



UNIVERSITY OF APPLIED SCIENCES LANDSHUT

FACULTY OF ELECTRICAL AND INDUSTRIAL ENGINEERING

Bachelor's Thesis

To what extent is TRIZ/TIPS an agile method?

Written by

David Fabian Leiner

from Taufkirchen (Vils)

Submitted: 31.08.2020

Supervisor: Professor Andrea Badura

Abstract

Since the late twentieth century, the business environment has changed dramatically: globalization, rapid and even accelerating change, as well as increasing interconnectedness have presented companies with both great opportunities and challenges. In order to achieve excellence and be successful, it is no longer enough to use conventional tools and methods. Due to their qualities to effectively deal with these changed circumstances, agile development methodologies are increasingly receiving access to the corporate world. The goal of this bachelor's thesis is to ascertain to what extent the "Theory of Inventive Problem Solving" (short: TRIZ/TIPS) can be considered an agile method. To get an overview of both TRIZ/TIPS and agile methods, a thorough literature review on both subjects of matter is conducted at the beginning. Subsequently, a catalogue of criteria for agile methods is developed based on an in-depth literature review on their character. Eventually, interviews with experts are carried out in order to gather their opinion about the concordance of TRIZ/TIPS with each of the prior developed criteria for agile methods. By doing so, it is found out that TRIZ/TIPS definitely has agile characteristics and can thus be called partially agile—however, the opinions of the individual experts sometimes vary greatly, and the topic itself seems to be controversial.

Table of contents

List of figures	v
List of abbreviations.....	vi
1 Introduction	7
1.1 Background.....	7
1.2 Aim and outline of the thesis.....	8
2 Theory of Inventive Problem Solving (TRIZ/TIPS).....	9
2.1 Historical background and development	9
2.2 Levels of inventiveness	10
2.3 Basic principles	12
2.4 The TRIZ/TIPS approach in contrast to conventional methods.....	13
2.5 The TRIZ/TIPS process.....	15
2.5.1 Case analysis.....	16
2.5.2 Detection of possible solutions.....	17
2.5.3 Solution selection	18
2.6 TRIZ/TIPS in the real working environment	18
2.6.1 Software solutions	18
2.6.2 TRIZ/TIPS in the industry.....	19
2.6.3 Different fields of expertise.....	20
2.6.4 Reasons for usage of TRIZ/TIPS.....	20
2.6.5 Application of TRIZ/TIPS tools.....	21
2.6.6 Benefits and challenges	22
3 Agile methods.....	23
3.1 Historical background and development.....	23
3.2 The Agile Manifesto.....	24
3.2.1 Values	24
3.2.2 Principles	25
3.3 Agile methods in the working environment.....	29
3.3.1 Different industries	29
3.3.2 Digital tools	30
3.3.3 Reasons for usage	30
3.3.4 Scrum.....	31
3.3.5 Benefits and challenges	32
4 Creation of a catalogue of criteria for agile methods.....	34
4.1 Development approach.....	34

4.2	Success measurement	35
4.3	Management style.....	35
4.4	Culture	36
4.5	Attitude towards change	36
4.6	Documentation	37
4.7	Emphasis.....	37
4.8	Process.....	38
4.9	Planning style	38
4.10	Mindset.....	39
5	Expert consultation	40
5.1	Intention and design of the interviews.....	40
5.2	Results	41
5.2.1	Development approach	41
5.2.2	Success measurement	42
5.2.3	Management style.....	43
5.2.4	Culture	43
5.2.5	Attitude towards change	44
5.2.6	Documentation.....	45
5.2.7	Emphasis.....	45
5.2.8	Process	46
5.2.9	Planning style	47
5.2.10	Mindset	47
6	Conclusion	49
7	References.....	50
	Appendices	55
	Declaration of Academic Integrity.....	71
	Release Certificate.....	72

List of figures

Fig. 1: Outline of the bachelor’s thesis	8
Fig. 2: Development of TRIZ/TIPS – Rough timeline	10
Fig. 3: Statistical occurrence of the different levels of inventiveness	12
Fig. 4: Random conservative approach to finding a solution	14
Fig. 5: The systematic TRIZ/TIPS approach	15
Fig. 6: The TRIZ/TIPS process with some of its tools	16
Fig. 7: Industries the respondents worked in	29
Fig. 8: The Scrum process	32
Fig. 9: Concordance of TRIZ/TIPS with the agile development approach	41
Fig. 10: Concordance of TRIZ/TIPS with the agile success measurement	42
Fig. 11: Concordance of TRIZ/TIPS with the agile management style.....	43
Fig. 12: Concordance of TRIZ/TIPS with the agile culture	43
Fig. 13: Concordance of TRIZ/TIPS with the agile attitude towards change.....	44
Fig. 14: Concordance of TRIZ/TIPS with the agile documentation.....	45
Fig. 15: Concordance of TRIZ/TIPS with the agile emphasis.....	45
Fig. 16: Concordance of TRIZ/TIPS with the agile process.....	46
Fig. 17: Concordance of TRIZ/TIPS with the agile planning style	47
Fig. 18: Concordance of TRIZ/TIPS with the agile mindset.....	47
Fig. 19: Concordance of TRIZ/TIPS with the characteristics of agile methods – Summary	49

List of tables

Tab. 1: The five levels of inventiveness – Overview	11
--	----

List of abbreviations

CAI	Computer Aided Invention/Innovation
i.e.	id est
TRIZ/TIPS	Theory of inventive problem solving
VUCA	Volatility, Uncertainty, Complexity, Ambiguity

1 Introduction

1.1 Background

The business environment has changed dramatically since the late twentieth century—globalization, steadily rapid and even accelerating change, as well as increasing interconnectedness have created both great opportunities as well as enormous challenges for companies (Joiner, 2019, p. 140). Furthermore, new technologies and megatrends such as digitalization, industrial internet of things, and artificial intelligence are presenting companies with a fundamentally changed situation (Kiesel & Hammer, 2018, pp. 55–57). Whereas industries, companies, and customer needs used to be clearly defined and delimited, organizations had well-established, traditional competitors, and operated in a climate of certainty, the aforementioned megatrends are now dissolving industry boundaries and changing the erstwhile character of products, processes, and services.

This unprecedented dynamic and challenging new industrial time is increasingly being referred to by experts and organizations with the acronym “VUCA”, which was coined by the “US Army War College” in 1998 and is composed of the first letters of “Volatility”, “Uncertainty”, “Complexity”, and “Ambiguity” (Saleh & Watson, 2017, p. 705). Therewith, the so-called “VUCA world” is characterized by unpredictable and quickly changing circumstances, lacking information with respect to future issues and events, overwhelmingly complex development conditions due to the many interrelated variables and the sheer volume of information involved, as well as numerous ambiguities due to missing precedents.

As a result, it is no longer enough to achieve excellence using traditional tools and techniques—in order to be successful, changes must be sensed and responded to in a fast and flexible manner, and companies generally must be able to quickly adapt to the ever-changing, turbulent environment (Horney et al., 2010, p. 33). Due to their qualities to deal with these circumstances, agile development methodologies are receiving increasing access to the corporate world (Casner et al., 2018, p. 85).

1.2 Aim and outline of the thesis

The aim of this thesis is to analyze to which extent TRIZ/TIPS can be assigned to agile methods. First, a thorough literature review on both subjects of matter will be conducted in order to gain a deep understanding of both of them. Second, a catalogue of criteria for agile methods based on an in-depth literature review on their character will be developed. Third, a consultation of experts in both TRIZ/TIPS as well as agile development will be carried out in the form of interviews with the goal to gather their opinions in respect to the concordance of TRIZ/TIPS with the prior developed catalogue of criteria for agile methods. Fourth and finally, a conclusion regarding the research question of this bachelor's thesis "To what extent is TRIZ/TIPS an agile method?" based on the results from the expert interviews will be formed.

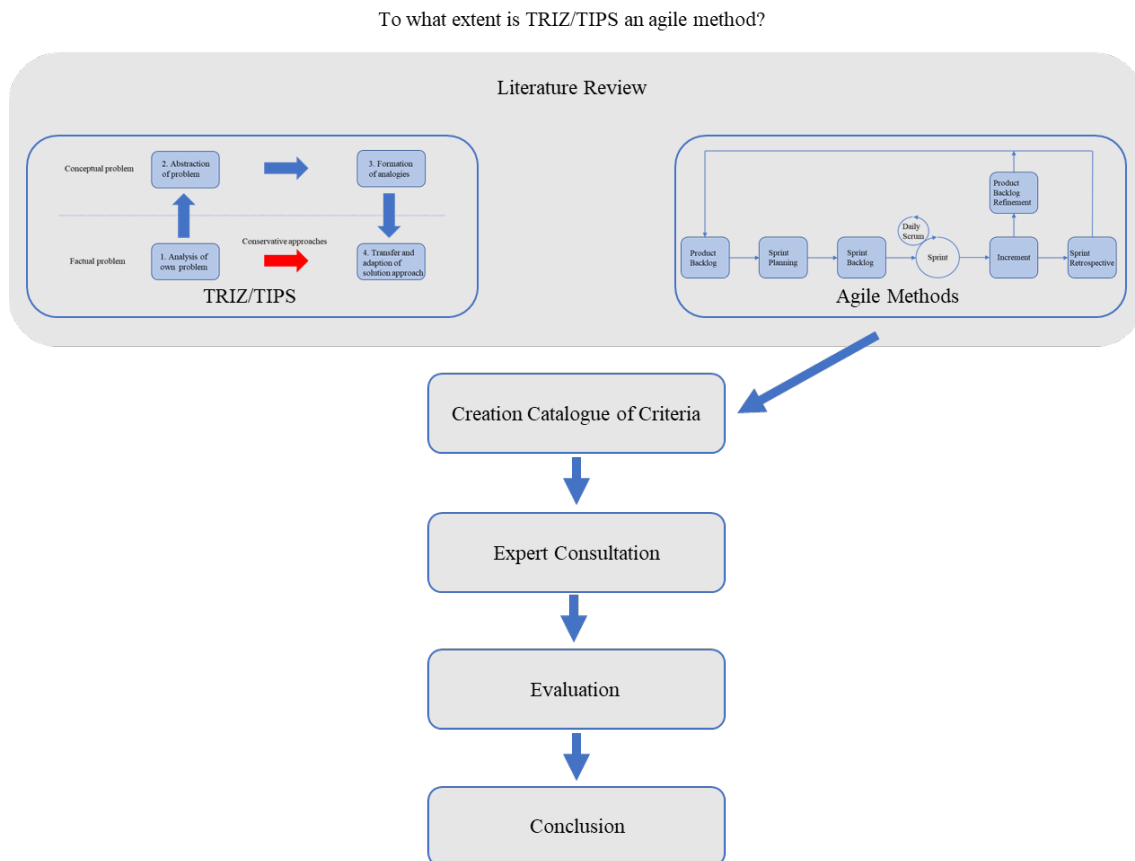


Fig. 1: Outline of the bachelor's thesis

(Own representation)

2 Theory of Inventive Problem Solving (TRIZ/TIPS)

2.1 Historical background and development

The “Theory of Inventive Problem Solving” was originally created by the Russian inventor, writer, and independent scientist Genrich Saulowitsch Altschuller (1926–1998) under the name “*Теория Решения Изобретательских Задач*” (short: ТРИЗ) and is nowadays known predominantly by the abbreviation TRIZ (Hentschel et al., 2010, pp. 25–26). Altschuller already applied for his first patents at an early age and, as a patent officer in the Russian Navy, gained extensive insight into the work of the developers, whom he assisted in the application for property rights. Prior to the Second World War, he began to analyze patents and their underlying technical and scientific inventions thoroughly, therewith laying the first foundation for TRIZ/TIPS.

In 1948, he was arrested and sentenced to 25 years imprisonment in a Siberian penal camp on the grounds of “acting in a seditious manner” by the Soviet Union’s state and party leader Joseph Vissarionovich Stalin due to his criticism of the technical problem-solving measures taken. However, this did not stop him to—under challenging conditions—further continue his research with other like-minded, intellectual people.

In consequence of Stalin’s death, Altschuller was released prematurely from prison in 1954—shortly afterwards, in 1956, he and his former classmate Rafael Borissowitsch Shapiro first introduced TRIZ/TIPS to the broad public with the publication of the paper “About a technology of creativity” (Wang et al., 2013, p. 992). Thus, they contradicted the common perception of creative processes as not projectable and uncontrollable but abrupt, unpredictable intuitions and illustrated a systematic approach to creativity and inventiveness which could be taught and learned by almost everybody (Mareis, 2015, p. 505).

Only a few years later, Altschuller and his companions fell again into disfavour of the Russian government—as a result, their teachings got prohibited (Klein, 2014, pp. 4–5). For this reason, TRIZ/TIPS had to be further developed and taught underground, without the government’s knowledge. Altschuller, the unconventional thinker that he was, did this by writing science fiction novels under the pseudonym “Henry Altov” in which he explained TRIZ/TIPS contents hidden in various stories.

Finally, when the political movement for reformation within the communist party of the Soviet Union, known as “Perestroika”, took place during the ‘80s, TRIZ/TIPS was once again allowed to be trained and applied openly. After the fall of the Iron Curtain in 1989, TRIZ/TIPS also became popular outside of Russia. Since then, it has been continuously developed further, built into innovation software, and a popular research topic within academia (Chechurin & Borgianni, 2016, p. 120).

