TAF Agile Framework

Reducing Uncertainty within Minimum Time and Resources

Rafael Hostettler
Institute of Robotics & Embedded Systems
Technical University of Munich
Munich, Germany
rh@gi.ai

Annette Isabel Böhmer
Institute of Product Development
Technical University of Munich
Munich, Germany
boehmer@pe.mw.tum.de

Udo Lindemann
Institute of Product Development
Technical University of Munich
Munich, Germany
lindemann@pe.mw.tum.de

Alois Knoll
Institute of Robotics & Embedded Systems
Technical University of Munich
Munich, Germany
knoll@in.tum.de

***Abstract*— TAF Agile Framework addresses the challenge of providing a methods framework for agile mechatronic product development. It aims to reduce uncertainty towards the product within minimum time and resources by taking a holistic, interdisciplinary perspective on the product and iteratively increases knowledge by applying the scientific method to the domains of desirability, feasibility and viability. It crystallizes this knowledge by building prototypes and artifacts to disseminate it quickly throughout large teams. It integrates best practices from traditional mechatronic development (e.g. Munich Procedural Model) with methods inspired by lean startup, scrum and design thinking. It coordinates cross-functional teams and helps to identify critical functions in order to build the right things before building them right. TAF is applied wherever companies need to adapt to a changing market, where the current business model or rather product lost its viability and where mechatronic products are developed de-novo. In this study, TAF Agile Framework is deduced from the combined experience of 37 agile mechatronic development projects and evaluated with the results of 13 further projects during a two weeks lab course at TU Munich.**

***Keywords*— agile; product development; mechatronics; digital innovation; physical product development**

I. Introduction

Four primary forces are driving the demand for agility. First, highly volatile market environments call for adaptive firms that react quickly to shifting markets or rapid technological change. Second, in most markets the rate of change is increasing, thus reducing the effectiveness in forecasting future development with traditional management approaches [1].

Third, product complexity has increased significantly. As products acquire more functionality, it is harder to specify requirements a priori. Fourth, the haziness of reality, the potential for wrong predictions, and the mixed effects due to various conditions lead to cause-and-effect confusion.

Within recent years, the speed of development has become a decisive competitive factor. Mastering fast-paced markets and high innovation and technology dynamics is highly relevant in order to avoid unnecessary long innovation processes [1]. Both aspects lead to uncertainty in the planning and specification of future products. In this context, traditional (product-) developments methods have a limited effectiveness, wherefore the incorporation of new ones is required [2]. Especially for new product development, companies deal with high level of uncertainty. Besides market- and technology-related uncertainties, the degree of novelty is a significant source of uncertainty [3].

According to [3], many innovation projects fail due to late clarification of technical feasibility. The earlier one solve the challenge of a new product being realized in the existing plants at reasonable cost, the higher the degree of success.

Handling uncertainty is accomplished by the capability of being agile that goes hand in hand with ever-present change [4]. An agile innovation process, for example, has a shortened period between feature freeze and delivery. By incorporating new technologies as well as new user needs, just before product launch, change is seen as a chance for flexibility. Unfortunately, large companies often lack the adaptation and implementation speed needed to address this challenge [5].

Thus, there is a need for an easy to roll out and consistent agile development framework for mechatronic products that allows corporates to become agile.

II. Relation to Existing Theories and Work

Given the impact of demand uncertainty and dynamic marketplaces, it is essential to understand the potential of agile approaches within a mechatronic context. An effective innovation process is increasingly demanded for a better synchronization of available resources and market requirements. Currently, the challenge of innovation is driven

978-1-5386-0774-9/17/$31.00 ©2017 IEEE

by the short life cycle of today's technology. One must get technology to market more quickly, in order to successfully tackle this challenge. In order to do so, agile frameworks facilitate building the right products before building them right. This minimizes the risk of a wrong product being launched too late.

This work focuses on mechatronic product development by creating an agile framework that applies PDCA (Plan-Do-Check-Act, after Deming [22]) to the three domains desirability, feasibility and viability where knowledge is crystallized through prototypes and artifacts. Thus, we review the current background on these topics in the following sections.

A. Mechatronics

Mechatronics involves co-engineering of different disciplines, like mechanics, electronics and informatics [6], [7]. In the past, the overlap of the various disciplines was less than today or even in the future (see Fig. 1).

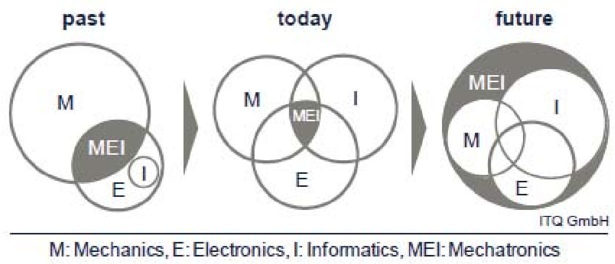

Fig. 1. Synergy of disciplines in mechatronics [10].

Especially the demand of highly individual products leads to a rising interaction as well as complexity for the corresponding engineering disciplines and processes [9]. Thus, the development of mechatronic products needs a close interaction of these three disciplines, which leads to a high level of integration [8].

In the past Systems Engineering has become more important for mechatronic engineering. However, these approaches are too generic and not sufficiently enough, to cope with the specific situations of multi-discipline engineering [11].

