen Raumfahrt legen werden: (1) Ein ECSS Standard (Handbuch) zu agiler Softwareentwicklung hat das öffentliche Review passiert und befindet sich nun in der Fertigstellung, und (2) nimmt das Raumfahrtmanagement des DLR erstmals einen Satz von Anforderungen für Lieferanten in seine Produktsicherungsanforderungen auf, welche auf agile Entwicklung ausgerichtet sind. Die hier vorliegende Zusammenfassung eines Vortrags auf dem diesjährigen Deutschen Luft- und Raumfahrt Kongress beschreibt die Eigenschaften agiler Entwicklung, stellt die zwei Artefakte vor und zeigt damit auf, wie sich agile Entwicklung in die Rahmenwerke institutioneller Raumfahrt einbinden lassen kann. Die Raumfahrt-Community nimmt das Thema sowohl mit Interesse als auch kritischen Fragen zu möglichen Folgen (z.B. Qualität, Vertragsgestaltung) auf.

KEYWORDS

Raumfahrt, Agile Softwareentwicklung, Software Engineering, NewSpace, ECSS, ECSS-E-HB-40-01A, Raumfahrtagenturen, Produktsicherung, Qualität

1. EINLEITUNG

Die institutionelle europäische Raumfahrt nimmt zunehmend die Bedeutung agiler Entwicklungsmethoden wahr. Hierbei geht es darum, rasch mit Produkten und Diensten am Markt zu sein, Feedback zu diesen einzuholen und die Produkte zu optimieren. Dies geht einher mit einer Veränderung des Produktportfolios hin zu NewSpace Produkten und -Technologien. Dabei ist es insbesondere wichtig, kurze Entwicklungszyklen zu haben, in denen genügend Flexibilität vorhanden ist, um Kundenanforderungen und sich verändernde Märkte adäquat berücksichtigen zu können (Horn, Frischauf, Baumann, & Heinrich, 2016). Im vergangenen Jahr sind in diesem Zusammenhang zwei wesentliche Artefakte entstanden, welche diese Strömungen bei DLR und ESA bündeln:

Im Rahmen der ECSS (European Cooperation for Space Standardisation) ist eine Arbeitsgruppe (ECSS-E-HB-40-01A, 2018) mit industriellen und institutionellen Vertretern einberufen worden. Diese verfolgt das Ziel, zu beschreiben, wie sich agile Softwareentwicklung

gestaltet, wenn klassische etablierte ECSS-Standards anzuwenden sind. Das Handbuch hat 2017 das öffentliche Review der Raumfahrt-Community durchlaufen, und die daraus stammenden Kommentare werden derzeit in die finale Version des Handbuchs eingearbeitet.

Unter der Führung der deutschen Raumfahrtagentur treffen sich immer wieder die Leiter der Qualitätsabteilungen der wichtigsten deutschen Raumfahrtunternehmen zu einem Arbeitskreis zu den Themen der Produktsicherung. Ziel ist es, die allgemeinen Qualitätsanforderungen – also unabhängig vom einzelnen Projekt – festzulegen, die später in den nationalen Projekten durch den Auftraggeber DLR Raumfahrtmanagement gefordert werden sollen. Im vergangenen Jahr haben sich so die wichtigsten „Qualitätser“ des Nationalen Programms für Weltraum und Innovation wieder zusammengesetzt. Da dabei für alle technischen Disziplinen (sowohl Hardware wie Software) der aktuelle Stand der Technik überprüft und in den Katalog integriert wird, ist auch die Methode der agilen Softwareentwicklung betrachtet und als wichtig für das nationale Raumfahrtprogramm eingestuft worden. Daher