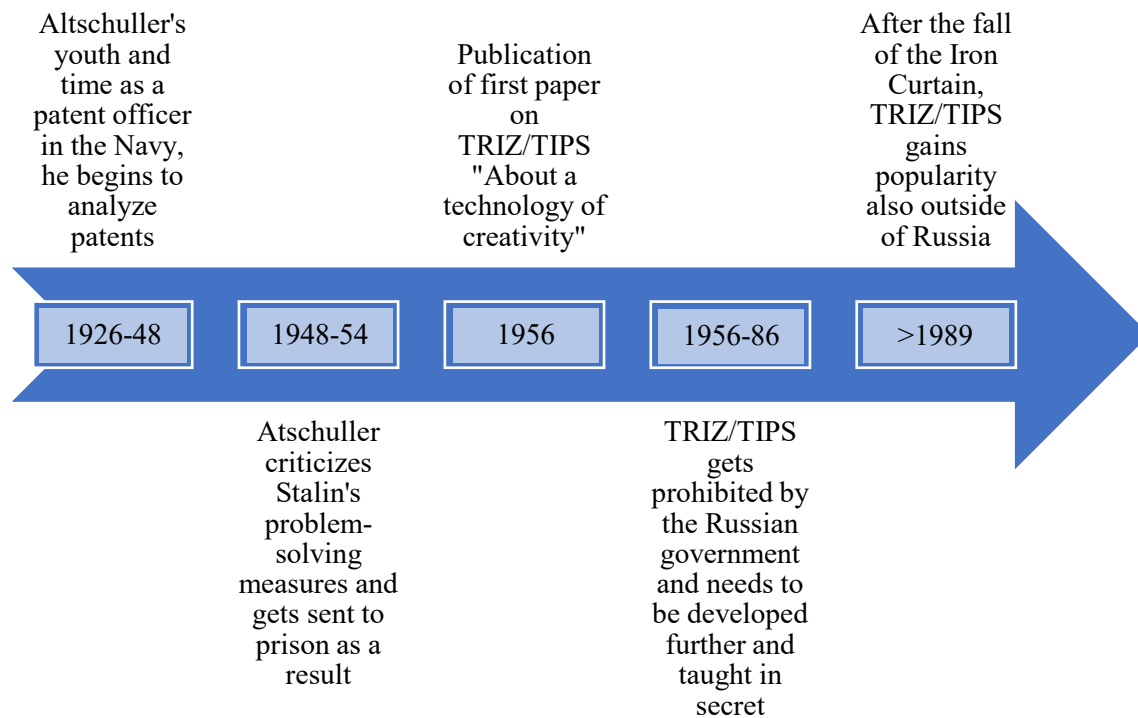


Fig. 2: Development of TRIZ/TIPS – Rough timeline

(Own representation according to Hentschel et al., 2010; Klein, 2014; Wang et al., 2013)

2.2 Levels of inventiveness

From 1964 to 1974, Genrich Altschuller thoroughly analyzed roughly 200,000 published patents—to date, a total of about 2,5 million international patents have been analyzed (Klein, 2014, p. 1)—and discovered that there were different levels of difficulty of solutions to engineering problems, each occurring with a different frequency (Hentschel et al., 2010, p. 28). Based on these findings, he quantified five levels of inventiveness, as shown in summary in **Tab. 1:** The five levels of inventiveness – Overview below, which are valid until today (Klein, 2014, p. 6).

Tab. 1: The five levels of inventiveness – Overview

(Own representation according to Gadd, 2011, pp. 26–29; Hentschel et al., 2010, p. 28; Klein, 2014, p. 6; Orloff, 2006, p. 19)

Level	Frequency	Description	Knowledge	Aspect
1	32%	Apparent solution	Company knowledge	Rationalization
2	45%	Minor improvement	Industry knowledge	Modernization
3	18%	Major improvement	Other industries	Principle
4	4%	New concept	Interdisciplinary	Synthesis
5	<1%	Breakthrough	New science	Discovery

With a total of about 77%, Altschuller found that the majority of patents did not have a high level of inventiveness and presented relatively straightforward and obvious solutions to problems that could be achieved by most engineers with the help of simple, basic knowledge (Gadd, 2011, pp. 26–28). Thereof, 32% were merely engineering enhancements, belonging to level one which is about using knowledge easily available inside your company in order to resolve a problem with an apparent solution (Klein, 2014, p. 6). The other 45% were minor improvements being classified as level two inventions which require knowledge that is not as accessible as is for level one and cannot be found inside your company but instead within your industry.

With a frequency of occurrence of 18%, Altschuller further defined level three solutions as significant improvements that require knowledge from a whole different industry and present a fundamentally different way of dealing with their problem by adopting solutions proven in other contexts (Gadd, 2011, pp. 26–27).

Level four solutions are much rarer and are being developed by people from entirely different fields working together and combining interdisciplinary knowledge (Gadd, 2011, p. 27; Klein, 2014, p. 7). These four per cent of patents are new concepts or applications that are created predominantly in science and represent real ground-breaking innovations. Finally, statistically occurring less than once per 100 registered patents and, thus, being the rarest, level five inventions are astonishing, often accidental scientific breakthroughs which exceed the limits of pre-existing knowledge (Gadd, 2011, p. 27; Klein, 2014, p. 7).

These discoveries often have a high impact and lead to completely novel products or technologies.

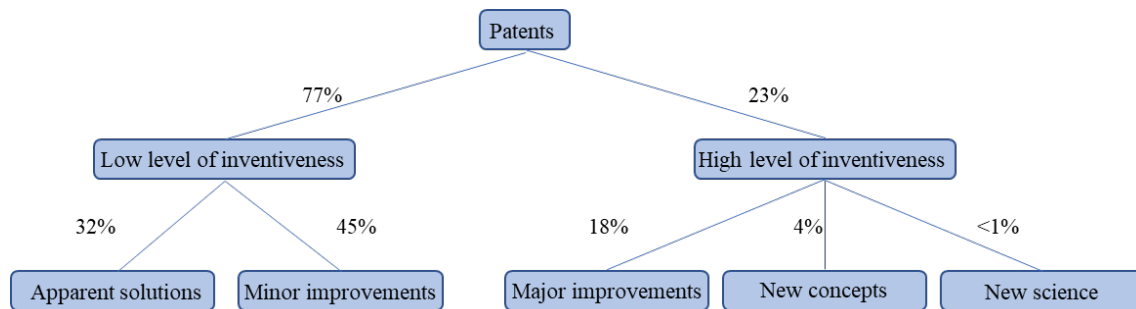


Fig. 3: Statistical occurrence of the different levels of inventiveness

(Own representation)

By uncovering these five levels of inventiveness, Altschuller made clear that the finding of solutions to engineering problems comes in various degrees of difficulty which often require much more than just your own company's knowledge (Gadd, 2011, p. 27). Furthermore, he recognized that—with level one solutions having practically no inventive depth and being solely slight advancements and level five inventions calling for totally new breakthroughs which cannot really be enforced—a generally applicable approach would be suitable mainly for levels two, three, and four (Hentschel et al., 2010, p. 28; Klein, 2014, p. 8). On account of this, the Russian scientist henceforth focused his work on facilitating inventions of these levels by developing a systematic procedure.

2.3 Basic principles

After Altschuller and his co-workers had conducted their patent analysis and defined the five different levels of inventiveness, they recognized that the majority of patents were based on standard procedures and techniques known and used by professionals in the working world (Hentschel et al., 2010, p. 26). Thereupon, in order to understand revolutionary inventions and their development more deeply, they examined the highly innovative minority of patents and identified three key findings which act as the foundation for TRIZ/TIPS.

First, they detected that every invention is based on an initial contradiction which seems to be insoluble (Hentschel et al., 2010, p. 27). In other words, real innovations neither

arise from compromise solutions nor optimizations, but from the elimination of an initial, seemingly unsolvable conflict of requirements (Orloff, 2006, p. 37).

Second, they realized that a large number of inventions share a comparably small number of underlying approaches to finding a solution (Hentschel et al., 2010, p. 27). To wit, even though problems differ many, show strong similarities and can be solved with the same solution approach (Orloff, 2006, p. 37).

Third and last, the evolution of technical systems happens in certain, recurring patterns (Hentschel et al., 2010, p. 27). That is, products' evolution patterns often resemble each other, allowing to make predictions as to how technical systems will develop further and when particular limits will be reached (Klein, 2014, p. 8).

Using the newly acquired knowledge in the form of these findings, Altshuller and his colleagues then proceeded to develop TRIZ/TIPS—a systematic approach with a set of different tools that aims at providing a more creative, target-oriented, and effective way of thinking in order to discover innovative solutions (Hentschel et al., 2010, p. 27; Klein, 2014, pp. 1–2; Orloff, 2006, pp. 37–38).

2.4 The TRIZ/TIPS approach in contrast to conventional methods

In order to achieve solutions for difficult problems, there is a whole variety of different conservative creativity methods. The oldest and still most widespread one thereof is “trial and error” which is the process of checking an assumed solution that, most of the time, has spontaneously come into the inventor's mind—if the solution does not work out the right way, it gets discarded, and another one is tested instead (Zobel, 2007, pp. 5–6). This cycle is repeated until a working solution is found or the search for such is aborted. Another well-known and widely used conservative method is “brainstorming” in which in a first phase a range of different ideas is generated and in a second one the best idea is selected therefrom (Zobel, 2007, p. 6).

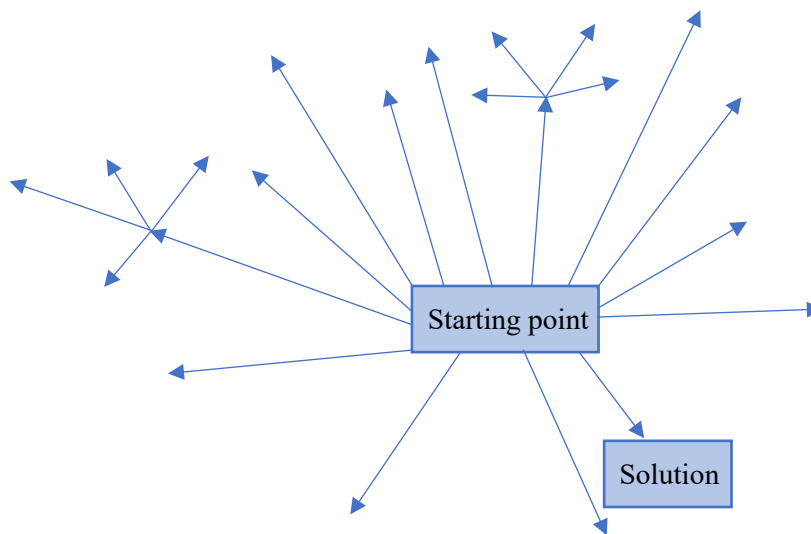


Fig. 4: Random conservative approach to finding a solution

(Own representation according to Hentschel et al., 2010, p. 22; Zobel, 2007, p. 5)

While these methods are to a large extent based on intuition, spontaneity, and lucky coincidences, TRIZ/TIPS offers a much more systematic guideline consisting of four successive steps which facilitate finding solutions by using knowledge unfamiliar to oneself but proven to work for someone with a similar problem elsewhere (Gadd, 2011, p. 29).

Withal, the first step is the analysis of the problem (Hentschel et al., 2010, p. 29; Klein, 2014, p. 8). This is because, in order to proceed with the next steps and eventually find a solution, one first needs to understand the problem at hand thoroughly.

The second step is the abstraction of the problem (Hentschel et al., 2010, p. 29; Klein, 2014, p. 8). To be precise, the real, factual problem is broken down to an abstract level by detaching it from all specific details and treated further as a simple conceptual problem (Gadd, 2011, p. 29).

The third step is the formation of analogies and the finding of ideas to solve the problem (Hentschel et al., 2010, p. 29; Klein, 2014, p. 8). During this step, analogous problems already known from Altschuller's patent analysis and their solution approaches are sought for the original, previously abstracted problem (Gadd, 2011, p. 29). This is achieved with the help of a variety of special tools and, in this process, it does not matter whether the analogous problems' solutions found derive from the same or a completely foreign area of expertise.

Lastly, the fourth step is the retransfer of the solution approach (Hentschel et al., 2010, p. 29; Klein, 2014, p. 8). In other words, the found solutions to the analogue problems are used to solve one's own problem (Gadd, 2011, p. 29). Transferring and adapting them away from the abstract, conceptual point of view to the real, factual problem at hand represents the greatest challenge in all of the TRIZ/TIPS process and requires a lot of expertise, knowledge, and engineering spirit.