An established mechatronic development process cannot be adapted to agile procedures overnight. Instead, present conventional models should be supported by agile procedures in an appropriate way. Agility in product development can be located in correlation and influence on Flexible Product Development [2], Lean Development [12, 13] as well as Agile Manufacturing [14]. The main aspect is the early creation of artifacts and prototypes in order to test hypotheses.

B. PDCA

Conventional procedures often focus on predefined stage-gates and phases as well as a complete requirement specification [15]. Agile procedures follow a mind-set that embraces change and uncertainty throughout the whole process [17]. Thus, requirements are specified, features are realized and evaluated incrementally iteratively through the complete development [18-20]. Various agile procedures have been developed, but most of them are successfully applied within software [18]. Scrum is the most successful framework, that is also been applied for project management, besides pure software development [16, 21].

The micro-logic of Scrum is a continuous process, transferring the first draft to the final product while iterating phases of planning, doing, checking and acting, that is similar to the PDCA-Cycle of Deming [22].

C. Fuzzy Frontend / Feasibility, Desirability, Viability (FDV)

Both agile and innovation are based on the act of learning and how to adopt the current approach of implementation. Scrum within agile development can be seen as a thought-provoking approach to transform ad hoc ideation teams to use iterative techniques to drive better product development.

Agile frameworks can be focused on a variety of aspects of a final design, however, we focus on desirability, feasibility, and viability for initial implementation and validation. These three aspects have previously been used in research within Think.Make.Start. and engineering design to evaluate design concepts [23], prototypes [24, 25], and final products [26].

According to [27], we define three product increments to be verified: Minimum Desirable Product (MDP), Minimum Feasible Product (MFP) and Minimum Viable Product (MVP). For a maximum overlap of these three products the final prototype is achieved. Prototyping for desirability is about creating prototypes that test the purchase-ability and consumer value of a product or solution. Prototyping for feasibility is a practice of creating prototypes to test the technical functionality of the product idea. Viability prototyping approaches facilitate testing the design’s likelihood of fitting into time and budget constraints.

D. Systematic Agility

In general, "[a]gility is the capability to react, and adopt to expected and unexpected changes within a dynamic environment constantly and quickly; and to use those changes (if possible) as an advantage." [28]. Agile product development, brings that capability into the field of product development. It is very valuable for uncertain and volatile environments, in which initial circumstances and assumptions change easily or rather quickly.

Phase-oriented, multi-gate approaches being used particularly in traditional models have been established for a long time. They just need a low degree of organizational maturity to be established. Due to the domination of mechanics, engineers keep to these well-known procedures even when it comes to more mechatronic products. Agile procedures are not initiated because engineers struggle with the chaotic aftermath of wrongly applied agility [31]. This often implies the disappearance of familiar planning and control mechanisms as well as the apparently missing systematic documentation.

To overcome these challenges, traditional models should be supported by agile procedures in an appropriate way. A systematic agility, combining the advantages of both conventional and agile models, is pursued to make use of agile potential within mechatronics best (see Fig. 2) [32].

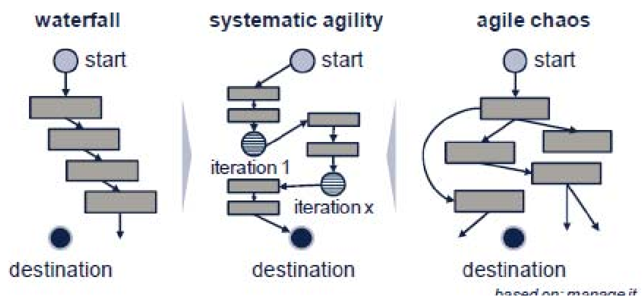

Fig. 2. Systematic Agility [32]

With regard to product architecture, a functional agility can be mentioned. Most products have a functional structure, existing of basic and optional parts. While the main functions are necessary for the product itself, sub-functions often represent customer's specific requirements. Agile approaches are especially helpful to define and find these sub-functions, having the involvement of the customer and fast time-to-market in mind [32].

E. Plan vs. Result driven

Both agile and conventional procedures may support the way of success in a hardware project according to several surveys [16, 21]. The benefits depend on the different home-grounds, in which rather agile or conventional procedures take place best [18, 33]. An agile process can be opposed to sequential, iterative and incremental procedures, using the classification of [34].

Agile combines both iterative and incremental aspects. They are mainly used in projects with high uncertainty and unclear requirements. Starting with an initial vision of the product, customer needs are gathered, structured and prioritized. Starting over with the requirement specification, a period of sketching, designing and prototyping sub-products follows. Insights gained are provided for a recapitulation of gathered requirements and product features. Whenever a prototype or rather a sub-product is realized, it is reviewed by the team or customer before the replanning starts over again.

III. RESEARCH/TECHNOLOGY/INNOVATION APPROACH

A. From four generations of Think.Make.Start. to TAF Agile Framework

Think.Make.Start. (TMS) is a two weeks curriculum for master students at TU Munich. Endowed with a 400€ budget, interdisciplinary teams of 4 to 6 students develop mechatronic products from ideation to minimal prototype. Throughout the course, they receive lectures on agile product development, such as lean startup, design thinking but also traditional product development methods. Furthermore, they receive pitch training as well as input on how to present their product to their stakeholders. The teams comprise of mechanical and electrical engineers, computer scientists, economists as well as "others" - ranging from designers over biologists to physicists. In each team, at least three faculties are present. The first iteration of Think.Make.Start. took place in 2015 and has since then repeated twice per year.