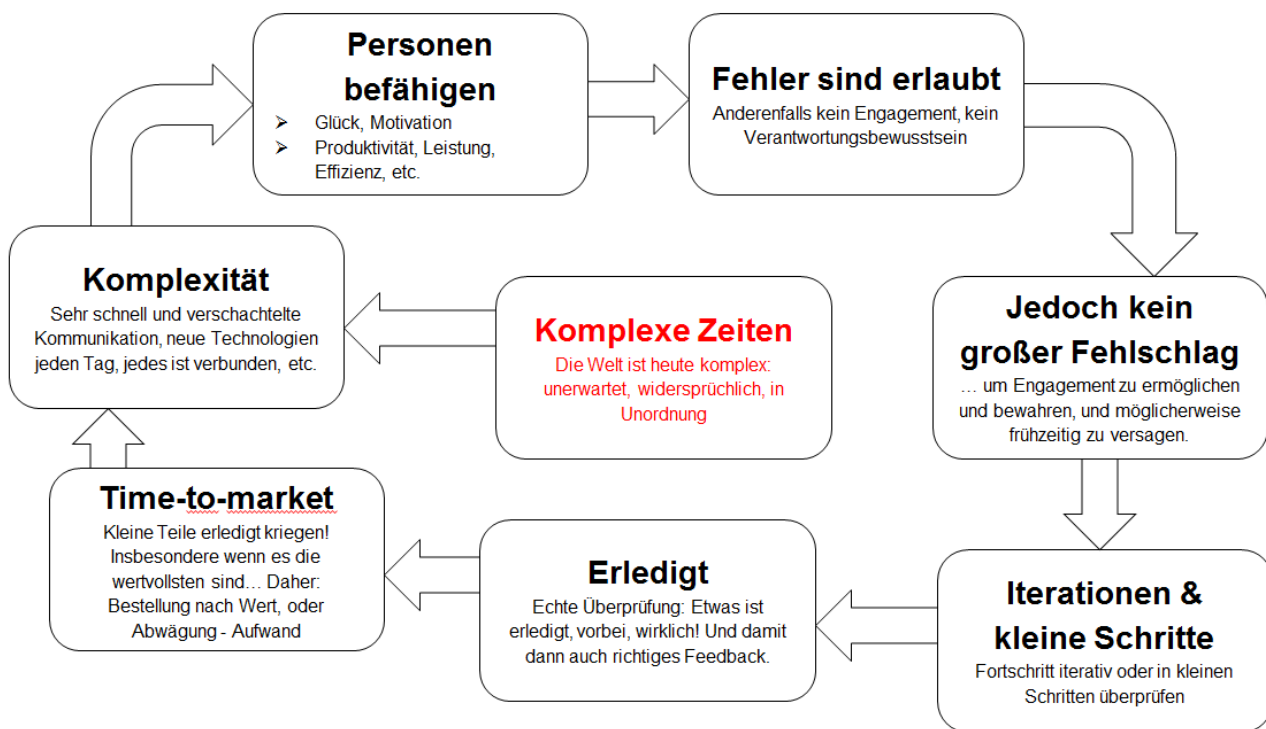


Abbildung 1: Agile Entwicklung in komplexen Zeiten (Pernot, 2013)

wurden entsprechende Anforderungen an agile Entwicklung in Projekten definiert und in den Katalog aufgenommen.

Das vorliegende Papier stellt Motivation (Abschnitt 2) und Eigenschaften (Abschnitt 3) agiler Entwicklungsmethoden dar, und zeigt, wie agile Entwicklung durch das neue ECSS Dokument (Abschnitt 4) bzw. durch die Produktsicherungsanforderungen der deutschen Raumfahrtagentur in die europäische institutionelle Raumfahrt einzieht (Abschnitt 5). Schließlich werden einige offene Fragen im Bezug zu agiler Entwicklung in der Raumfahrt kritisch betrachtet (Abschnitt 6).

2. WARUM AGILE ENTWICKLUNG?

Die Welt wird zunehmend komplexer und dadurch weniger vorhersagbar als früher. Das zeigt sich unter anderem auch an den neuen Technologien. Aber nicht nur hier: In allen Bereichen bis hin zu vermeintlichen Alltäglichkeiten steigt die Komplexität. Darüber hinaus arbeiten viele Unternehmen nicht mehr nur lokal, sondern europa- und weltweit. Um neue Märkte zu erschließen stehen sie dabei neuen Aufgaben und neuen Risiken gegenüber. In der institutionellen Raumfahrt sind traditionell viele verschiedene Parteien, oft aus unterschiedlichen Ländern, an Projekten beteiligt: Raumfahrtagenturen, Auftraggeber, Auftragnehmer, diverse Ebenen von Unterauftragnehmern, Betriebsorganisationen, Endnutzer wie z.B. Wissenschaftler, Besteller von Technologien, Überwachungsstellen, etc.

Durch die modernen Kommunikationswerkzeuge steigen die Geschwindigkeit, Reichweite und Vernetzung in der Kommunikation. Insbesondere gilt gerade Software gilt als „Schwamm“ für die Komplexität in technischen Dingen (ECSS-E-HB-40A, 2013).

Unter diesen Rahmenbedingungen ändert sich auch die Rolle der Führungskräfte: Sie können nicht mehr alles alleine bewältigen, also an sich reißen und allein entscheiden. Ihre Aufgabe wird vielmehr zu dienen, Rahmenbedingungen zu setzen, die Organisation in die Lage zu versetzen, das Richtige zu tun, zu coachen und anzuleiten. Sie befähigen Personen, ermöglichen Motivation und steigern dadurch Leistung, Produktivität und Effizienz. Fehler sind erlaubt, um den Weg für Engagement und Verantwortungsbewusstsein frei zu machen. Hier setzen NewSpace-Projekte und konkret im Bereich der Software, die agile Entwicklung an. Doch wie in klassischen Projekten auch, müssen jedoch große Fehlschläge vermieden werden. Daher werden Projekte iterativ und in kleinen Schritten vorangetrieben. Die einzelnen Schritte können erledigt und abgeschlossen werden; und sind damit tatsächlich vorbei. Hier erfolgt wirkliches Feedback. Die kleinen abgeschlossenen Einheiten stellen für sich schon einen Mehrwert dar; insbesondere dann, wenn die wertigsten Einheiten zuerst abgeschlossen werden. Dadurch verkürzt sich die Zeit bis zur Markteinführung. Siehe dazu Abbildung 1, wie auch Bund, Heuser, & Tatje (2012); Horn, Frischauf, Baumann, & Heinrich (2016); Kreuzer & Haader (2017) und Pernot (2013).

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions	over	Processes and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

“That is, while there is value on the right, we value the items on the left more.”

Abbildung 2: Das Agile Manifest.

3. WAS IST AGILE ENTWICKLUNG?

Der Begriff „agile Softwareentwicklung“ geht auf das Agile Manifest (siehe Abbildung 2) zurück, und bezeichnet jene Methoden und Praktiken der Softwareentwicklung, die dessen Werte und Prinzipien aufgreifen. Das Manifest wurde 2001 von 17 Erstunterzeichnern formuliert (darunter bekannte Softwareentwickler wie Kent Beck, Alistair Cockburn, Ward Cunningham, Martin Fowler oder Ron Jeffries.)