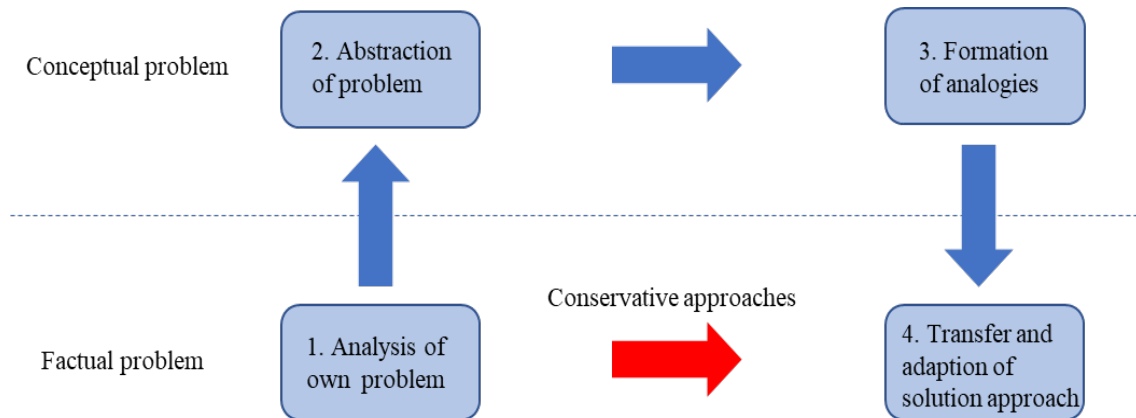


Fig. 5: The systematic TRIZ/TIPS approach

(Own representation according to Hentschel et al., 2010, p. 30)

In summary, TRIZ/TIPS—in contrast to the conventional creativity methods—does not directly search for a solution to an existing problem but takes a seemingly longer route via the analysis of the problem, its abstraction, and the formation of analogies and generation of ideas (Hentschel et al., 2010, pp. 29–31). This systematic approach, however, saves a tremendous amount of time and energy and almost guarantees high-quality solutions for challenging problems while being independent of spontaneous inspirations or even luck as is often the case for the conventional methods.

2.5 The TRIZ/TIPS process

In order to systematically solve difficult problems step-by-step with the help of TRIZ/TIPS, there is a multitude of available methods, principles, strategies, approaches, techniques, and insights—essentially, TRIZ/TIPS is a toolbox for systematic creativity and innovation (Koltze & Souchkov, 2017, p. 35). The entire process with some of its tools can be seen in **Fig. 6** and is gradually explained hereinafter.

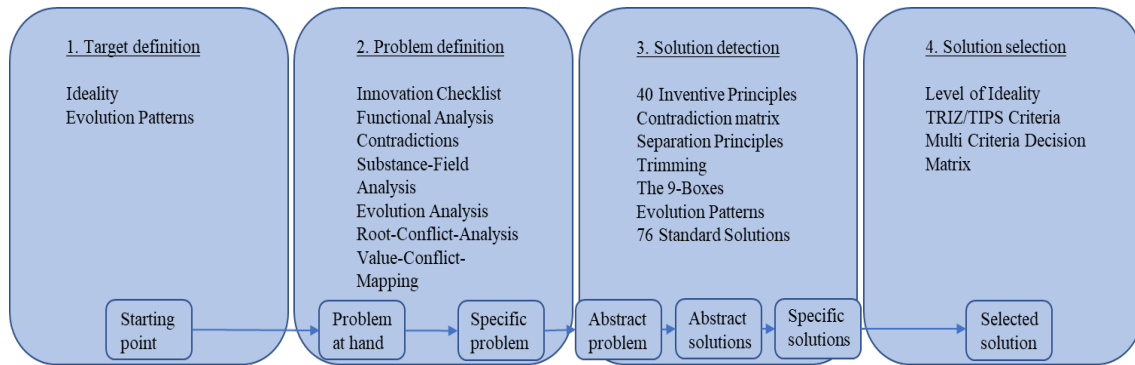


Fig. 6: The TRIZ/TIPS process with some of its tools

(Own representation according to Koltze & Souchkov, 2017, p. 225)

2.5.1 Case analysis

2.5.1.1 Definition of desired outcome

In a first and vital step, TRIZ/TIPS mainly uses the concept of ideality in order to define the desired outcome of the problem-solving process (Koltze & Souchkov, 2017, pp. 36–40)—in that matter, ideality describes the quality of the solution to a problem. Initially, when dealing with a problem, a best imaginable solution is come up with while altogether leaving out of consideration whether it is at all possible in reality or certain compromises must be accepted in its realization. This best imaginable solution then serves as an orientation aid in the further process steps, and an attempt is made to approach it as closely as possible. The so-called “Ideality Equation”, as shown below, illustrates the relation of the ideal solution’s benefits to its costs and harms and should ideally approximate infinity.

$$Ideality = \frac{\Sigma \text{benefits}}{\Sigma \text{costs} + \Sigma \text{harms}} \sim \infty$$

As with any other fraction, increasing its result (ideality) is achieved by increasing the value of the numerator (benefits) and/or decreasing the value of the denominator (costs and harms). This vision of a completely “ideal technical system”, short “ideal machine”, describes the utopia of achieving a maximum benefit without having to put up with any costs or harms and allows an unlimited focus on creating a best-case scenario for orien-

tation purposes (Koltze & Souchkov, 2017, p. 40). When it comes to developing an already existing technical system further TRIZ/TIPS uses a principle called the “Ideal Final Result”, short IFR (Koltze & Souchkov, 2017, pp. 41–42). The idea thereby is to achieve the ideal solution with the system’s pre-existing resources exclusively and to change as little as possible of the already established, previous solution. In summary, the increase of ideality is the overarching, universal development goal of TRIZ/TIPS and allows to work goal-oriented and efficient by showing a clear path (Koltze & Souchkov, 2017, pp. 42–45).

2.5.1.2 Detailed definition of the problem

Now that the desired outcome has been formulated and a clear direction is given, the next step is to understand and define the problem at hand meticulously. There are many different tools available for this purpose such as the “Innovation Checklist” which—although not being one of the original methods—has proven to be very helpful especially for TRIZ/TIPS newbies (Koltze & Souchkov, 2017, pp. 226–230). The checklist helps the TRIZ/TIPS user to precisely describe the prevalent problem, narrow it down to a more specific problem, and define the contradiction to be dealt with—this is done with several questions which help to gain a deeper understanding of the whole situation. In order to break free from already cemented ideas and subject-specific patterns of thought, it is tried to answer these questions in the simplest possible terms without using technical expressions. Ultimately, this step of defining the problem ends with its abstraction that is detaching it from all specific details and acts as the starting point for further action.

2.5.2 Detection of possible solutions

In this step of the process, the detection of possible solutions for the previously abstracted problem takes place. TRIZ/TIPS again offers a variety of different tools for this purpose from which the user can flexibly and situationally choose the most efficient and appropriate ones. If, for example, a technical contradiction—the targeted improvement of a parameter of a system that results in an inadvertent deterioration of another parameter of the same system—has to be solved the “40 Inventive Principles” tool is the right choice (Koltze & Souchkov, 2017, pp. 66–93). This tool is based on Altschuller’s finding from his patent analysis that the resolution of similar contradictions is achieved by applying

similar solution approaches most of the time and presents forty different concepts which have often led to a successful solution in the past and are very likely to do so for other problems as well. To find out which of the principles should be applied in order to find a solution, the so-called “Contradiction Matrix” is a helpful tool (Koltze & Souchkov, 2017, pp. 100–104). However, if the solution has to overcome a physical contradiction—the case when a physical parameter is supposed to have a particular characteristic to fulfil a requirement and at the same time have the exact opposite characteristic to fulfil another requirement—the tool of the “Separation Principles” is the right one (Koltze & Souchkov, 2017, pp. 106–112). Withal, the two conflicting requirements are analyzed with regard to the four aspects space, time, structure, and change of conditions since each physical contradiction can be resolved by one or more of these aspects. As already mentioned, there is a wealth of different tools for the detection of solutions—the just described ones serve as examples and the TRIZ/TIPS user must decide which method is the most suitable for him on a case-by-case basis. Furthermore, the user must thereby bear in mind that it is not the quantity of the solutions but their quality, i.e. the desired outcome defined at the beginning of the process that is most important (Koltze & Souchkov, 2017, pp. 221–222).

2.5.3 Solution selection

Once the TRIZ/TIPS user has found various possible solutions to his problem, their evaluation and prioritization follow (Koltze & Souchkov, 2017, pp. 221–224). This is done, for example, by comparing the different solutions to the initially defined ideality and then creating a ranking based on the results, with the first place being the solution closest to ideality. Further possibilities are their evaluation on the basis of the “TRIZ/TIPS Criteria” or the “Multi Criteria Decision Matrix”. Eventually, the solution concept that is perceived to be the best, most suitable one is selected and implemented.

2.6 TRIZ/TIPS in the real working environment

2.6.1 Software solutions

Since the emergence of TRIZ/TIPS, numerous software products have been developed by different software enterprises that all aim to support working with the comprehensive

toolkit and to facilitate access to its extensive knowledge (Koltze & Souchkov, 2017, pp. 257–259). One of the best known and most widespread “Computer Aided Invention/Innovation” (short: “CAI”) program in the anglophone world is “Innovation Workbench” from Ideation International, USA (IDEATION INTERNATIONAL INC, 2020), the most popular German CAI program is “TriS AIDA” from TriS Europe, Germany (TriS Europe, 2019). Essentially, these computer programs have four different functions: the use of graphic tools for the sketching and analysis of models, of algorithms for a guided step-by-step problem-solving process, and of knowledge databases as well as the archiving and administration of gained insights and results. Although these functions can be beneficial, CAI software is by no means required for success. It often even makes TRIZ/TIPS thinking as well as good cooperation difficult and, moreover, is expensive to purchase—hence, companies should consider well whether the software actually entails valuable advantages compared to working manually before acquiring it (Gadd, 2011, p. 21).

2.6.2 TRIZ/TIPS in the industry

With some companies using the aforementioned CAI software, and others do not, TRIZ/TIPS is generally very widespread in the global industry (Souchkov, 2016, pp. 8–9). There are numerous organizations which offer TRIZ/TIPS based services, most of which are located in the US and South Korea, as well as thousands of manufacturing companies that use TRIZ/TIPS—from large global players such as BMW, General Electric, or Unilever to small and medium sized enterprises. The latter play a unique role because they usually have much fewer resources available that can be allocated to solving complex problems in comparison to the global companies, yet they have to tackle problems and be innovative in order to survive in the competitive environment. Due to its effectiveness and efficiency, TRIZ/TIPS can help these firms to do so without demanding too many resources.

Aside from treating it as a stand-alone methodology, TRIZ/TIPS is also often used in the industry in combination with other methods such as “Quality Function Deployment” (short: “QFD”), “Failure Mode and Effects Analysis” (short: “FMEA”), and “Six Sigma” (Spreafico & Russo, 2016, p. 54). Therefore, TRIZ/TIPS can not only increase innovation directly but also enhance the effectiveness of other methods.

2.6.3 Different fields of expertise

Generally, the TRIZ/TIPS methodology is applicable in any field of expertise. In an analysis of published scientific papers with “TRIZ” in the title, summary, or keywords indexed by Elsevier’s Scopus database—thereby including high-quality scholarly articles according to peer-reviewing standards only—Bayer found in his thesis that the majority of them could be assigned to engineering science, followed by computer science, business management and accounting, and mathematics (Bayer, 2019, p. 25). Similarly, in a 2012 questionnaire that, among other things, aimed to understand the employment of TRIZ/TIPS in practice, the vast majority of respondents used TRIZ/TIPS for technical problem solving and technology-oriented fields in general while only a few applied it in business management (Ilevbare et al., 2013, pp. 33–34). The use of TRIZ/TIPS in non-technical fields of expertise seems to be so little widespread because business people who do not have technical backgrounds and thereby only have very little technical know-how often get the methodology taught by technology experts using a technical terminology as well as technical examples (Souchkov, 2016, pp. 15–16). That is why, due to the cognitive gap, this knowledge does not get deeply comprehended by them and, as a result, is not applied very often.

2.6.4 Reasons for usage of TRIZ/TIPS

In 2015, scientists Spreafico & Russo conducted a representative study in which they analyzed more than 200 papers on TRIZ/TIPS based consultancy projects published in “The TRIZ Journal”, “ETRIA TRIZ Future Conference”, as well as other major journals of engineering design. Among others, the study addressed the question of why exactly companies needed TRIZ/TIPS, more specifically, for what purposes it was mainly used. The scientists found that the most common reasons for the utilization of TRIZ/TIPS were quality improvement of products, systems, and services, pollution reduction in order to stay within eco-design guidelines and generate brand new concepts, launching new products, improving the productivity while problem-solving, and improving already existing products and processes. Substantially less often, companies used TRIZ/TIPS for energy reduction, safety improvement, and cost reduction. With these goals being relatively general, Spreafico & Russo further analyzed the case studies to gain a deeper understanding

of the specific activities in which TRIZ/TIPS was used. Their findings showed that companies used the methodology in e.g. early design to find alternative physical effects or characterize the behaviour of potential solutions, decision making and forecasting, and optimization and robust design.