During the first four generations of Think.Make.Start. agile methods have been taught without a circumventing framework on when to apply which method. Rather a general build-measure-learn approach with hypothesis testing has been applied with a recommendation to use product development methods such as the morphological box or function trees.

Analyzing the use of tools as well as the feedback from the students, we found that while many prototypes have been built, their concrete purpose was often unclear. Rather, it seemed they were incrementally approaching their envisioned final product instead of using the prototypes as a method to gain knowledge over partial systems in order to adapt their product-solution fit. This lead in several cases to teams that tried to build complex modules before having figured out their basic components and subsequent failure in building a prototype that demonstrates the intended goal.

In one example, the goal was to build a robot that would be able to climb a pole. Rather than building three components, one to attach to and from the pole, one to move up and down a pole and one to move around the pole, the team tried to solve the moving module all at once and developed their own spherical wheel as a solution that in the end did not work.

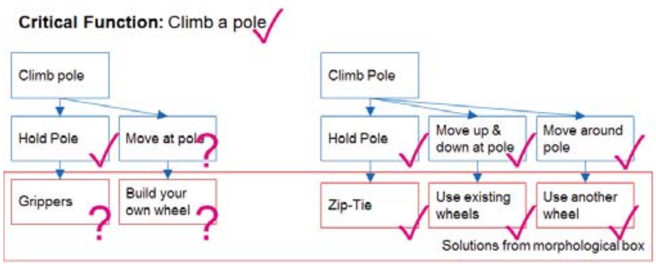

Fig. 3. Functional tree and planned implementation of a critical function. Left: as tried by the team, Right: as solving by component first would have suggested.

This finding is also reflected in a quantitative analysis conducted on the usage of methods during the fourth iteration of TMS.

Table. I

Artifact	**Iteration**							**Σ**
	2	**3**	**4**	**5**	**6**	**7**	**8**	
Desirability	***12***	***19***	***14***	***12***	***11***	***6***		***74***
Interviews	2	8	7	6	5	3		31
Hypotheses	2	8	7	5	5	1		28
Lean Canvas	7	3				1		11
User Story	1			1	1	1		4
Feasibility		***2***	***6***	***7***	***10***	***5***	***1***	***31***
Funct. model		1	2	2	2	1		8
Req. List			1		1	2		4
Building Strct.				1	4	1	1	7
Morph. Box		1	3	4	3	1		12

Artifacts generated during TMS #4 by the 10 Teams. More than twice as many artifacts focusing on desirability were generated than those focusing on feasibility. It can be seen, that desirability was tested before feasibility.

Methods that help the teams to identify such shortcomings like function tree and building structure were hardly used (See Table I). These results made clear that a stronger focus on "what to build" was required and thus a stronger focus on product development methods.

We started from the existing structure of the course, where in the first "Think."-phase a customer problem was to be identified - relating to desirability, subsequently in the "Make."-phase prototypes were built - relating to feasibility and ultimately a business model had to be found in the "Start."-Phase - relating to viability. Two shortcomings of the sequential approach were observed. First, the three phases are interdependent and often viability was checked too late as well as some projects while desirable failed to be feasible. And second, the teams required very different amounts of time to converge to a final desirable solution, often well into the third day, whereas other teams found their solutions to be desirable on the first day. Furthermore, desirability and feasibility prototypes have very different requirements and are thus mostly disjoint, especially in early phases where much of the functionality can be left to a user's imagination and feasibility prototypes give a wrong impression to potential users due to their focus on functionality rather than looks.

In consequence, the approach was to split the three different phases into independent desirability-feasibility-viability (FDV) cycles, so they can be iteratively and independently evaluated. This allows for different methods to be applied in each cycle type, giving a clear guideline on what to do next, without compromising on flexibility.

In order for a cycle to be valuable to the product, it must generate an outcome that increases the knowledge and thus reduces uncertainty. To this end, the scientific method, formulating an experiment with a dedicated learning goal is appropriate, as the gained information can be then communicated to the whole team and yields a stable finding to build upon in subsequent cycles. We implement this as a PLAN-DO-CHECK-ACT-cycle after Deming [22], where the ACT phase is the central synchronization point to integrate the findings of the different cycles and to decide on how to continue.

B. *TAF Agile Framework*

TAF Agile Framework[1] (see Fig. 4) starts from a product vision and an initial problem-solution idea. These can be generated through methods like design thinking. Its main goal is then to reduce uncertainty on all aspects of the product while maintaining as much flexibility as possible to react to changing boundary conditions.