Zu den agilen Prinzipien zählen Kundenzufriedenheit, Auslieferung neuer Versionen funktionsfähiger Software in kurzen Zeitabständen, beinahe tägliche Zusammenarbeit zwischen Fachexperten und Entwicklern, Schaffung eines Motivation-fördernden Umfeldes, Einhaltung eines gleichmäßigen Arbeitstempos, technische Exzellenz und gutes Design, Selbstorganisation der Teams und Selbstreflexion (Laplante, 2012). Die Prinzipien werden durch Agile Methoden wie Paarprogrammierung, Testgetriebene Entwicklung oder ständige Refaktorisierungen unterstützt und in agilen Prozessen wie Extreme Programming, Feature Driven Development oder Scrum umgesetzt. In den vergangenen zwei Jahrzehnten hat die agile Softwareentwicklung sich dabei in immer mehr Bereichen als Mainstream durchgesetzt (vgl. Kuhrmann, et al., 2018).

Im Rahmen ihrer Untersuchung zu NewSpace gelangen auch Horn, Frischauf, Baumann, & Heinrich (2016), zu dem Ergebnis, dass agile Softwareentwicklung ein Ansatz sei, „der zunehmend auf die Raumfahrt übertragen wird. Bei dieser Vorgehensweisen wird die Entwicklung nicht im Voraus in allen Einzelheiten geplant und anschließend durchgeführt, stattdessen

wird die Entwicklung zyklisch iteriert. Dies ermöglicht es, Anforderungen während der Projektlaufzeit zu ändern und laufend Anpassungen vorzunehmen.“

4. ECSS AGILE SOFTWARE DEVELOPMENT HANDBOOK

Die ECSS wurde 1993 als eigenständiger Zusammenschluss von ESA, nationalen europäischen Raumfahrtagenturen, und dem Interessenverband der Industrie Eurospace gegründet. So entstanden gemeinsame Standards für Projektmanagement, Entwicklung, Konstruktion und Betrieb von Raumfahrtssystemen. Die Mitglieder von ECSS sind zugleich die Entwickler und Anwender der Standards. Dabei konzentrieren sie sich auf Projektarbeit und -prozesse: auf Entwicklung, Produktion und Betrieb technischer Systeme im Weltraum wie Satelliten oder Raumsonden und am Boden wie Sende-, Empfangs- und Kontrollzentren. Die ECSS gliedert sich in die Themenbereiche Space Project Management (Standards im ECSS M-Branch), Space Engineering (E-Branch), Space Product Assurance (Q-Branch) und Space Sustainability (U-Branch) (Schiller & Heinemann, 2014). Dabei gelten für die Softwareentwicklung neben den zwei Hauptstandards Software Engineering ECSS-E-ST-40C (im E-Branch) bzw. Software Product Assurance ECSS-Q-ST-80C (im Q-Branch) zusätzlich diverse allgemeinere Standards wie ECSS-M-ST-40C (Configuration Management) oder ECSS-Q-ST-30C (Dependability). Das Handbuch zu agiler Softwareentwicklung ECSS-E-HB-40-01A ist dem Hauptstandard ECSS-E-ST-40C zugeordnet (siehe Abbildung 3).