2.6.5 Application of TRIZ/TIPS tools

In order to figure out how TRIZ/TIPS tools are applied by companies in the real working world, Moehrle conducted an analysis of more than forty case studies of European and American companies within different industries directly related to the TRIZ/TIPS methodology and accessible by the public (2005). The scientist found that none of the companies included in his study used all of the available TRIZ/TIPS tools and all of them successfully used different combinations of tools, showing that there are different routes to success when using the methodology. Withal, thinking in contradictions was the mostly applied tool similar to the findings of other studies (Spreafico & Russo, 2016, p. 53). Furthermore, Moehrle identified three, non-industry-specific clusters of application—namely “basic TRIZ”, “resource and ideality based TRIZ”, and “substance-field based TRIZ”. As suggested by its name, the first one therefrom represented quite simple to understand and easy to apply TRIZ/TIPS and only came with the utilization of two tools: thinking in contradictions for target-oriented creativity as well as using the “40 Inventive Principles” in combination with the “Contradiction Matrix”, thereby taking advantage of Genrich Altschullers extensive collection of technical knowledge. In contrast, the second and third listed application clusters worked with considerably more advanced tools such as the “Resource Analysis”, the “Substance-field Analysis”, and the concept of ideality while at the same time also using a higher number of tools. Based on these findings, Moehrle recommends TRIZ/TIPS newbies to begin with what he identified as “basic TRIZ” and gain experience, before proceeding to work with the more advanced and complicated tools. Moreover, he suggests experienced TRIZ/TIPS users to compare the clusters identified in his study with their own application habits and get inspiration on how to improve their innovation/problem-solving processes therefrom.

2.6.6 Benefits and challenges

As is with every other methodology, TRIZ/TIPS has both advantages as well as disadvantages. In 2012, Ilevbare et al. carried out an online survey of TRIZ/TIPS users to get insights from the practical perspective of the method. Their questionnaire was made accessible to two LinkedIn groups which had specifically been created to mutually support each other with the understanding and applying of TRIZ/TIPS and asked the survey participants to name the method's benefits and challenges they had experienced in their work with it. According to the respondents, one significant benefit is the structured approach to problems in contrast to the somewhat luck-requiring brainstorming or random searching for solutions and its ability to help to clear up problems and to provide solution hints. In addition, they cited the method's capability to generate high quality, innovative ideas that lead to useful and often novel solutions as a benefit. This is, moreover, achieved within shorter amounts of time compared to other methods due to a quick finding and focussing on the problem. Other benefits mentioned were the provisioning of a view on future developments as well as an increased effectiveness of teamwork. Despite these great benefits of TRIZ/TIPS, the respondents also listed several challenges they had encountered when working with it. Many criticized the methodology's rigidity and difficulty to adapt for applications in different contexts as well as its general complexity. Another challenge often associated with the method was the fact that it requires a thorough understanding and practical experience for it to deliver successful results, which in turn implies that first time users need to dedicate much time in order to achieve successes with it. Due to the just mentioned challenges as well as the sheer amount of tools and many different approaches to resolving problems with TRIZ/TIPS, it makes sense that some of the study participants showed their frustration and called for a standardized guide of best practices.

3 Agile methods

3.1 Historical background and development

The emergence of the agile methodology dates back as far as the 1930s—according to scientists Larman and Basili, it originated from quality expert Walter Shewhart’s proposal in the 1930s to do an iterative series of short plan-do-study-act (PDSA) cycles in order to control and continuously improve products and processes (2003). In the following decades, numerous new approaches in the context of iterative and incremental software development such as the waterfall model, the spiral model, and the Rapid Application Development method were presented—all of which show strong similarities in their proceedings (Hohl et al., 2018, pp. 3–4). Even though these methods yielded significant improvements for the world of software development, many practitioners criticized their limitations being, among others, the difficulty to understand and implement them quickly, the lacking ability to adapt to changing circumstances as well as requirements, and the great labour/time-intensity (Misra et al., 2012, p. 974).

Following a meeting of Extreme Programming advocates in 2000 during which the attendees showed their support for what was then called “light” or “lightweight” methodologies, software professional Robert Martin invited all of the lightweight leaders—of whom some had developed their own methods and processes—to find common ground and discover advanced ways to develop software during a two-day conference in 2001 (Beck et al., 2001). At this conference, which was held in the Wasatch Mountains of Utah, US, the group of seventeen independent experts named themselves “The Agile Alliance” and developed the basis of all of today’s agile approaches—“The Manifesto for Agile Software Development”, short “The Agile Manifesto”. In the years following this historic meeting, the agile mindset not only became increasingly popular in the world of software development but also in the context of physical product development and other lines of business. Furthermore, companies from all over the world progressively perceived overall organizational agility to be the solution to survive and thrive in the VUCA world.

3.2 The Agile Manifesto

Based on compatible values, mutual trust and respect for each other, as well as the will to promote a more people and collaboration focused organizational model in which they would want to work themselves, Beck et al. formulated a set of four values backed up by a set of twelve principles in The Agile Manifesto (2001).

3.2.1 Values

The four values stated in The Agile Manifesto by Beck et al. (2001) are:

1. **“Individuals and interactions over processes and tools”**

The members of a team should be encouraged and invoked to use their special skills and intensely interact with each other in order to solve problems and make progress in the development of the product without being forced to follow specific processes and use specific tools (Layton & Ostermiller, 2018, pp. 42–43). This means, in an agile environment, people are the focus of attention.

2. **“Working software over comprehensive documentation”**

The focus should lie on the development of actually usable, potentially deliverable functionality instead of unnecessarily detailed documentation of the process (Layton & Ostermiller, 2018, pp. 44–46). In the agile mindset, a document is only considered useful if it actively supports the development of the product rather than distracting from it.

3. **“Customer collaboration over contract negotiation”**

Instead of being in contact with the customer only at the beginning and end of a project as well as when contractual modifications need to be discussed as is the case with traditional development projects, the agile methodology aims at establishing an active cooperation with the customer throughout the entire project (Layton & Ostermiller, 2018, pp. 46–47). Thus, the customer is closely involved in all the project activities and can make valuable contributions helpful to both parties.

4. **“Responding to change over following a plan“**

Instead of blindly following a rigid plan regardless of what happens, a certain flexibility should be guaranteed at all times during the development of a product

(Layton & Ostermiller, 2018, pp. 47–48). This allows the development team to respond quickly and effectively to changes requested by the customers and users as well as changed market situations in order to develop a successful and relevant product.

Beck et al. stress that the elements on the right are neither unimportant nor should be ignored—in their opinion, however, it is the elements on the left that really matter and, hence, should be given the greater emphasis.

3.2.2 Principles

The twelve principles stated in The Agile Manifesto by Beck et al. (2001) are:

1. **“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”**

The main goal of the agile mindset is to continuously deliver business value to the customer starting from early on in the development process, thereby making them benefit quickly from their investment as well as know the progress made so far (Cooke, 2010, p. 105). The focus lies on fulfilling the top priority requirements first rather than spending time on unnecessary work so that each delivery provides actual value to the customer.

2. **“Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.”**

Embracing change and understanding that it is inevitable is mandatory in order to keep up with the ever-changing circumstances as well as emerging opportunities during a development project and gain a competitive advantage over other organizations (Cooke, 2010, pp. 104–105). This is achieved by planning in a responsive manner and concentrating on short time frames rather than planning the whole project upfront as is done traditionally—thereby, the ability to easily adapt to changes without any difficulties is ensured.

3. **“Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”**

This third principle is an extension to the first one—the customer should not only be supplied with business value continuously but also at regular intervals, which should be kept short rather than long in order to remain responsive and mitigate

the risk of doing unnecessary work (Cooke, 2010, p. 105). Only if a sub-product gets accepted by the customer, the delivery counts as successful and completed, and the development process is continued further.

4. “Business people and developers must work together daily throughout the projects.”

Agile aims on directly involving all relevant external as well as internal stakeholders in the process and keeping all of them informed at all times in order to allow an active participation and best meet the customer’s requirements (Cooke, 2010, p. 106). Especially internal business people and developers should thereby cooperate as closely as possible to discuss current issues, update each other about the progress made so far, and keep the documentation lightweight.

5. “Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”

Due to the motivating effects, it is fundamentally important for the success of a project to empower the team members, trust them, and give them the ability to self-manage their work (Cooke, 2010, pp. 109–110). Agile approaches take place in an environment embossed by interdependency—the business people set the economic priorities, measure the outcomes, and determine the development objective, whereas the developers estimate the time needed for the different tasks required to achieve the objective considering ongoing work and other obligations and ultimately execute them. While the execution is taking place, the business people do not need to closely monitor the developers’ actions due to the constant exchange of information that is happening anyway as well as the frequent deliveries where they see the results of the developers’ work at the latest. The developers, in turn, get satisfaction, pride, and motivation out of the trust that is showed to them as well as the results they see after each iteration.

6. “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”

The authors of The Agile Manifesto strongly advocate for communicating face-to-face in real or virtual (video conferences) meetings whenever it is possible (Cooke, 2010, p. 107). In contrast to calls or e-mails, this form of communication is much quicker and more effective, allows to see non-verbal cues of the participating stakeholders and, therewith, reveals precious information that would have

stayed hidden otherwise. In addition, it positively influences active participation in discussions.

7. “Working software is the primary measure of progress.”

Instead of endlessly creating time-consuming status reports which can be designed to look like everything is perfectly acceptable when it is not, real, tangible outputs are taken to measure the progress—these allow an unadulterated look on the outcomes of the developers’ actions (Cooke, 2010, pp. 108–109). Aside from that, focussing mainly on producing tangible outputs once more leads to avoiding unnecessary work whenever possible and causes satisfaction and motivation within the developers because they continuously get to see the results of their hard work.

8. “Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”

Working agile is all about working sustainable (Cooke, 2010, pp. 110–111). All people involved in an agile development process should be able to maintain a high level of productivity as well as quality at all times. Being productive, responsive, and quickly adapting to changed circumstances should in no case mean developers working overtime, pulling all-nighters and suffering from exhaustion or burnout—instead, this is achieved through joint effort and a certain flexibility in the design of deliverables.

9. “Continuous attention to technical excellence and good design enhances agility.”

In order to reach a high level of agility, teams should place a great emphasis on continuous technical excellence and good design rather than on unnecessary other things. If they do not do so, problems that could have been avoided otherwise will occur sooner or later, requiring the allocation of resources to resolve them and impacting agility in a negative way.

10. “Simplicity—the art of maximizing the amount of work not done—is essential.”

This principle of agile methodologies points out the immense importance of learning to differ work that is actually relevant for the development of a product from work that does not add any value or maybe even decelerates the entire development process and can, thus, be neglected (Cooke, 2010, pp. 107–108). The developing team’s focus should be on doing work that adds value for the customer,

improving wherever and whenever possible, and minimizing waste by, for example, eliminating processes that are outdated or automating work. Thereby, regular feedback from the customer as well as encouraging the developers to shape their work as efficiently as possible help.

11. “The best architectures, requirements, and designs emerge from self-organizing teams.”

Agile supporter firmly believe that the best results arise from empowering teams, enabling them to self-manage their work, and supporting them within the development process, not from micro-managing them and planning top-down only. This conviction is based on several other agile principles already mentioned. First, according to the fourth principle, agile calls for daily cooperation between business people and developers in order to best meet requirements and keep documentation lightweight. Second, as contained in the fifth principle, showing trust to developers has a motivating effect and, therewith, positively influences their work. Last and stated in the seventh principle, tangible outputs are the primary measures of progress instead of theoretical work. In conclusion, developing teams know best what needs to be done—by giving them a certain level of autonomy in their job through the promotion of self-management, the best outcomes are achieved.

12. “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. “

Finally, the twelfth and last principle of the agile philosophy is the pursuit of continuous improvement (Cooke, 2010, p. 111). Agile teams regularly review their way of working in terms of efficiency and improve it whenever possible. Furthermore, they also ensure that their focus is on essential activities only, and no valuable resources are wasted unnecessarily.

Essentially, the twelve principles penned by Beck et al. in *The Agile Manifesto* are product development best practices regarding customer satisfaction, quality assurance, teamwork, and project management (Layton & Ostermiller, 2018, p. 49).

3.3 Agile methods in the working environment

3.3.1 Different industries

For more than a decade, Singapore-headquartered software firm CollabNet VersionOne (CollabNet VersionOne, 2020), now part of Digital.ai following a merger with other software companies (Digital.ai, 2020c), has been sponsoring and publishing an annual study on the evolution of the agile workplace called the “State of Agile Report” (Digital.ai, 2020a) conducted by an independent survey consultancy named Analysis.Net Research (Digital.ai, 2020b, p. 5). The study examines, for example, the spread of agile methods, factors for success and failure, as well as the use of agile methods in different industries. In this year’s “14th annual State of Agile Report”, 1121 complete survey responses of agile practitioners from differently sized companies all over the world were gathered and analyzed providing interesting insights. With regard to the industry in which they are active, 27% of the survey respondents indicated that they work in the technology sector as can be seen in **Fig. 7**.