It comprises three PLAN-DO-CHECK-ACT (PDCA) cycles, one for desirability, feasibility and viability each. The feasibility cycle depends on at least one completed desirability cycle to be able to derive functional requirements and the viability cycle depends on a completed feasibility cycle to be able to estimate costs. Apart from this, there is no

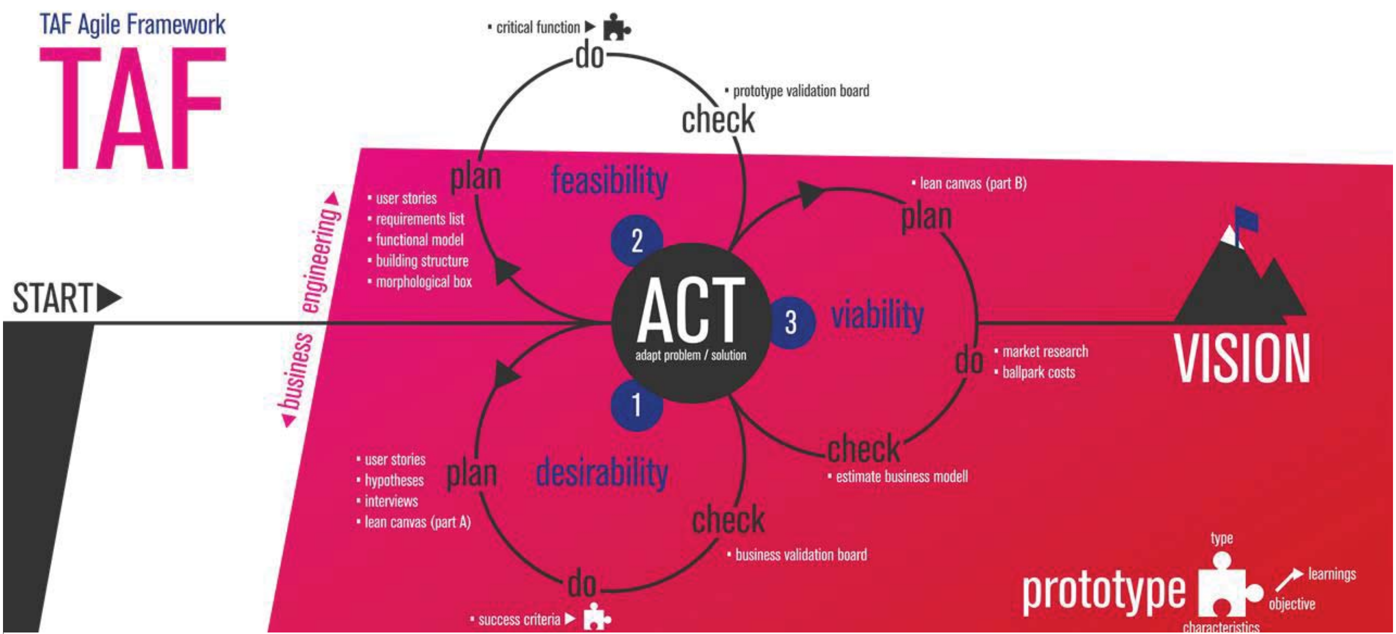

Fig. 4. TAF Agile Framework poster, displaying all important elements of TAF.

[1] TAF Agile Framework is a recursive acronym, where the T in TAF stands for TAF in TAF Agile Framework. This recursion is among the central features of TAF, as TAF can be applied to components individually. Additionally, it's an onomatopoetic equivalent to tough, hinting at the complexity of agile mechatronic product development.

interdependence and the cycles can be completed in arbitrary order and even in parallel.

Each cycle creates or modifies existing artifacts and prototypes that together represent the current joint knowledge of the team on the product.

In the following, we outline the different phases in the three cycles.

1) PLAN

The goal of the plan phase is to gather information and deduce a research question to be evaluated.

a) Desirability

In the desirability cycle, the research question is to evaluate a user story towards its validity. To that end, the problem description is translated into user stories, where the problem is formulated from the perspective of all relevant stakeholders. As these formulations are initially best guesses from the team, the underlying assumptions are extracted and turned into hypotheses. For each of the hypotheses an acceptance criteria is formulated that would, upon a quantitative test, serve to accept or reject the hypothesis.

b) Feasibility

In the plan phase of the feasibility cycle, the envisioned solution is split into technical user stories that reflect the solution on a coarse level. These are then used as an input to derive a functional tree, based on that a morphological box and building model are deduced. Subsequently a requirements list can be generated that encompasses the functional and non-functional requirements of the product. Each requirement is evaluated towards its criticality, depending on its importance in the product and how uncertain the team is to be able to build it.

c) Viability

In the viability plan phase, basic assumptions on the business environment are collected. We encourage using a lean canvas, as it helps to structure this information in a good overview.

The research question of the viability cycle is whether, given the current state of information, a viable business model can be derived.

2) DO

In the do phase, a prototype is being created with the sole objective to answer the previously formulated research question. To this end, the prototype is first characterized towards its type and characteristics to reach the desired objective. Subsequently, the prototype is being built.

a) Desirability

In the desirability cycle, the prototypes often comprise of paper prototypes, sketches or questionnaires in combination with an identified set of stakeholders that allow to answer questions on why a product is desirable and which aspects are most important to the desirability.

b) Feasibility

From the requirements list, critical functions are identified. These are functions that define the product's success. E.g. for a plane this might be the ability to fly in an early iteration, or a maximal required landing strip length of 2500m in a later iteration. Then, a prototype that allows to verify exactly that critical function is derived. Especially in early iterations, it's important to show that components can be built before attempts at integrating them into modules are made.

c) Viability

In the viability do phase, a market research is conducted to understand the size of the business opportunity. At the same time the costs of developing, building and selling the current solution is calculated in a ballpark estimate.