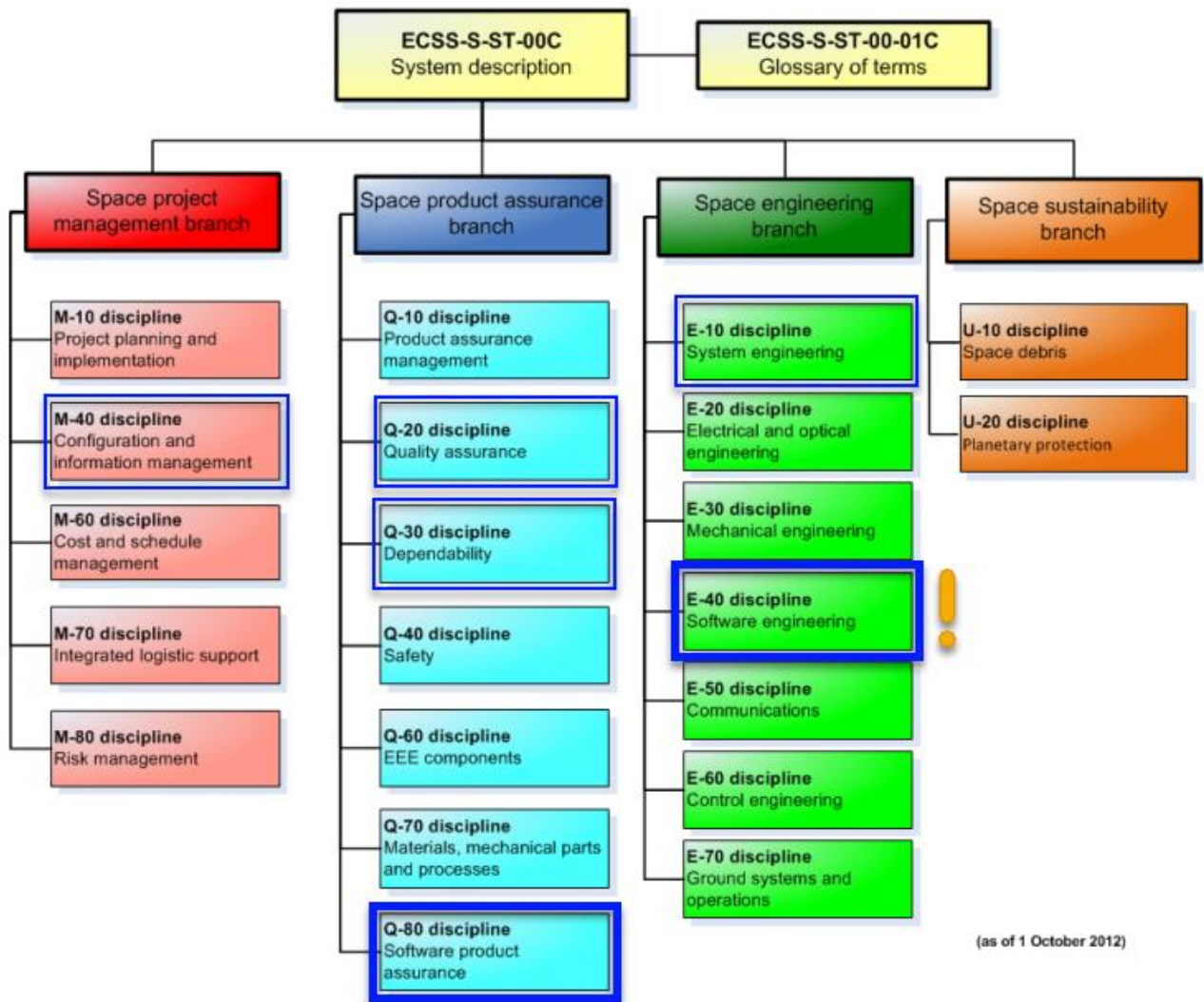


Abbildung 3: Übersicht über die Hauptstandards der ECSS (2012)

Die Arbeitsgruppe zur Erarbeitung des neuen ECSS-Handbuchs setzt sich zusammen aus industriellen und institutionellen Vertretern von ESA, DLR, CNES, GMV, Thales Alenia Space, Airbus DS, Telespazio, Intecs und SpaceBel. Das Handbuch hat das öffentliche Review der europäischen Raumfahrt-Community vom April 2017 bis Januar 2018 durchlaufen, und die daraus stammenden Kommentare werden derzeit in die finale Version des Handbuchs eingearbeitet.

Das Ziel des Handbuchs ist es, zu beschreiben, wie sich agile Softwareentwicklung gestaltet, wenn klassische etablierte ECSS-Standards anzuwenden sind. Im Handbuch wird ein Leitfaden zur Anwendung von agilen Entwicklungsansätzen unter Beachtung der ECSS-E-ST-40C und ECSS-Q-ST-80C dargestellt. Es werden alle Bereiche behandelt, die von diesen zwei Standards adressiert werden, also u.a. Flugsoftware, Bodensoftware und Simulatoren. Nicht behandelt werden dagegen z.B. Themen des Risiko-basierten Tailorings der ECSS Standards, oder vertragliche Aspekte dieses be-

sonderen Entwicklungsansatzes (vgl. ECSS-E-HB-40-01A, 2018).

Konkret ist ein Handbuch entstanden, welches zuerst eine Einführung in die agile Softwareentwicklung gibt und dabei Themen wie Charakteristika agiler Entwicklung, Lean Management, und agile Methoden vorstellt. Auch generelle Überlegungen und Fragen die sich vor der Implementierung von agilen Methoden stellen, werden in einem Kapitel adressiert. Im nächsten Kapitel schließen sich Richtlinien für die Auswahl eines agilen Lebenszyklusmodells an. So werden zum Beispiel Schlüsselfaktoren des Kunden-, Lieferanten-, Projekt- und Team-Kontextes besprochen. Der Kernteil des Dokumentes stellt den Scrum-Prozess vor und wie dieser auf Raumfahrtprojekt-Anforderungen angepasst werden kann. Der Scrum-Prozess ist einer der am weitesten verbreiteten agilen Methoden. Nach der Einführung wird für alle zentralen Disziplinen wie Projektmanagement, Software-Entwicklung, Konfigurationsmanagement und Qualitätsmanagement detailliert be-