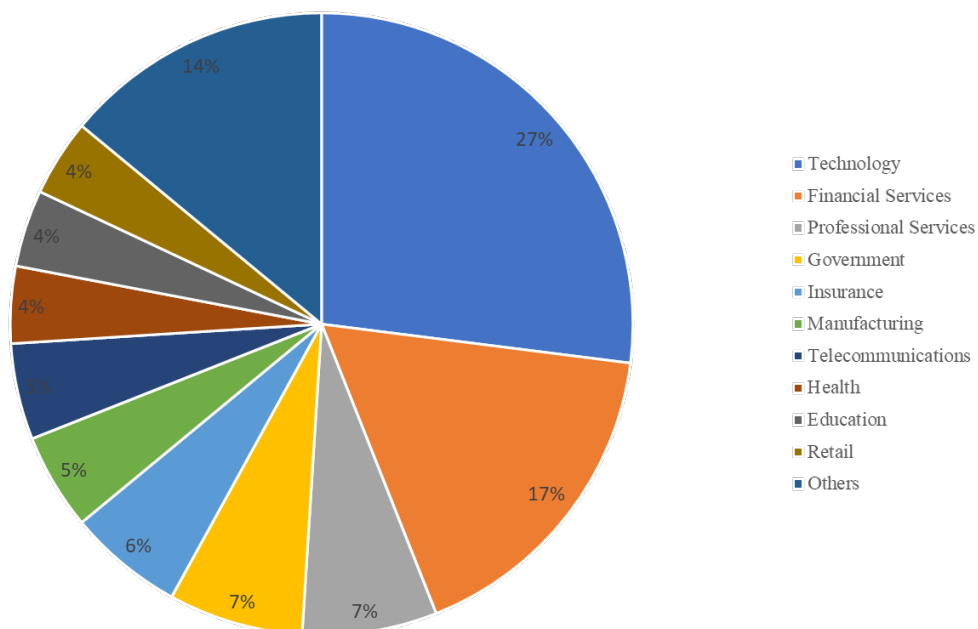


Fig. 7: Industries the respondents worked in

(Own representation according to Digital.ai, 2020b, p. 7)

This is not surprising at all, considering the fact that the origin of the agile methodology lies in software development. However, the pie chart also shows that agile methods have long since found their way into fundamentally different industries and application areas such as manufacturing, health, and education and can basically be applied everywhere.

3.3.2 Digital tools

In order to be successful in a business environment shaped by VUCA, many organizations and product development teams are using agile methods. Apart from conventional physical approaches to implement these methods, numerous digital tools have been developed in the past years to do so. Withal, probably the best known therefrom is “Jira” from the Australian software company Atlassian, which supports users in planning and implementing agile development processes by providing a corresponding web-based platform (Atlassian, 2020). Other frequently used programs by agile teams include Microsoft Excel, Microsoft Azure DevOps, Google Docs, and Version One (Digital.ai, 2020b, p. 16). Some key advantages of these programs in contrast to the physical approaches are the enormous amount of optionally available features, the high transparency as well as complete documentation of the entire process, and the accessibility to and possibility of collaborating with any stakeholder from any location (Kropp et al., 2017, p. 124). Furthermore, they are highly customizable for different purposes and can generate business reports fully automated via the touch of a button. However, some users perceive these tools to be too extensive, miss a good overview and are overwhelmed by the many features. It furthermore requires much effort to set them up and administer them. All in all, every agile team must decide for itself whether a digital tool brings significant advantages over the physical way of working agile and if so, which of the many different ones on the market is best suited in terms of costs, features, and the team’s specific needs.

3.3.3 Reasons for usage

In 2016/2017, the Laboratory for Business Process Management and Organizational Excellence under the direction of Prof. Dr. Ayelt Komus at the University of Applied Sciences Koblenz conducted a study on the success and application of agile methods (Komus & Kuberg, 2017). With the help of the online survey tool “LimeSurvey”, a questionnaire

in both English as well as German was created, activated, and then distributed to agile users via various specialist blogs, websites, publications, et cetera. Within a period of three months, more than 1000 responses were received from employees of companies within 30 different countries with annual turnovers of less than 250k€ to more than 50bn€ and consequently evaluated. In relation to the question “Why did your company decide to implement agile methods?”, more than 60% of the respondents indicated the optimization of their time-to-market. Further decisive reasons for the adoption were the improvement of product quality, the decrease of project risks, the improvement of the team morale, as well as the enhanced ability to make reliable plans which also includes an increased level of responsiveness to changing circumstances. These reasons for using agile methods are to a large extent the same as those displayed in the “14th annual State of Agile Report” (Digital.ai, 2020b) and illustrate the many positive characteristics companies associate with agile methods in comparison to traditional ones.

3.3.4 Scrum

The best known and most applied out of all agile methods is Scrum, of which the process can be seen in **Fig. 8** (Digital.ai, 2020b, p. 10; Komus & Kuberg, 2017, p. 23). A Scrum team includes three different roles—the “Product Owner”, the “Scrum Master”, as well as the “Development Team” (Pfeffer, 2020, pp. 30–35). At the beginning of a development project, the commissioning stakeholders communicate their motivation and goals. The “Product Owner”, who is responsible for the success of the project and corresponds to the role of the Project Manager, then uses this information as well as direct communication with the stakeholders to identify their requirements with the highest business value and records them in the “Product Backlog” according to their priority. These are worked off by the “Development Team” in scheduled feedback loops, the so-called “Sprints”. This team has the responsibility for the technical functionality of the product to be developed and, at the beginning of each “Sprint”, determines in the “Sprint Planning” meeting how many requirements of the “Product Backlog” can be implemented—these requirements are recorded in the “Sprint Backlog”. Afterwards, the development work of the next 24 hours is planned in “Daily Scrum” meetings, and during the whole “Sprint” the “Product Owner” together with the “Development Team” continuously updates the “Product Backlog” according to newly acquired information and, thus, prepares it for future “Sprints”.

At the end of each “Sprint”, the developed sub-product, the so-called “Increment”, gets tested, documented, and presented to the “Product Owner” as well as, if requested, to the stakeholders in the “Sprint Review”—this demonstrates the current state of development, which builds the basis for the further procedure. If mistakes have been made during the development of the “Increment” or the feedback is bad for other reasons, this is the right point to update the “Product Backlog” to avoid errors in future “Sprints”. In the last event of each Scrum “Sprint”, the “Sprint Retrospective”, the complete team meets and identifies the potential for improvement in the development process and collaboration. Throughout the whole just explained process, the “Scrum Master”—an experienced Scrum user himself/herself—helps to implement the method correctly and to continuously improve all processes by removing obstacles that slow down the team.

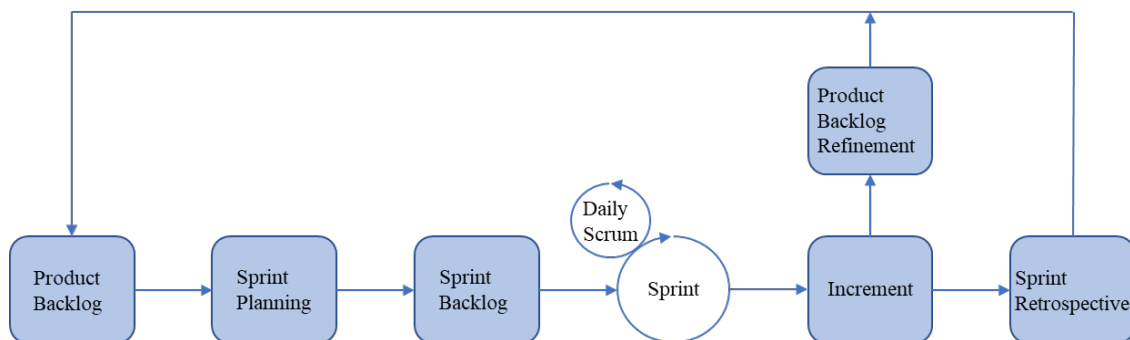


Fig. 8: The Scrum process

(Own representation)

3.3.5 Benefits and challenges

In 2014, scientists Solinski & Petersen conducted a study on, among other things, agile benefits and limitations in the working environment. For their study, they got in touch with personal contacts and used specialist agile forums and communities to find survey participants. Through this, they won over 63 people who started the online questionnaire, of whom 39 completed the time-intensive benefits and limitations part. In order to obtain insights into strengths and weaknesses of the agile methodology in relation to practical usage, the researchers applied a hierarchical cumulative voting approach in which the agile practitioners had to prioritize benefits and limitations derived from ten different studies on the subject. Withal, they differentiated between external benefits (referring to

aspects that are relevant to the customer) as well as internal benefits (referring to advantageous aspects of the agile process used over conventional approaches). In respect to the former, the results showed that especially the business value created for and the relationship established with the customer, as well as the achieved product quality are seen to be enormous advantages (Solinski & Petersen, 2016, pp. 461–462). Relating to the latter, the respondents mainly perceived the improved knowledge and learning achieved through a high level of communication, social skill development, employee satisfaction and motivation through the experienced empowerment and the feeling of being purposeful, as well as the regular feedback with the thereby gained confidence to be significant benefits. The by far most significant limitation of the agile practice according to the prioritization, in turn, was the requirement of a high and comparable level of qualification of the developers and of well-trained agile managers, and the lowered interchangeability of team members (Solinski & Petersen, 2016, pp. 462–463). Furthermore, the poor scalability in distributed environments and low suitability for projects with large teams were serious downsides of agile methods in the eyes of the respondents. To conclude, working agile has some major benefits but at the same time limitations to it—companies should consider both well before deciding to go agile and not just do so because it is a buzzword in the industry and fashionable to call oneself agile.

4 Creation of a catalogue of criteria for agile methods

Agile is a way of thinking that is shaped by the values and principles of The Agile Manifesto (Agile Alliance, 2020). The primary goal of the Manifesto's authors was to help workers implement and respond to change and deal with uncertain environments. Agile methods are the practical implementation of the Agile Manifesto and provide teams and organizations with structures and practices to follow the agile principles and values. In other words: agile methods provide generative rules that are designed to help users do the right actions under any circumstances (Highsmith & Cockburn, 2001). It is important to understand though that the stubborn adherence to a chosen agile method is the opposite of agile as defined by the authors of the Agile Manifesto—rather, one should choose the most appropriate agile method and adapt it to one's own circumstances. That is why agile enthusiasts prefer the term "framework" instead of "methodology" for Scrum and co. In the following, a catalogue of criteria for agile methods/frameworks is developed based on their characteristics.

4.1 Development approach

Agile methods are highly adaptive and responsive contrary to heavyweight methods that try to anticipate and predict the whole process up-front (Awad, 2005, p. 19; Moniruzzaman & Hossain, 2013, p. 6). The authors of The Agile Manifesto understood that change is inevitable and, due to this, it is best to reduce the cost of responding to it (Highsmith & Cockburn, 2001, p. 120). Trying to eliminate change and obsessively sticking to a detailed initial plan would mean being unresponsive to the highly volatile and competitive business environment and would, thus, ultimately lead to business failure. Hence, agile methods only plan short-term and use short iterations at the end of which it is easy to adjust to changed circumstances—regardless of whether that is changed market conditions, customer requirements, or other things—and begin a new iteration with an updated objective. Therewith, change is facilitated and accommodated optimally, and the market's and stakeholder's needs are met continuously. Besides, using short iterations also fulfils the demands of principle 1 and 3 of The Agile Manifesto (Beck et al., 2001).

4.2 Success measurement

As stated in principle 7 of The Agile Manifesto, the primary measures of success in agile development are real, tangible outputs—not conformation to plan (Beck et al., 2001). At the end of each iteration of an agile development process stands a so-called increment—a sub-product that ideally adds business value to the customer's product (Awad, 2005, p. 19). Whether the increment actually adds value does not depend on the developers' opinions but the customer's according to their requirements (Highsmith, 2006, p. 32). If it does not, corrective action must be taken, which can easily be accomplished due to the iterative and adaptive nature of agile methods. If it does, however, the customer does not care how the developers created the increment. This is the exact reason for why agilists measure the progress/success of their projects by actual business value: they know that the internally motivated developers who are highly skilled and know the customer's vision give it their all to fulfil the requirements. If they conformed to a detailed plan put together at the beginning of a project and the customer would be dissatisfied, the fact that they conformed to the plan would not make the situation better. Plans become obsolete and wrong, working products that satisfy the customers are always right.

4.3 Management style

In agile development, the management style is decentralized instead of autocratic as is within most traditional projects (Awad, 2005, p. 19; Moniruzzaman & Hossain, 2013, p. 6). Specifically, this means that the project leaders transfer the decision-making authority regarding the actual development process onto the developers because they know best what needs to be done when in order to fulfil the customer's requirements. So unlike managers foreign to the subject assigning tasks to the developers, they themselves express and then distribute them—this is exactly what is described as self-organizing teams producing the best outcomes in principle 11 of The Agile Manifesto (Beck et al., 2001). In doing so, the management removes hindering blockades, sets the goals and constraints of the project, and, following principle 5 and 8 of The Agile Manifesto, creates a sustainable development environment minted by support and trust which can be maintained at a constant pace indefinitely (Cockburn & Highsmith, 2001, pp. 132–133). Thus, the developers

can thrive and fully concentrate on what is essential and produce high-quality products in short amounts of time.

4.4 Culture

The fact that collaboration related issues are contained in two value, as well as principle 4 and 6 of The Agile Manifesto, shows that its authors were well aware of the great importance it has for projects—the agile culture is a highly collaborative one (Awad, 2005, p. 20; Beck et al., 2001; Moniruzzaman & Hossain, 2013, p. 6). Close collaboration with the customer helps to always stay on track and never get lost during the process due to the constant feedback (Highsmith & Cockburn, 2001, p. 121). Moreover, changes can be implemented at the instant, which prevents project costs from an uprise and produces high-quality results. Close collaboration between the business people and the developers of an agile project, as demanded in principle four of The Agile Manifesto, keeps both sides updated at all times. The managers get to know what is happening regarding the actual product development, and the developers get in contact with business expertise. Close collaboration between the developers themselves increases the level at which the team operates in terms of pace and quality significantly (Cockburn & Highsmith, 2001, p. 132). Consequently, knowledge about the wishes and requirements of the customers is always up-to-date, and shared expertise of all parties and sparked lively discussions about relevant topics lead to favourable outcomes. According to principle six of The Agile Manifesto, face-to-face is the preferred way of communication and should be chosen whenever possible.