3) CHECK

In the check phase, the previously built prototype is evaluated and the learnings derived. To this end, all data from the DO phase is collected and evaluated to accept or reject the original research question.

a) Desirability

In the desirability cycle, the hypotheses are evaluated and accepted/rejected based on the findings and initially formulated acceptance criteria.

b) Feasibility

The prototype is evaluated if it exhibits the desired critical function, as well as if achieving this function leads to new requirements not yet reflected in the requirements list.

c) Viability

In the viability check phase, the previously collected data is integrated into a business model calculation and a simple P&L calculation can be used as the acceptance criteria.

4) ACT

The ACT phase is common to all cycles and the central integration point of information, where the team adapts the problem-solution fit based on the CHECK phases of the completed cycles. In the ACT phase it is also decided on how to continue. This is done by evaluating the remaining uncertainty in FDV and allocating resources accordingly. It is important to note, that a completed cycle can change the uncertainty in the other areas. E.g. if a feasibility cycle shows that the envisioned solution cannot be built, this also increases uncertainty in desirability and viability.

5) Cycle Synchronisation

As in TAF cycles can be run in parallel and there is no defined run-time of a cycle the ACT phase can be reached out of sync. This is to be addressed by an ACT meeting, where a team member of each still running cycle is present together with the teams that just completed their cycles. There the learnings from the completed cycles are to be presented and for each still running cycle, it's decided if the information gained by completing it, is still relevant to the product. If so, the cycle is continued, if not the cycle is cancelled. The team of the just completed cycle either decides to start another cycle or join running cycles.

6) Summary

In summary, TAF is an approach to minimize the uncertainty u of a product, depending on the three main variables desirability d, feasibility f and viability v i.e. $\min_{v,f,d} u(d,f,v)$ using a variable descent method, i.e. $\min_{v,f,d}[\ldots \min_{v,f,d}[\min_{f,d}[\min_{d} u(d,f,v)]]]$, where the order of the variables is to be evaluated after every iteration and the minimizer is the scientific method.

C. Evaluation of TAF

In order to evaluate the efficacy of TAF we rolled it out during the 5th iteration of Think.Make.Start. (TMS) with 69 students in 13 teams. TAF has been introduced in 3 lectures, on the second day on the general framework, on the fourth day on the feasibility cycle and on the fifth day on the viability cycle. These lectures were interspersed with lectures on SCRUM, design thinking and lean startup.

Each Team had its own Google Drive folder shared with the organization team and TAF was implemented as a series of Google Docs corresponding to the different methods on the TAF framework poster. The documents were split by desirability, feasibility and viability, each of which was implemented as a folder with the corresponding documents within. Furthermore, we provided a physical methods box with additional methods for each of the three categories. We asked the teams to save the artifacts created by these methods into the corresponding folders. Google Docs is tracking each revision of every document which allowed us to track changes to the documents and evaluate how well and when the methods have been applied, as well as in what order and how many TAF cycles have been completed.

Finally, we conducted a survey with the students to assess their experience in applying TAF throughout the course.

1) Criteria

To assess the efficacy of TAF, we would assume to

1. Find no teams that failed to build a prototype with proven viability, desirability and feasibility unless they were not applying TAF at all.
2. Have overall better products than in previous iterations of TMS.
3. Have the majority of the students find the framework at least partially helpful from their perspective.
4. Have the majority of the students apply the framework if they were to build their own product in the future.

IV. Findings

A. TAF's efficacy

Assessing TMS #5 efficacy qualitatively, there was a clear progress in product quality in comparison to previous iterations of TMS.

All the 13 teams where able to present working prototypes on stage, compared to 7 out of 9 teams in TMS #1, #2 and #3 and 8 out of 10 teams in TMS #4.

Furthermore, the complexity of the presented solutions was much higher. For the first time, we observed teams using deep neural networks in software, also for the first time, there were teams using the lathe and welding equipment in the MakerSpace, a high-tech workshop they could use throughout the course. In previous iterations of TMS, the use of tools was mostly restricted to FDM 3D printers and the laser cutter. Another novelty was that one team integrated research technology (BLDC motor controllers from the myrobotics.eu project) into their project that was developed at one of the chairs hosting TMS at TU Munich.

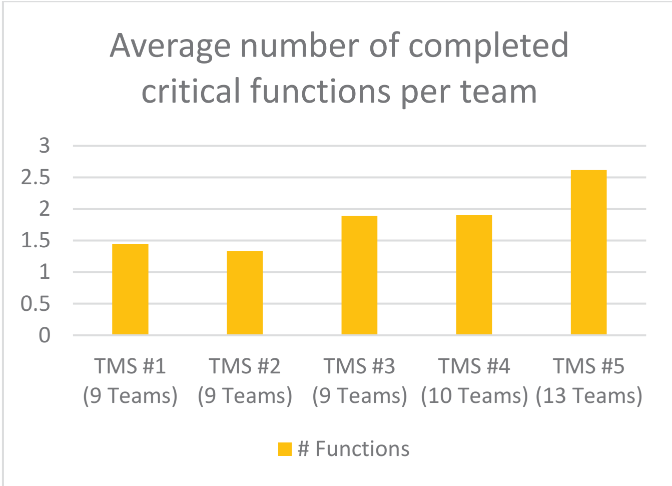

Fig. 5. Average number of completed critical functions per team during the different iterations of TMS

Additionally, during TMS #5, the teams completed more than 35% more critical functions than in previous iterations of TMS. At the same time, these functions showed similar or higher complexity than in previous iterations. It is to be assumed that this is due to a stronger focus on creating product structures and focusing on building critical functions before starting to build the prototypes, as we have seen teams wasting days on infeasible modules in earlier iterations of TMS. Thus, we see Criteria 1 and 2 confirmed.