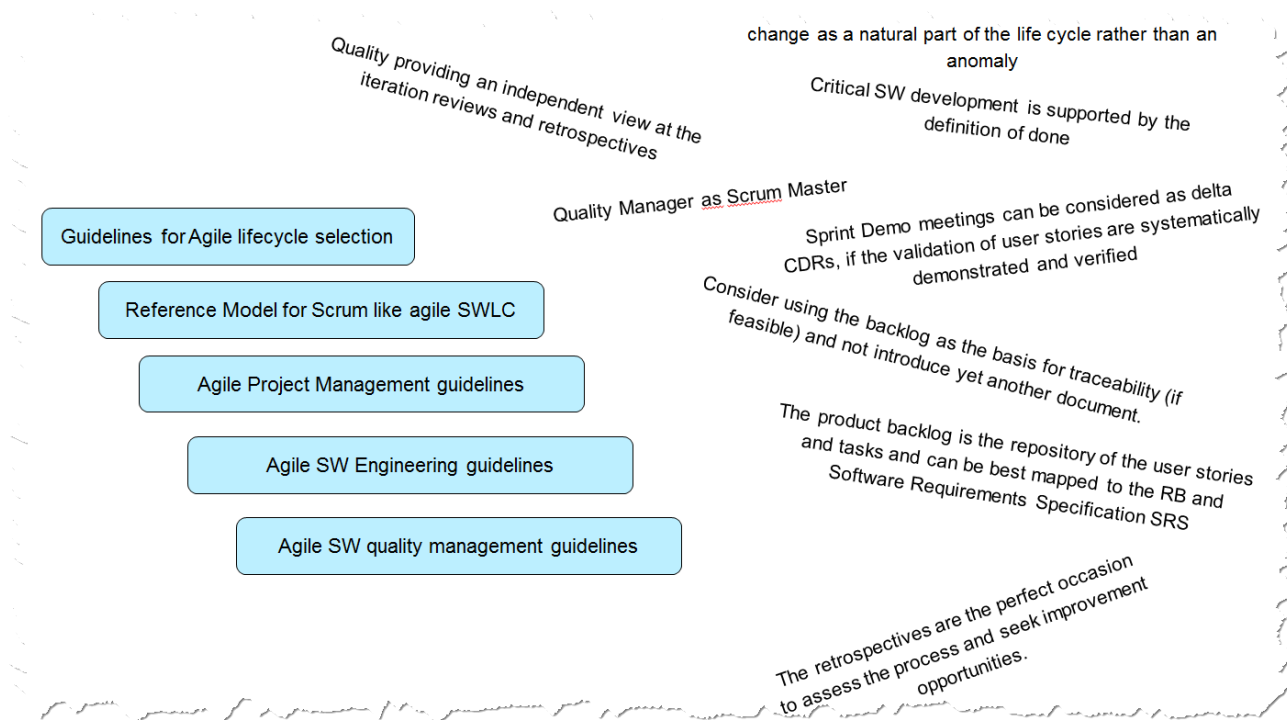


Abbildung 4: Themen des und Zitate dem Handbuch ECSS-E-HB-40-01A

schrieben, wie diese in einem agilen Projektszenario gestaltet werden können. Das Handbuch schließt ab mit verschiedenen Modellen, wie sich klassische ECSS-Reviews mit agilen Methoden vereinbaren lassen.

4.1 Beispiel: ECSS Reviews mit agilem Entwicklungsansatz

Zur Umsetzung der agilen Entwicklung in ECSS Projekten wurden vier verschiedene Modelle erarbeitet (siehe Abbildung 5): Es gibt Review-getriebene Software-

Lebenszyklen, die näher an der ECSS sind, und Sprint-getriebene Software-Lebenszyklen, die mehr einem Scrum-Vorgehen ähneln. Anhand einer Tabelle im Handbuch können entscheidende Projektcharakteristika mit Empfehlungen für oder gegen das eine oder andere der vier Modelle abgeglichen werden.

Dementsprechend könnte z.B. ein Sprint-getriebener Lebenszyklus mit Formalismus gewählt werden. So können in einem Modell beispielsweise nach jeder Iteration/Sprint Delta-Reviews zu den Meilenstein-Reviews durchgeführt werden (siehe Abbildung 6).

Project characteristics	model 1	model 2	model 3	model 4
Strong coordination with higher level team via reviews	Green	Green	Yellow	Yellow
Strong coordination with higher level team via agile	Red	Red	Green	Green
Formal endorsement required	Green	Yellow	Green	Yellow
Requirements need to be flexible	Red	Red	Yellow	Green

Abbildung 5: Projektcharakteristika und Lebenszyklusmodelle. Quelle: ECSS-E-HB-40-01A (2018)

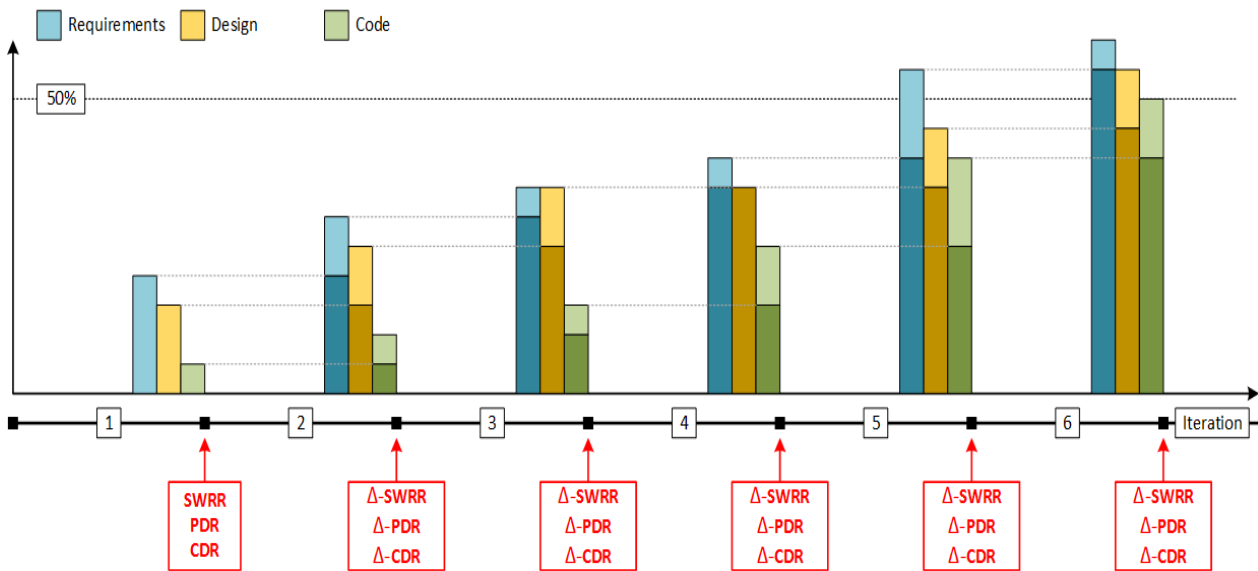


Abbildung 6: Scrum-Sprints zwischen den Reviews (ECSS, Modell 4). Quelle: Buzgan (2017)