4.5 Attitude towards change

Whereas traditional approaches tend to be change-averse, agile ones embrace and welcome change (Awad, 2005, p. 19; Moniruzzaman & Hossain, 2013, p. 6). As already mentioned in 4.1 Development approach and 4.2 Success measurement of this catalogue of criteria, the agile fathers understood that being responsive to change is more important than following a plan, which is achieved with the iterative nature of agile methods as well as measuring success by actual value to the customer. As can be seen in principle 2 of The Agile Manifesto, this does not only count for early stages in a project (Beck et al.,

2001). Instead, one of the central agile tenets is to welcome and encourage changed requirements in all stages of a project—even in late ones—as long as they do not violate the broad scope, project schedule, and cost constraints defined by the customer (Highsmith & Cockburn, 2001, p. 122). Agile teams thus create a decisive competitive advantage for their customers who operate in turbulent business environments and, therefore, need a certain amount of flexibility and are successful themselves due to this very reason.

4.6 Documentation

As outlined in The Agile Manifesto, working products are more important than comprehensive documentation—in other words: keeping documentation lightweight is the agile way to go as opposed to putting a great emphasis on documenting everything of a project like conservative projects do (Beck et al., 2001; Moniruzzaman & Hossain, 2013, p. 6). The reason for this is the comprehension that keeping everything simple as demanded in principle 10 of The Agile Manifesto automatically implies to not work more than absolutely necessary, which includes creating documents that will become outdated sooner or later (Awad, 2005, p. 20). Furthermore, the more comprehensive documentation gets, the more complicated it is to find the information needed as well as keep everything updated. By minimizing documentation which—from the agile point of view—means work that does not add any value to the actual product, the amount of resources needed is significantly reduced. One might think successfully finishing a project without massive documentation does not work, and this very much is the case with traditional projects. However, due to the highly collaborative culture prevailing in agile projects as explained in 4.4 Culture and the therewith fostered tacit knowledge of the whole team, it functions perfectly fine in agile environments (Boehm & Turner, 2005, p. 32).

4.7 Emphasis

For agile enthusiasts, people and their interactions are more important than processes and tools (Beck et al., 2001). So, whereas traditional development approaches are process-centred, agile ones are people-centred (Awad, 2005, p. 18; Moniruzzaman & Hossain, 2013, p. 6). The underlying understanding is thereby that if people working on a project are skilled enough, they can use nearly all existing processes and tools to fulfil their tasks—

if they are not skilled enough in their job, however, no specific process or tool will compensate for their lacking knowledge (Cockburn & Highsmith, 2001, pp. 131–133). Naturally, they need the right support from their management as described in 4.3 Management style/Allocation of work. In addition to the importance of the individual competence of each developer in a project, another fundamental reason for this attitude is the fact that the competence of the entire team is significantly increased by keeping the level of communication and interaction between the individuals at a constantly high level. This is why agilists value and strive for a highly collaborative environment, as explained in 4.4 Culture. In conclusion, agile methods do not try to standardize individuals and force them in certain processes but to make the best out of each individual's and the whole team's strengths.

4.8 Process

In engineering, processes are either defined or empirical (Williams & Cockburn, 2003, p. 40). A process that is classified as defined produces the same outputs every time it is run through, following the same predefined steps from start to finish without any variations—this is the case for traditional processes. Agile processes, in turn, are considered empirical processes due to the fact that no process is ever the same. Predefined steps as in defined processes would not lead to favourable outcomes that satisfy the customer: the highly volatile environment in which agile development takes place and for which it was invented demands a consistent responsiveness and ability to adapt to the ever-changing circumstances.

4.9 Planning style

Whereas traditional development projects are planned comprehensively up-front, agile projects only involve minimal planning (Awad, 2005, p. 35; Moniruzzaman & Hossain, 2013, p. 6). This is founded upon the highly adaptive and change welcoming character of agile methods, as explained in 4.1 Development approach and 4.5 Attitude towards change. Trying to lay out the whole agile process up-front would entail having to continually update the plan due to the inevitable, continuously happening changes—there are just far too many potentially changing variables to be able to take them all into account

(Awad, 2005, p. 19). This would violate the from the authors of The Agile Manifesto demanded simplicity, since it meant work that could have been easily avoided (Beck et al., 2001). Agile users are well aware of this and, hence, only plan very little of their process ahead.

4.10 Mindset

Agile teams are motivated and constantly thrive for efficiency and high quality in their work rather than just completing tasks and looking forward to going home at the knocking-off time. Agilists know that unceasingly paying attention to the quality of their work positively influences agility as stressed in principle 9 of The Agile Manifesto (Beck et al., 2001; Highsmith & Cockburn, 2001, p. 120). Furthermore, as shown in principle 12, they come together at regular intervals with the intent to reflect on their work, uncover ways to become more efficient, and then adjust their working behaviours accordingly. This is backed up by a prevailing mood of motivation provoked by the agile kind of management as described in 4.3 Management style/Allocation of work.

If a development method holds the characteristics just explained above, it follows the values and principles of The Agile Manifesto and can, therefore, be considered agile—if it holds some of these characteristics, but not all of them, it can be argued congruously that it is partially agile. However, a method that does not possess any of the agile characteristics is not agile by any means.

5 Expert consultation

Both TRIZ/TIPS as well as agile methods are profoundly complex topics, which require years of experience and an in-depth study to be fully comprehended by the individual user. Therefore, in order to find an answer to the research question “To what extent is TRIZ/TIPS an agile method?” of this bachelor’s thesis, the qualitative research method of expert interviews was selected. In the process, the extensive expertise of experts in both areas of subject was systematically queried in relation to the research question.

5.1 Intention and design of the interviews

The specific intention of the interviews was to gather the experts’ opinions regarding the concordance of TRIZ/TIPS with the criteria of the prior developed catalogue of criteria for agile methods. In order to guarantee that the individual interviews were as similar as possible and, thus, ensure comparability, a guideline was drawn up in advance, which can be found in Appendix A.

After the initial greeting, the assurance of the protection of personal data as well as anonymity of what will be said, five opening questions at the beginning of each interview were used to get an insight into the experts’ level of experience with both TRIZ/TIPS and agile methods. In addition, the essential characteristics of agile methods were polled in the form of an open question—on the one hand, to get a small and unbiased feedback on the catalogue of criteria and on the other, to read from the answers whether the TRIZ/TIPS experts are actually also acquainted with the agile methodology.

This was followed by an evaluation of the agility of TRIZ with the key questions based on the catalogue of criteria developed earlier. To do so, each criterion was first defined and then rated by the experts on a scale from 0 to 5, with 0 meaning “This criterion is not applicable to TRIZ.”, 1 meaning “This criterion does not apply to TRIZ at all.”, 3 meaning “This criterion partially applies to TRIZ.”, and 5 meaning “This criterion fully applies to TRIZ.”. In addition to this rating, the experts were asked to provide a justification for each of their choices.

5.2 Results

In total, three TRIZ/TIPS experts were surveyed during the interviews—a detailed graphical summary and evaluation of all questions can be found in Appendix B. All of the experts showed much more than five years of practical experience in TRIZ/TIPS but differed significantly in their respective MATRIZ levels. However, a MATRIZ certification is not required for working with TRIZ/TIPS and is entirely voluntary. Hence, the expert with MATRIZ level one does not necessarily have less expertise than the other two experts with levels four and five, but simply has not pursued a higher-level certification yet. With regard to agile methods, all three experts had to show several years of experience both as users as well as coaches—withal, each of them stated to have experience with Scrum, which is not surprising since this is the most popular of all agile development methods. Besides, two of the experts demonstrated experience with Design Thinking and Kanban, and only one expert with Fastworks, Lean Startup, and Effectuation. In relation to the last one of the opening questions “What characterizes agile methods for you?”, all interviewees answered with essential aspects of the prior developed catalogue of criteria, thus proving it right and at the same time showing their acquaintance with agile methods.

5.2.1 Development approach

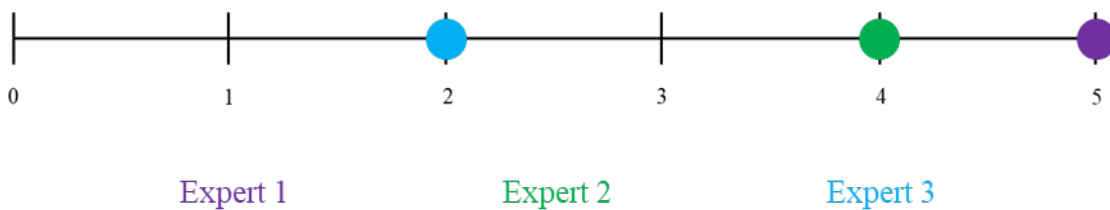


Fig. 9: Concordance of TRIZ/TIPS with the agile development approach

(Own representation)

With five and four, experts 1 and 2 rated the conformance of TRIZ/TIPS with the first of the agile criteria, the development approach, as high. The former thereof argued that TRIZ/TIPS essentially works iteratively as well due to the occurrence of secondary problems whilst trying to solve the main problem, with the only difference being that there is no prescribed systematology as in agile methods. The latter one justified their rating by the fact that TRIZ/TIPS is a toolbox and leaves it to the user to decide in which order to

use certain tools and whether to use them at all, making the method very adaptive and flexible.

Pursuant to expert 3, this criterion only applies a little bit to TRIZ/TIPS since the method itself does not know an iterative procedure and is by no means as adaptive and responsive as agile methods are—it possesses so many tools, though, that a certain degree of flexibility is achieved by simply choosing the appropriate one therefrom.

5.2.2 Success measurement

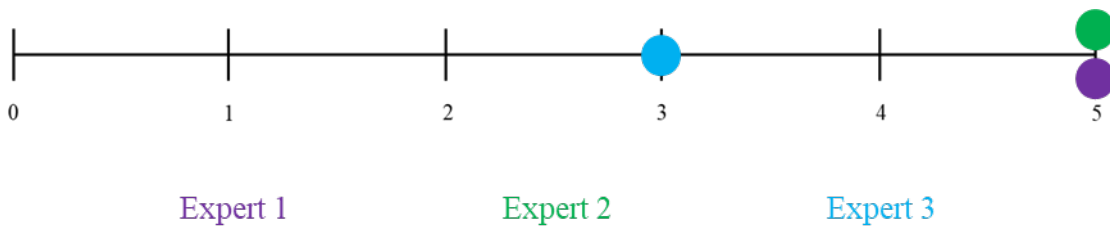


Fig. 10: Concordance of TRIZ/TIPS with the agile success measurement

(Own representation)

According to experts 1 and 2, the success measurement of TRIZ/TIPS is exactly the same as with agile methods, hence the rating of five. Both even-handedly said that with TRIZ/TIPS, the decisive factor is the result at the end of the problem-solving process and definitely not the agreement with a previously created plan.

Giving it a rating of three and thereby saying that this criterion only partially applies to TRIZ/TIPS, the opinion of expert 3 differed a little. They explained this with the fact that even though TRIZ/TIPS generates incredibly inventive ideas and does not focus on conforming to a plan, in practice the user is often overwhelmed by the ideas resulting from the process. Consequently, in many cases, the idea is not implemented, and there is no real, tangible output at the end.

5.2.3 Management style

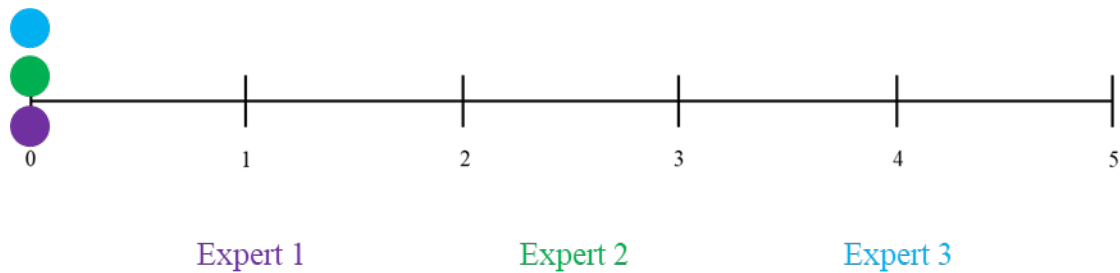


Fig. 11: Concordance of TRIZ/TIPS with the agile management style

(Own representation)

This criterion was the only one in the entire catalogue where all three of the experts agreed in their opinions: according to their statements, it cannot be applied to TRIZ/TIPS. This was justified by the fact that management is not a part of the teachings of TRIZ/TIPS.