In a survey taken during TMS #5, we asked the students on their experience with TAF:

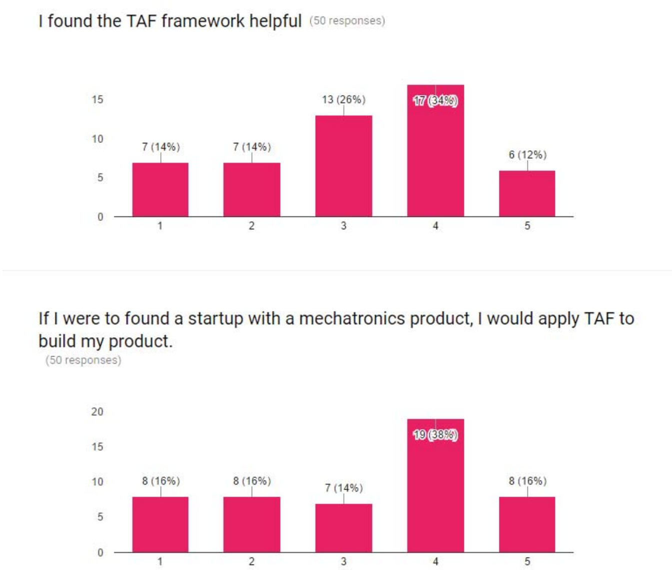

Fig. 6. Survey results with questions on TAF during TMS #5, answers range from 1 = not at all, to 5 = definitely.

As Fig. 6 shows, 46% of the students answered positively (4 or 5) to the question "I found the TAF framework helpful", while 54% of the students would "apply TAF to build their product, if they were to found a startup with a mechatronics product".

Thus, we see Criteria 3 and 4 only partially fulfilled. Asking for feedback on these results, we received statements, that we should have taught the methods earlier and focus on fewer methods. As we also introduced methods like Lean Startup and Design Thinking, we will focus on a more focused curriculum in the future. Another critique was that the methods should have been more thoroughly explained and a number of teams had issues with the Google Docs implementation of them. We believe that we can remedy these shortcomings in future iterations of TMS.

In summary, all criteria were at least partially fulfilled. Thus, we see the first instantiation of TAF as a success with room for improvement for future iterations, especially in the communication of its components.

B. *TAF vs. SCRUM*

While SCRUM is the de facto state of the art solution for agile product development in software, it has a number of direct limitations that TAF addresses. These can be classified into two categories: handling innovation and mechatronics.

1) Handling Innovation

SCRUM is a framework that assumes a product owner who can answer any question on what to build. The product owner can define user requirements in the form of user stories that then only need to be split and refined by the team into actual tasks to be executed. This prior assumption is invalid for innovations. There is no product owner with complete knowledge but rather a team with a vision and a corresponding problem-solution idea. Further, SCRUM has no in-built method to react on any change on the business environment (desirability & viability) - instead it assumes that these changes are facilitated by the product owner who then prioritizes features in the backlog differently. As such, SCRUM provides guidance for efficient management of projects in a way that allows for high flexibility and adaptability. Nevertheless, SCRUM is mainly geared at pure development teams. There are only limited reports for SCRUM teams, that added innovation to the development process. During innovation planning meetings, the team brainstormed ideas and clients could also add ideas. These ideas formed an innovation backlog managed by the product owner. Nevertheless, with Scrum there was a danger that the team lost sight of the product [34].

In contrast, TAF starts with a holistic view on the product from the beginning, encompassing business and engineering of a product. This holistic view allows to handle the limited knowledge on what to build and to react swiftly to changing environments, be it on the business or engineering side. Thus there is no requirement for a product owner role, as it is assumed by the framework and the interaction of the team with potential customers and users.

2) Mechatronics

SCRUM lives off the assumption that there is an increment after every sprint that can be shipped to and evaluated with the customers and users. However, the nature of physical products prohibits this cycle. Once a physical product is shipped, making modifications to it is costly or even impossible except for the soft- or firmware part of it. Thus for a product to be mass-produced there is a start of production (SOP) date after which typically no changes are made to its hardware. It is thus in the nature of product development models such as VDI2221 to find and eradicate any errors through extensive specification, design and evaluation phases before signing off a product for SOP.

However, where these methods are too rigid, requiring detailed specifications upfront and thus cannot accommodate for changes in the market and technology landscape, SCRUM is too loose as there is no specification until the refinement takes place.

TAF addresses this by iteratively refining technical specifications and applying traditional product development methods as part of its process. Incremental development and rapid prototyping for example gives many opportunities for students to reflect and improve. This is based on the assumption that early guesstimates are better than not assessing complexity of parts at all.

For more complex products where different components and modules have vastly different evolution time, we envision a recursive application of TAF to fast evolving modules to maintain flexibility and more traditional approaches to modules with a slow evolution and long development time. However, at the time this is just outlined and will require evaluation in future work.