5. NATIONALES RAUMFAHRTPROGRAMM UND DLR PS-ANFORDERUNGEN

Die Produktsicherung (PS) des DLR Raumfahrtmanagements hat in den vergangenen Jahren immer wieder die Leiter der Qualitätsabteilungen der großen nationalen Raumfahrtunternehmen zu Arbeitskreisen zum Thema "Product Assurance & Safety Requirements for DLR Space Projects" eingeladen. Auch im vergangenen Jahr haben sich so die wichtigsten „Qualitätser“ des Nationalen Programms für Weltraum und Innovation wieder zusammengesetzt, um gemeinsam zu beraten und festzulegen, welche Anforderungen der ECSS Q-

Standards für das nationale Programm besondere Bedeutung haben (Abbildung 7, Schritt 3) bzw. wo die ECSS Lücken aufweist, die für nationale Vorhaben geschlossen werden sollten. Die vereinbarten Qualitätsanforderungen werden in einem umfassenden Anforderungskatalog festgehalten. Aus dem freigegebenen Katalog heraus werden an die Projekte individuell angepasste Forderungen mittels Tailoring in die jeweiligen Projektverträge übernommen (Prause, Bibus, Dietrich, & Jobi, 2016). Die Arbeitsgruppe hat ihre Arbeit weitgehend abgeschlossen, jedoch finden derzeit noch Feinabstimmungen statt, wie z.B. die Festlegung von Tailoringsschlüsseln. Diese Schlüssel dienen später dazu, die für die Projekte zutreffenden Anforderungen teil-automatisiert festzulegen.

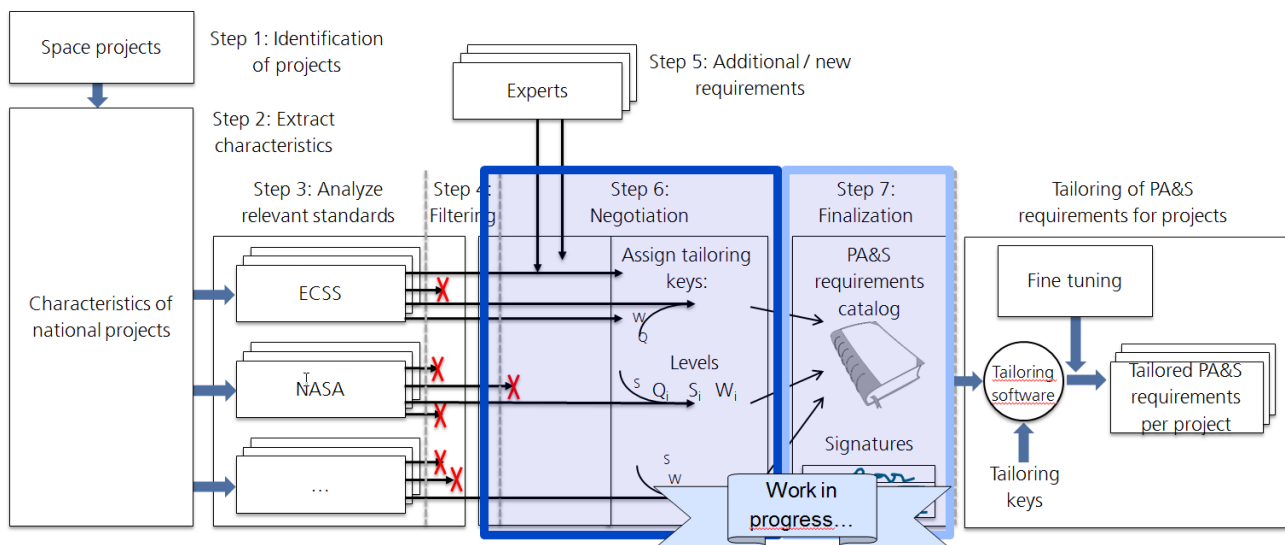


Abbildung 7: Erarbeitung und Übernahme der Anforderungen zu agiler Softwareentwicklung in die Produktsicherungsanforderungen und den DLR PA & Safety Requirements Tailoring Katalog

Im Zuge der Arbeitsgruppe wird für alle Disziplinen der aktuelle Stand der Technik überprüft und auf Anwendbarkeit im nationalen Programm hin bewertet. Ergibt sich daraus die Notwendig, weitergehende Anforderungen an Produkterstellungsprozesse zu stellen, werden entsprechende Formulierungen in den Katalog integriert (Abbildung, Schritt 5). Als ein Thema hat die Expertengruppe auch erstmalig agile Softwareentwicklung als wichtig für das nationale Raumfahrtprogramm eingestuft. So sind spezifische entsprechende Anforderungen an die agile Softwareentwicklung in Raumfahrtprojekten diskutiert, definiert, und in den Katalog aufgenommen worden.