5.2.4 Culture



Fig. 12: Concordance of TRIZ/TIPS with the agile culture

(Own representation)

In the opinion of experts 1 and 2, the work culture of TRIZ/TIPS is mostly or just as highly cooperative as the one of agile methods. Both justified this with the close cooperation of the developers recommended by the method and stated that even though it is possible to apply TRIZ/TIPS by oneself, it is not as beneficial. One expert even reported that, according to their experience, the best results are achieved when working together closely across company borders which, however, entails challenges in terms of patent law.

Expert 3 differed in his opinion from those of his fellow professionals and evaluated the culture of TRIZ/TIPS as much less cooperative than the agile method one's. They justified their evaluation with the merely at the beginning, as part of the innovation checklist,

existing examination of the needs of the customers and complained about leaving the developers alone in the further course of the process.

5.2.5 Attitude towards change

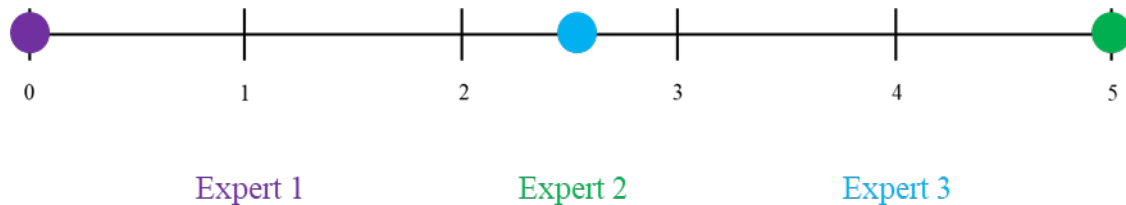


Fig. 13: Concordance of TRIZ/TIPS with the agile attitude towards change

(Own representation)

The opinions of the different experts about the conformity of TRIZ/TIPS with this agile criterion were very different and spanned the entire evaluation scale. Expert 1 chose zero on the scale and, therewith, said that the criterion was not applicable to TRIZ/TIPS. They justified this, stating that attitude towards change is not a part of the teachings of TRIZ/TIPS.

Expert 2, in turn, chose five, expressing that this criterion fully applies to TRIZ/TIPS. According to them, TRIZ/TIPS is a method to create change and transformation and tells the user in many different ways what can be changed—it, thus, is not at all change-averse, but much rather change friendly.

Expert 3 was kind of in the middle of the scale and assessed this criterion to apply a little to TRIZ/TIPS. As per their justification, the typical user is open to changes. Classic TRIZ/TIPS, however, is basically a waterfall model—at the beginning, the desired changes are queried with the innovation checklist, but then a linear process is initiated. Later in the process, it is very difficult to deviate from the chosen path to respond to changed circumstances.

5.2.6 Documentation

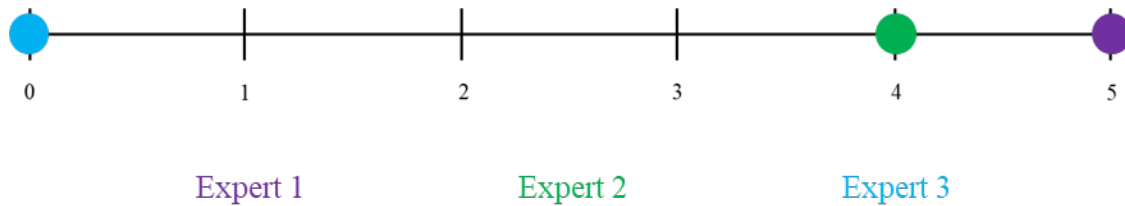


Fig. 14: Concordance of TRIZ/TIPS with the agile documentation

(Own representation)

As per the opinions of experts 1 and 2, the nature of documentation of TRIZ/TIPS is exactly or mostly the same as the agile methods' one. According to both of them, the documentation of the problem-solving process is secondary and by no means excessively stands in focus. Instead, as with agile methods, what really matters is the result at the end of the process.

Expert 3, in contrast, rated this criterion zero, expressing that it is not applicable to TRIZ/TIPS. According to their explanation, the manner of documentation is not dictated by the method itself and is not part of the teachings of TRIZ/TIPS, but depends much more on the personal style of every individual user.

5.2.7 Emphasis

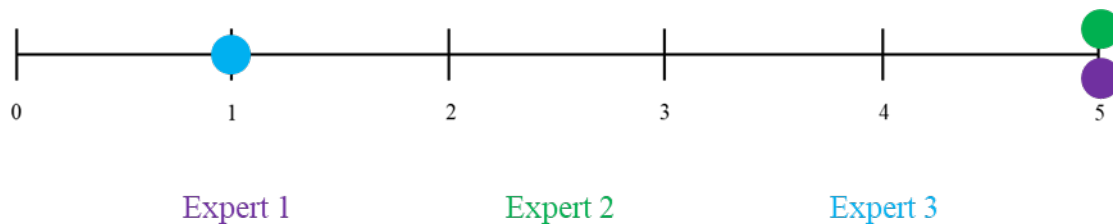


Fig. 15: Concordance of TRIZ/TIPS with the agile emphasis

(Own representation)

Both expert 1 as well as 2 were of the opinion, that this criterion fully applies to TRIZ/TIPS. Both argued that TRIZ/TIPS is a method to solve problems and generate ideas, which can only be achieved if the developers are qualified enough and have the

appropriate expertise—therefore, the method definitely is people-centred. One of the experts added that, as is with agile methods, with TRIZ/TIPS it should be avoided under all circumstances that incompetence becomes part of the team.

Selecting one on the scale and therewith saying that this criterion does not apply to TRIZ/TIPS at all, the third expert had a completely different opinion than the two others. According to them, the human being does not play a role in TRIZ/TIPS at all—the method is very technocratic. As the reason behind this, they pointed out the time when TRIZ/TIPS was being developed: in those days, Stalinism prevailed in the Soviet Union and the needs of the individual did not matter—this can still be observed with the method to this day.

5.2.8 Process

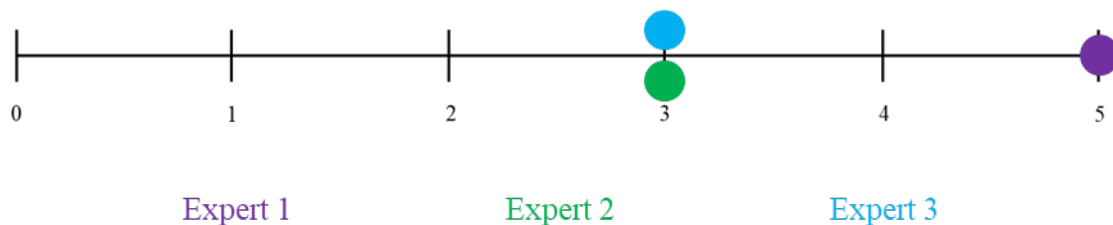


Fig. 16: Concordance of TRIZ/TIPS with the agile process

(Own representation)

For the nature of the process, expert 1 chose five on the scale, claiming that similar to agile methods, TRIZ/TIPS is an empirical process. This was justified by the fact that even though the method provides a certain framing in which the problem-solving process takes place, each individual one is different.

The other two experts, however, differed in their opinions, both declaring this criterion to apply to TRIZ/TIPS only partially. The two alike explained that no problem-solving process is ever the same and the TRIZ/TIPS user has to select the appropriate tools depending on the specific problem to be solved—once he has made a selection, though, there are strict instructions and defined steps. Additionally, after the initial stages of the process, the typical user apparently falls into a standard procedure relatively quickly.

5.2.9 Planning style

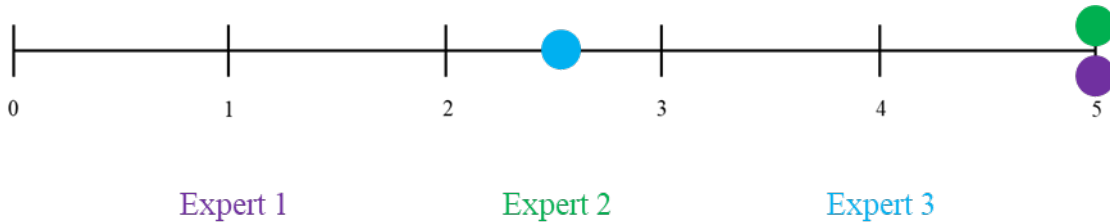


Fig. 17: Concordance of TRIZ/TIPS with the agile planning style

(Own representation)

As stated by experts 1 and 2, this criterion fully applies to TRIZ/TIPS. While conventional projects are comprehensively planned from start to finish in advance, projects with TRIZ/TIPS are only ever planned very short-term akin to agile methods. Consequently, its planning style can definitely be considered minimal.

In contrast to their opinion, expert 3 assessed this criterion to only apply a little to TRIZ/TIPS. As reported by them, the method's planning style can feasibly be arranged minimal, yet many users excessively plan the entire approach at the beginning of each problem-solving process.

5.2.10 Mindset

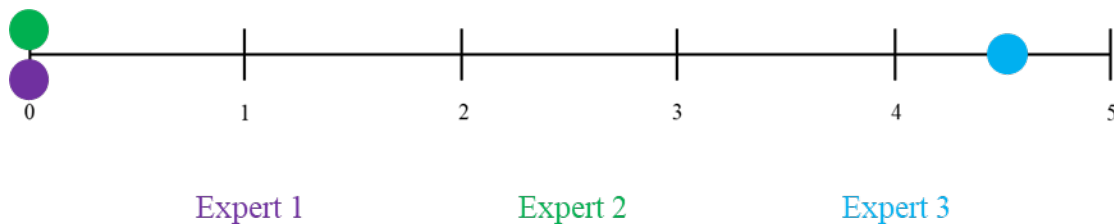


Fig. 18: Concordance of TRIZ/TIPS with the agile mindset

(Own representation)

Both experts 1 and 2 were on par and said that this criterion cannot be applied to TRIZ/TIPS. This was justified by the fact that the method simply is a toolbox and a certain mindset is not a part of the teachings of TRIZ/TIPS.

Opposingly, expert 3 attested TRIZ/TIPS to have a very characteristic behaviour of striving for efficiency and high quality just like agile methods and, on account of this, affirmed that this last criterion broadly applies. As explained by them, TRIZ/TIPS is very much

geared towards efficiency, and those who use the method do so because of the high quality of the solutions that result from it.

6 Conclusion

Fig. 19 shows a summary of the results from the individual expert interviews regarding the concordance of TRIZ/TIPS with the in part 4 of this thesis developed agile characteristics. Indubitably, with three interviewees, this expert consultation does not provide a representative result, it nonetheless does serve as an indication though. In any case, TRIZ/TIPS undoubtedly possesses agile characteristics and can, thus, be definitely called partially agile. On all accounts, the entanglement of TRIZ/TIPS and agile methods seems to be a controversial topic. In almost all agile criteria to be evaluated the opinions of the interviewed experts differed from one another—sometimes more, sometimes less. What seems to be clearly recognizable, however, are two different streams of opinion: on the one hand, the opinion that TRIZ/TIPS can be considered agile for the most part and some aspects are just not part of its teachings and on the other hand, the opinion that TRIZ/TIPS can only be considered agile to a very little extent. Future academic work could conduct a more extensive consultation of experts to obtain representative results and explore the topic in more detail. Furthermore, it could be tried to purposefully extend the classical TRIZ/TIPS with agile elements and approaches and, thus, enhance it.

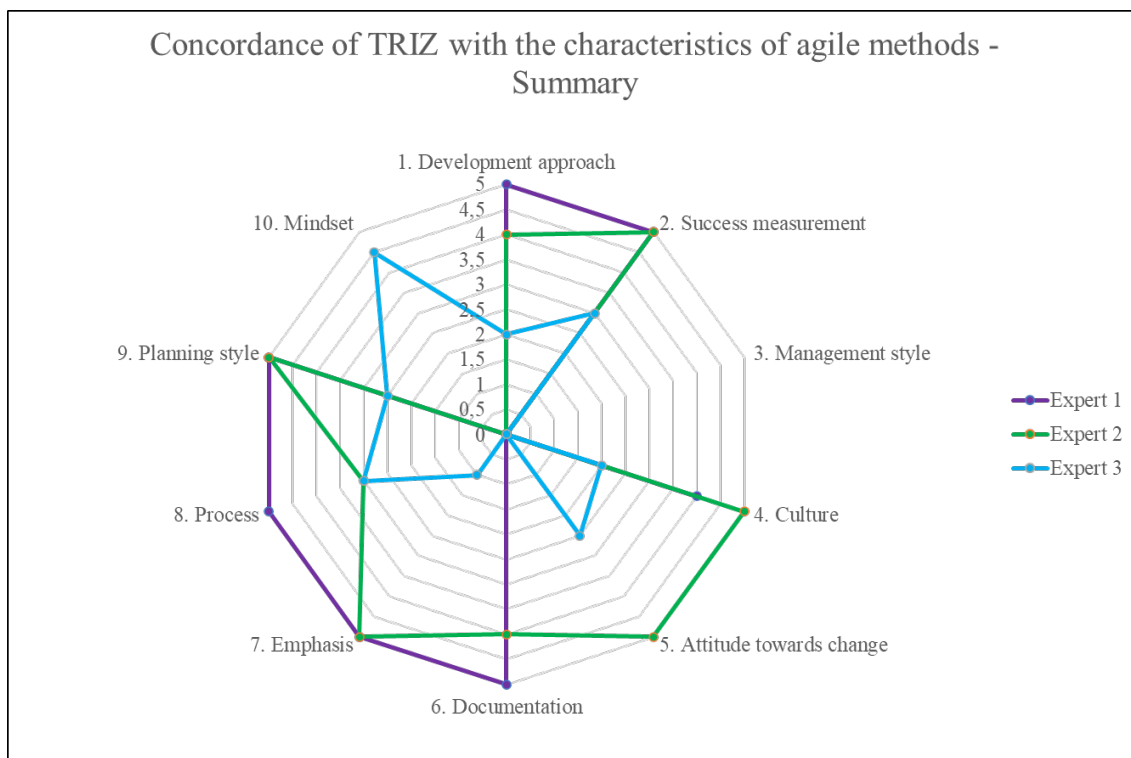


Fig. 19: Concordance of TRIZ/TIPS with the characteristics of agile methods – Summary (Own representation)