V. Conclusion

In this study, we analyzed the outcome of 37 student projects in combination with existing literature to deduce an agile framework for mechatronic product development. To this end, we evaluated the most helpful methods that have been applied throughout the course and deemed most useful by the students. These methods are inspired from Lean Startup, Design Thinking as well as traditional product development. They were then structured into a guiding framework of independent PDCA cycles for Desirability, Feasibility and Viability.

This new framework was then introduced to another batch of 13 mechatronic student projects in order to evaluate its efficacy. What we have observed is that applying the framework led to a more consistent use of the methods provided. Also, while in earlier iterations TMS there were always teams which were not able to complete their prototypes, in this iteration all teams were able to complete them. This was also the first time, all prototypes were presented live on stage and worked.

While the results provide a promising direction to further the understanding and evaluation of TAF, this study has three important limitations that will need to be reviewed in future work. First, we could only observe a two week period of early stage product development. While this was sufficient to see the application of many PDCA cycles and actual agile product development, the time is too short to evaluate how this would have affected later stages of product development. Especially, for products with development time of several years, the efficacy of TAF will need to be separately studied. Second, all the products, while mechatronic in their nature, were of limited complexity. Some examples, such as “We ‘R Cars” were integrating computer vision, mechanics and electronics and built indeed very complex systems. However, to be manageable in the short timeframe, they hid much of much of this complexity by using existing products - in this case, existing remote controlled cars whose controller they hacked. Thus, it is still to be evaluated if TAF can help towards building complex mechatronic products in an agile fashion. Third, the study takes place in a lab course rather than a corporate context. While this allows to quickly examine a very broad number of samples and thus a quantitative evaluation, there is the question in how much the results are transferable to a corporate context.

One main goal of TAF is to handle a volatile, uncertain, complex and ambiguous (VUCA) environment in the global markets for such corporates [36]. To this end, we argue that while for corporates VUCA is mostly a challenge due to the increased pace of change and the subsequent necessity for reevaluation of the knowledge base of the company, for the student teams, the lack of a priori knowledge creates a comparable situation. Nonetheless, this issue will require more in-depth study, where TAF is applied in a corporate context.

Another goal for TAF is to enable its users to cope with the vastly different evolution and development time for different components. Where e.g. a motor in a car typically takes several years to bring to production, consumer electronics change within months to years and software may change within weeks to months. We envision to solve this by applying TAF recursively to fast changing components, where maintaining more rigid processes such as adaptations of VDI2221 for components where TAF has converged to stable requirements that take a long time to build. The synchronization of requirements and modularization would then take place in the ACT phase of the parent TAF-cycles. (See Fig. 7)

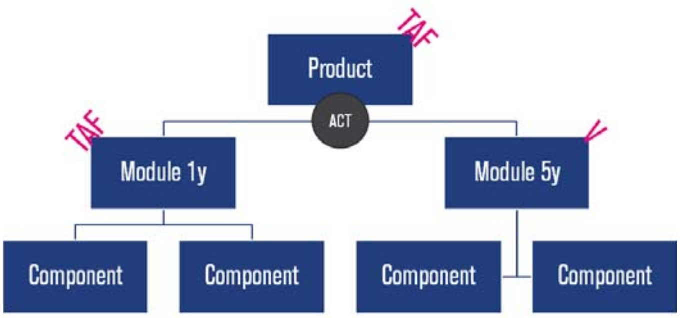

Fig. 7. Recursive application of TAF with ACT synchronization.

Another issue not yet encompassed within the current manifestation of TAF is where to handle regulatory requirements as well as how to harmonize TAF with Quality Management Systems such as ISO9001 and ISO13485. While we currently believe that this is best handled in the ACT phase, as it touches Desirability, Feasibility and Viability, a closer analysis is required.

In summary, we see this first manifestation of TAF as a promising approach to agile mechatronic product development, built from the experience of more than 50 mechatronic products prototypes and see it to be mature enough to introduce first pilots into corporate product development processes.

ACKNOWLEDGMENT

Great acknowledgments to the team of UnternehmerTUM and the students of TUM for their support and enthusiasm. Besides Technical University of Munich, the Vector foundation, and the Zeidler research foundation funded TMS. Furthermore, our thanks go to Johannes Nies, for the beautiful TAF design. Finally, a big thanks to the MakerSpace team for their support. This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Grant Agreement No 720270 (Human Brain Project SGA1)

VI. REFERENCES

[1] Thomke, S., & Reinertsen, D. (1998). Agile product development: Managing development flexibility in uncertain environments. California Management Review, 41(1), 8pp.

[2] P.G.Smith, „Flexible Product Development – Building Agility for Changing Markets“, San Francisco: Jossey-Bass, 2007

[3] K. Gaubinger et al., Innovation and Product Management, Springer Texts in Business and Economics, DOI 10.1007/978-3-642-54376-0_8, # Springer-Verlag Berlin Heidelberg 2015

[4] P.Link, "Agile Methoden im Produkt-Lifecycle-Prozess - Mit agilen Methoden die Komplexität im Innovationsprozess handhaben," In: Schoeneberg, K.-P. (ed.): Komplexitätsmanagement in Unternehmen (p.65-92). Wiesbaden: Springer Fachmedien, 2014

[5] Wedeniwski, Sebastian. "Enterprise Architectures: Surrounded by Historical Structures." The Mobility Revolution in the Automotive Industry. Springer Berlin Heidelberg, 2015. 29-74.