Der Großteil der Anforderungen im Anforderungskatalog entstammt der ECSS. Bei der agilen Entwicklung ist die Situation jedoch etwas anders: Das neue ECSS Handbuch zu agiler Softwareentwicklung ist dem E-Branch der ECSS zugeordnet ist. Zwar erhält das Thema Produktsicherung und Safety seinen Platz in dem Handbuch, jedoch ist es insgesamt der ECSS-E-ST-40C zugeordnet. Ferner handelt es sich bei dem Handbuch um ein „ECSS Handbuch“, d.h. es werden Erläuterungen und praktische Empfehlungen ausgesprochen, jedoch keine normativen Anforderungen gestellt. Dadurch mussten in den Katalog passende Anforderungen, die die agile Entwicklung betreffen, erst von der Expertengruppe von Grund auf erarbeitet werden. Da bisher wenig Erfahrung mit der Einbindung von agiler Entwicklung in die Projekte des nationalen Programms besteht, und insbesondere unklar ist, ob es evtl. Reibungspunkte mit den althergebrachten Prozessen gibt, stehen die neuen Anforderungen bzgl. agiler Entwicklung grundsätzlich unter dem Vorbehalt, dass in den Projekten ein agiler Lebenszyklus vereinbart ist.

6. ZUKÜNFTIGE ENTWICKLUNG UND OFFENE FRAGEN

Die Technical Authority, also das Führungsgremium der ECSS, hatte sich 2015 bewusst dafür entschieden, der agilen Softwareentwicklung vorerst nur ein untergeordnetes Handbuch und keinen eigenständigen Standard einzuräumen, um nicht im Fall von Widersprüchen die Anforderungen der zwei Softwarestandards ECSS-E-ST-40C und ECSS-Q-ST-80C zu konterkarieren. Zu viele kritische Fragen sind noch offen:

Lässt sich eine eingebettete Satellitensoftware mit kritischen Funktionalitäten inkrementell ausliefern? Ein System wie ein Satellit oder eine Sonde, muss viele Jahre unter harschen Bedingungen im Weltraum autonom überleben können, ohne dass jemand bei Bedarf eingreifen oder den Reset-Schalter drücken kann. Hier würde man kaum eine teilfertige Software mit Versionsnummer kleiner 1.0 installieren wollen. Eine kontinuierliche Weiterentwicklung nach Erreichen des vollständigen notwendigen Funktionsumfangs mag hin-

gegen möglich sein (z.B. die SpaceX Falcon-Raketen). Auch in der Entwicklung weniger kritischer, aber Software-intensiverer Systeme wie Informations- und Planungssystemen am Boden kann agile Entwicklung sicher besser ihre Stärken ausspielen. Noch erheben die Standards ECSS-E-ST-40C und ECSS-Q-ST-80C den Anspruch, auch diese Bereiche vollständig abzudecken. Hier wird es in den kommenden Jahren womöglich eine Neuaushandlung der Rollen von traditioneller und agiler Entwicklung geben müssen; vielleicht auch in Richtung hybrider Vorgehensmodelle (vgl. Kuhrmann, et al., 2018). In der Praxis in deutschen Organisationen gibt es möglicherweise eine Tendenz, für hochkritische Anwendungsfälle eher klassische Vorgehensmodelle, und agile Entwicklung bei weniger kritischen Anwendungen einzusetzen (Klünder, et al., 2018).

Doch auch weitere Themen im Zusammenhang mit agiler Entwicklung werfen für die institutionelle Raumfahrt noch Fragen auf: Wie können öffentliche Ausschreibungen ausgestaltet werden, die typischerweise auf einen vorab vollständig spezifiziertem Leistungsgegenstand beruhen? Iteratives Vorgehen mit Kundenbeteiligung stellt im juristischen Sinn oft bereits eine Teilabnahme dar. Dies hat dann Auswirkungen auf die Themen der Verantwortung und Haftung, da hiermit Verantwortung vom Auftragnehmer auf den Auftraggeber übergeht (vgl. Gennen, 2016 und Horn, Frischauf, Baumann, & Heinrich, 2016). Nicht zuletzt bleibt zu nennen, dass agile Entwicklung den Kommunikationsaufwand zwischen Kunde und Lieferant erhöhen kann, was auf Seiten des institutionellen Kunden einen entsprechenden Mehraufwand verursachen würde. Dies läuft möglicherweise dem Ziel einer schlanken öffentlichen Verwaltung entgegen. Auch die Verwendung klassischer Vertragsmuster ist wahrscheinlich nicht möglich, sondern es sind „agile Verträge mit starker Individualisierung zu erstellen“ (Horn, Frischauf, Baumann, & Heinrich, 2016). Auch ansonsten stellt agile Entwicklung evtl. gerade für stabile und bürokratische Organisationen eine besondere Herausforderung dar (Lappi & Aaltonen, 2017).

Die Spannung in der europäischen Raumfahrt-Community zeigte sich auch auf dem „ESA SW Product Assurance and Engineering Workshop 2017“. Neben mehreren Beiträgen zu agiler Entwicklung, verwies der Leiter der Abteilung Softwareproduktsicherung bei der ESA in seiner pointierten Abschlussrede darauf, dass agile Entwicklung sexy klinge, die Leute Teil des Spiels sein wollten, jedoch „Hoffnung“ keine neue Produktsicherungsmethode sei. Vielleicht sei auch die neue Rolle des *Scrum Masters* nur ein neuer Name für die gleichen alten Funktionen. Der anwesende Referent zum Thema agiler Entwicklung, den er fragte, ob er eine Firma gründen würde, die Flugsoftware agil entwickeln würde, verneinte die Frage. Unter mehreren angesprochenen Zuhörern bejahte nur einer die

Frage, ob er als Hauptauftragnehmer (Prime) einer kleinen Firma ohne erfolgreiche Reifegradbewertung einen Unterauftrag geben würde. Dafür war letzterer jedoch umso überzeugter, da er ja wisse, dass jene Firma, den agilen Prinzipien folgend, kontinuierliche Prozessverbesserung betreibe.

7. ZUSAMMENFASSUNG

Das vorliegende Papier motiviert agile Entwicklung in der institutionellen europäischen Raumfahrt. Insbesondere werden zwei Artefakte – das neue ECSS Handbuch zu agiler Entwicklung und ein Kapitel zu agiler Softwareentwicklung im nationalen Anforderungskatalog für Produktsicherung und Safety – vorgestellt. Die institutionelle europäische Raumfahrt macht mit den zwei beschriebenen Artefakten einen deutlichen Schritt in Richtung NewSpace und agile Entwicklung. Doch es bleiben auch noch einige Fragen offen. Um diese in Zukunft zufriedenstellend beantworten zu können und die Rahmenwerke weiter zu entwickeln, haben Ingenieure in den Projekten und deren Führungen nun in den nächsten Jahren Gelegenheit, praktische Erfahrungen mit agiler Softwareentwicklung in der Raumfahrt zu sammeln. So kann die Transformation der Softwareentwicklung in der Raumfahrt selbst den agilen Prinzipien der Iterationen und kleinen Schritte folgen.

LITERATURVERZEICHNIS

- Bund, K., Heuser, U. J., & Tatje, C. (2012). Super-Männchen statt Alpha-Tiere. *WirtschaftsWoche (online)*.
- Buzgan, A. (September 2017). Methods of Harmonizing Agile Software Development with ECSS (Präsentation). *ESA SW Product Assurance and Engineering Workshop*. Darmstadt.
- ECSS. (2012). *Release Note for ECSS standards*. Noordwijk, The Netherlands: ESA-ESTEC.
- ECSS-E-HB-40-01A. (2018). *Agile Software Development (Entwurf, in finaler Überarbeitung)*. Noordwijk, The Netherlands: ECSS Secretariat, ESA.
- ECSS-E-HB-40A. (2013). *Space engineering - Software engineering handbook*. Noordwijk, The Netherlands: ECSS Secretariat, ESA.
- ECSS-E-ST-40C. (2009). *Space engineering - Software*. Noordwijk, The Netherlands: ECSS Secretariat, ESA.
- ECSS-Q-ST-80C. (2009). *Space product assurance - Software product assurance*. Noordwijk, The Netherlands: ECSS Secretariat, ESA.
- Gennen, K. (2016). Auswirkungen hybrider Projektvorgehensmethoden auf den Softwareerstellungsvorgang. In M. Engstler, M. Fazal-Baqaie, E. Hanser, O. Linssen, M. Mikusz, & A. Volland, *Projektmanagement und Vorgehensmodelle* (S. 37-48). Bonn: Gesellschaft für Informatik e.V.
- Horn, R., Frischauf, N., Baumann, I., & Heinrich, O. (2016). *NewSpace: Geschäftsmodelle an der Schnittstelle von Raumfahrt und digitaler Wirtschaft*. Bundesministerium für Wirtschaft und Energie.
- Klünder, J., Hohl, P., Küpper, S., Krusche, S., Lous, P., Fazal-Baqaie, M., et al. (2018). Towards Understanding the Motivation of German Organizations to Apply Certain Software Development Methods. In *International Conference on Product-Focused Software Process Improvement (PROFES)*. Springer.
- Kreuzer, B., & Haader, M. (Januar 2017). Zukunft im Fokus. *Audi Magazin*, S. 24-27.
- Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Trektore, K., McCaffery, F., et al. (Jan 2018, Early Access). Hybrid Software Development Approaches in Practice: A European Perspective. *IEEE Software*.
- Laplante, P. A. (2012). *Real-Time Systems Design and Analysis*. Hoboken, New Jersey: John Wiley & Sons.
- Lappi, T., & Aaltonen, K. (2017). Project governance in public sector agile software projects. *International Journal of Managing Projects in Business*, S. 263-294.
- Pernot, P. (2013). *La Horde Agile. «Qui ne transitionne pas se fossilise.»*, haiku préhistorique. Online: <https://pablopernot.fr/pdf/lahordeagile.pdf>.
- Prause, C., Bibus, M., Dietrich, C., & Jobi, W. (2016). Managing Software Process Evolution for Spacecraft from a Customer's Perspective. In M. Kuhrmann, J. Münch, I. Richardson, A. Rausch, & Z. H., *Managing Software Process Evolution*. Springer.
- Schiller, D., & Heinemann, J. (2014). ECSS - 20 Jahre Zusammenarbeit für die europäische Raumfahrt. *DLR Newsletter Countdown*.