[6] Kallenbach, E.; Birli, O.; Saffert, E.; Schäffel, C. (1997): Zur Gestaltung integrierter mechatronischer Produkte. In: Verein Deutscher Ingenieure (pub.): Mechatronik im Maschinen und Fahrzeugbau, Düsseldorf: VDI-Verlag 1197, pp. 1-14.

[7] Dohmen, W. (2002): Interdisziplinäre Methoden für die integrierte Entwicklung komplexer mechatronischer Systeme. Dissertation, München: Herbert Utz Verlag GmbH, pp. 2-47.

[8] Uhlmann, E.; Schröder, C. (1998): Agile Produktion als Antwort auf den Wandel der Märkte. In: ZWF, Vol. 93, No. 5, p. 181.

[9] Gausemeier, J. (2008): Domänenübergreifende Vorgehensmodelle. In: TransMechatronik, pp. 1-17.

[10] Reinhart, G.; Angerer, T.: Automated Assembly of Mechatronic Products. In: Annals of the CIRP, Vol. 51, No. 1, pp. 1-4.

[11] Klein, Thorsten P., and Gunther Reinhart. "Approaches for integration of agile procedures into mechatronic engineering of manufacturing systems." Enabling Manufacturing Competitiveness and Economic Sustainability. Springer International Publishing, 2014. 225-230.

[12] J. M. Morgan and J. K. Liker, "The Toyota Product Development System – Integrating people, process, and technology", New York 2006.

[13] M. Poppendieck, "Principles of Lean Thinking" (2002), <http://www.sel.unsl.edu.ar/ApuntesMaes/Anteriores/MetodologiasAgiles/LeanThinking.pdf>, 21.12.2015.

[14] M. Poppendieck and T. Poppendieck, "Lean Software Development – An Agile Toolkit for Software Development Managers", Boston 2003.

[15] C. Hammers, "Modell für die Identifikation kritischer Informationspfade in Entwicklungsprojekten zur projektindividuellen Umsetzung der Quality-Gate-Systematik", Aachen 2012.

[16] A. Komus, "Status quo agile", Koblenz 2012.

[17] K. Beck, "Embracing change with extreme programming", in: IEEE Computer 32 (1999) 10.

[18] P. Abrahamsson, "Agile software development methods", Espoo 2002.

[19] I. Sommerville, "Software engineering", München 2012.

[20] K. Schwaber, "Agiles Projekmanagement mit Scrum", München 2007.

[21] A. Komus, "Status Quo Agile 2014", Koblenz 2014.

[22] W. E. Deming, "Out of the crisis", Cambridge 1986..

[23] Lim, Y. K., Stolterman, E., & Tenenberg, J. (2008). The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. ACM Transactions on Computer-Human Interaction (TOCHI), 15(2), 7-38

[24] Yang, M. C. (2005). A study of prototypes, design activity, and design outcome. Design Studies, 26(6), 649-669.
[25] Neeley, W. L., Lim, K., Zhu, A., & Yang, M. C. (2013). Building fast to think faster: exploiting rapid prototyping to accelerate ideation during early stage design. In ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. V005T06A022- V005T06A022). American Society of Mechanical Engineers.
[26] Boothroyd, G. (1994). Product design for manufacture and assembly. Computer-Aided Design, 26(7), 505-520.
[27] Menold, Jessica, Timothy W. Simpson, and Kathryn W. Jablokow. "The Prototype for X (PFX) Framework: Assessing the Impact of PFX on Desirability, Feasibility, and Viability of End Designs." ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2016.
[28] Böhmer, Annette I., Andreas Beckmann, and Udo Lindemann. "Open Innovation Ecosystem-Makerspaces within an Agile Innovation Process." ISPIM Innovation Symposium. The International Society for Professional Innovation Management (ISPIM), 2015.
[29] Komus, A. (2012): Status Quo Agile - Verbreitung und Nutzen Agiler Methoden. BPM Labor Hochschule Koblenz, University of Applied Sciences, Koblenz.
[30] Gloger, B. (2011): Scrum. Produkte zuverlässig und schnell entwickeln. Vol. 3, München: Carl Hanser Verlag.
[31] Kirchhof, M.; Aghajani, B. (2010): Agil in die Sackgasse. Mange it, Vol. 8, No. 9, p.1-6.
[32] Klein, Thorsten P., and GunthBer Reinhart. "Approaches for integration of agile procedures into mechatronic engineering of manufacturing systems." Enabling Manufacturing Competitiveness and Economic Sustainability. Springer International Publishing, 2014. 225-230.
[33] T. Chow and D-B. Cao, “A survey study of critical success factors in agile software projects”, in: The Journal of Systems and Software 81 (2007).
[34] B. Jenny, “Projektmanagement”, Zürich 2009.
[35] Raithatha, Dharmesh. "Making the Whole Product Agile–A Product Owners Perspective." International Conference on Extreme Programming and Agile Processes in Software Engineering. Springer Berlin Heidelberg, 2007.
[36] Johansen, B. (2007). Get There Early: Sensing the Future to Compete in the Present. San Francisco, CA: Berrett-Koehler.