7 References

- Agile Alliance. (2020). *Agile 101*. Retrieved June 23, 2020, from <https://www.agilealliance.org/agile101/>
- Atlassian. (2020). *Jira Software*. Retrieved June 10, 2020, from <https://www.atlassian.com/software/jira>
- Awad, M. A. (2005). *A comparison between agile and traditional software development methodologies* [Dissertation, The University of Western Australia, Australia]. RIS.
- Bayer, T. (2019). *Ausarbeitung einer potenziellen Methodenintegration aus der TRIZ Lehre in den Design Thinking Prozess* [Bachelor's Thesis, HAW Landshut, Landshut]. <https://opus4.kobv.de/opus4-haw-landshut/frontdoor/index/index/docId/133>
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A [Alistair], Cunningham, W., Fowler, M., Grenning, J., Highsmith, J [Jim], Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. Retrieved August 28, 2020, from <https://agilemanifesto.org/>
- Boehm, B., & Turner, R. (2005). Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Software*, 22(5), 30–39. <https://doi.org/10.1109/ms.2005.129>
- Casner, D., Souili, A., Houssin, R., & Renaud, J. (2018). Agile' TRIZ Framework: Towards the Integration of TRIZ Within the Agile Innovation Methodology. *IFIP Advances in Information and Communication Technology*, 541, 84–93. https://doi.org/10.1007/978-3-030-02456-7_8
- Chechurin, L., & Borgianni, Y. (2016). Understanding TRIZ through the review of top cited publications. *Computers in Industry*, 82, 119–134. <https://doi.org/10.1016/j.compind.2016.06.002>
- Cockburn, A [A.], & Highsmith, J [J.] (2001). Agile Software Development: The People Factor. *Computer*, 34(11), 131–133. <https://doi.org/10.1109/2.963450>
- CollabNet VersionOne. (2020). *DevOps and Agile Solutions for the Enterprise*. Retrieved June 9, 2020, from <https://www.collab.net/>
- Cooke, J. L. (2010). *Agile Principles Unleashed: Proven Approaches for Achieving Real Productivity in Any Organisation*. IT Governance Publishing.

- Digital.ai. (2020a). *14th Annual State of Agile Report*. Retrieved June 9, 2020, from <https://stateofagile.com/#>
- Digital.ai. (2020b). *14th annual State of Agile Report*. Retrieved June 9, 2020, from <https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report>
- Digital.ai. (2020c). *Intelligent Value Stream Management Platform*. Retrieved June 9, 2020, from <https://digital.ai/>
- Gadd, K. (2011). *Triz for engineers: Enabling inventive problem solving*. Wiley.
<https://doi.org/10.1002/9780470684320>
- Hentschel, C., Gundlach, C., & Nähler, H. T. (2010). *TRIZ - Innovation mit System* (1. Aufl.). Carl Hanser Fachbuchverlag. <https://doi.org/10.3139/9783446426290>
- Highsmith, J [J.]. (2006). *Agile Software Development Ecosystems: The Agile Software Development Series* (3. printing). *The agile software development series*. Addison-Wesley.
<https://books.google.de/books?id=uE4FGFOHs2EC&printsec=frontcover&hl=de#v=onepage&q&f=false>
- Highsmith, J [J.], & Cockburn, A [A.] (2001). Agile Software Development: The Business of Innovation. *Computer*, 34(9), 120–127. <https://doi.org/10.1109/2.947100>
- Hohl, P., Klünder, J., van Bennekum, A., Lockard, R., Gifford, J., Münch, J., Stupperich, M., & Schneider, K. (2018). Back to the future: origins and directions of the “Agile Manifesto” – views of the originators. *Journal of Software Engineering Research and Development*, 6(15), 1–27. <https://doi.org/10.1186/s40411-018-0059-z>
- Horney, N., Pasmore, B., & O'Shea, T. (2010). Leadership agility: A business imperative for a VUCA world. *People & Strategy*, 33(4), 32–38. <http://luxorgroup.fr/coaching/wp-content/uploads/Leadership-agility-model.pdf>
- IDEATION INTERNATIONAL INC. (2020). *Software*. Retrieved June 2, 2020, from <http://www.whereinnovationbegins.net/software/>
- Ilevbare, I. M., Probert, D., & Phaal, R. (2013). A review of TRIZ, and its benefits and challenges in practice. *Technovation*, 33(2-3), 30–37. <https://doi.org/10.1016/j.technovation.2012.11.003>
- Joiner, B. (2019). Leadership Agility for Organizational Agility. *Journal of Creating Value*, 5(2), 139–149. <https://doi.org/10.1177/2394964319868321>

- Kiesel, M., & Hammer, J. (2018). Triz – Develop or Die in a World Driven by Volatility, Uncertainty, Complexity and Ambiguity. *IFIP Advances in Information and Communication Technology*, 541, 55–65. https://doi.org/10.1007/978-3-030-02456-7_5
- Klein, B. (2014). *TRIZ/TIPS: Methodik des erfinderischen Problemlösens* (3. Aufl. [elektronische Ressource]). *Studium*. De Gruyter Oldenbourg. <https://doi.org/10.1515/9783486859126>
- Koltze, K., & Souchkov, V. (2017). *Systematische Innovation: TRIZ-Anwendung in der Produkt- und Prozessentwicklung* (2., überarbeitete Auflage). *Praxisreihe Qualitätswissen*. Hanser. <https://doi.org/10.3139/9783446452572>
- Komus, A., & Kuberg, M. (2017). *Status Quo Agile 2016/2017: 3. Studie über Erfolg und Anwendungsformen von agilen Methoden*. Retrieved June 15, 2020, from https://www.hs-koblenz.de/fileadmin/media/fb_wirtschaftswissenschaften/Forschung_Projekte/Forschungsprojekte/Status_Quo_Agile/Studie_2016/SQA-Abschlussbericht-de_V.1.02_-_Highlights.pdf
- Kropp, M., Anslow, C., Mateescu, M., Burkhard, R., Vischi, D., & Zahn, C. (2017). Enhancing Agile Team Collaboration Through the Use of Large Digital Multi-touch Cardwalls. *Agile Processes in Software Engineering and Extreme Programming. XP 2017. Lecture Notes in Business Information Processing*, 283, 119–134. https://doi.org/10.1007/978-3-319-57633-6_8
- Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, 36(6), 47–56. <https://doi.org/10.1109/MC.2003.1204375>
- Layton, M. C., & Ostermiller, S. J. (2018). *Agiles Projektmanagement für dummies®*. Wiley-VCH Verlag GmbH & Co. KGaA.
- Mareis, C. (2015). TRIZ in the Aftermath of a Transnational Post-war History. *Procedia Engineering*, 131, 500–508. <https://doi.org/10.1016/j.proeng.2015.12.444>
- Misra, S., Kumar, V., Kumar, U., Fantazy, K., & Akhter, M. (2012). Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*, 29(9), 972–980. <https://doi.org/10.1108/02656711211272863>

- Moehrle, M. G. (2005). How combinations of TRIZ tools are used in companies – results of a cluster analysis. *R&D Management*, 35(3), 285–296.
<https://doi.org/10.1111/j.1467-9310.2005.00390.x>
- Moniruzzaman, A. B. M., & Hossain, S. A. (2013). Comparative Study on Agile software development methodologies. *Global Journal of Computer Science and Technology*, 13(7), 1–25. <https://arxiv.org/pdf/1307.3356>
- Orloff, M. A. (2006). *Grundlagen der klassischen TRIZ: Ein praktisches Lehrbuch des erfinderischen Denkens für Ingenieure* (3rd ed.). VDI-Buch. Springer Berlin Heidelberg.
- Pfeffer, J. (2020). *Produkt-Entwicklung Lean & Agile*. Carl Hanser Verlag.
- Saleh, A., & Watson, R. (2017). Business excellence in a volatile, uncertain, complex and ambiguous environment (BEVUCA). *The TQM Journal*, 29(5), 705–724.
<https://doi.org/10.1108/TQM-12-2016-0109>
- Solinski, A., & Petersen, K. (2016). Prioritizing agile benefits and limitations in relation to practice usage. *Software Quality Journal*, 24, 447–482.
<https://doi.org/10.1007/s11219-014-9253-3>
- Souchkov, V. (2016). TRIZ in the World: History, Current Status, and Issues of Concern. In *the 8th International Conference “TRIZ: Application Practices and Development Issues” on November 11-12, 2016*. https://www.researchgate.net/publication/333055843_TRIZ_in_the_World_History_Current_Status_and_Issues_of_Concern
- Sprefafico, C., & Russo, D. (2016). TRIZ Industrial Case Studies: A Critical Survey. *Procedia CIRP*, 39, 51–56. <https://doi.org/10.1016/j.procir.2016.01.165>
- TriS Europe. (2019). *Computer-Aided Innovation für starke Produktideen*. Retrieved June 8, 2020, from <http://www.tris-europe.com/software/innovationssoftware.htm>
- Wang, H., Lin, H., & Yang, J. (2013). History, Current Situation and Development Prospect of the TRIZ. *Advanced Materials Research*, 798, 992–997.
<https://doi.org/10.4028/www.scientific.net/AMR.798-799.992>
- Williams, L., & Cockburn, A [A.] (2003). Agile Software Development: It's about Feedback and Change. *Computer*, 36(6), 39–43.
<https://doi.org/10.1109/mc.2003.1204373>

Zobel, D. (2007). *TRIZ FÜR ALLE: Der systematische Weg zur Problemlösung Mit 61 Bildern und 7 Tabellen* (2., durchges. Aufl.). expert verlag.

Appendices

Appendix A: Guideline for the expert interviews (German)

Forschungsfrage

In welchem Ausmaß ist TRIZ eine agile Methode?/Inwiefern besitzt TRIZ agile Charakteristika?

Einstieg

- Begrüßung des Interviewpartners und Dank für die Teilnahme
- Schutz der persönlichen Daten und Anonymität der Untersuchung
- Darf das Gespräch aufgezeichnet werden?
- Beschreibung des Themas: Forschungsfrage, Kriterienkatalog
- Beschreibung der Intention des Interviews: Inwieweit sind die agilen Kriterien/Merkmale auf TRIZ anwendbar?, Expertenmeinung zum Maß an Agilität von TRIZ bezüglich der Kriterien

Einstiegsfragen

- Welches MATRIZ Level haben Sie?
- Wie lange arbeiten Sie sich schon mit TRIZ?
- Haben Sie Erfahrung mit agilen Methoden? Wenn ja, mit welchen?
- In welcher Form haben Sie Erfahrung mit agilen Methoden (Coach, Workshopteilnehmer, Anwender, theoretische Grundkenntnisse)?
- Wie lange beschäftigen Sie sich schon mit agilen Methoden?
- Wie zeichnen sich für Sie agile Methoden aus?

Schlüsselfragen

- Nun folgt die Bewertung der Agilität von TRIZ auf Basis des von mir erstellten Kriterienkatalogs für agile Methoden - zunächst werde ich dafür das jeweilige Kriterium definieren. Dann bitte ich Sie, TRIZ in Bezug auf das Kriterium auf einer Skala von 0 bis 5 zu bewerten. Dabei bedeutet:

0: „Das Kriterium ist nicht für TRIZ anwendbar.“

1: „Das Kriterium trifft überhaupt nicht auf TRIZ zu.“

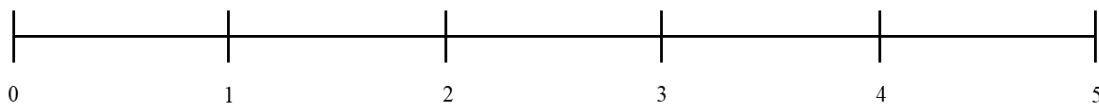
3: „Das Kriterium trifft teilweise auf TRIZ zu.“

5: „Das Kriterium trifft voll und ganz auf TRIZ zu.“

Zusätzlich zu dieser Einschätzung bitte ich Sie, jeweils eine Begründung für Ihre Auswahl zu geben.

1. Entwicklungsansatz

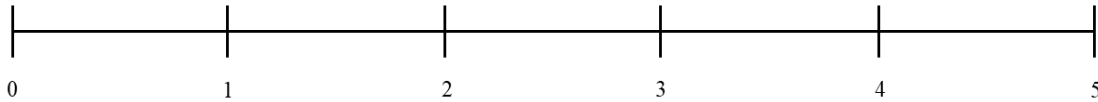
Agile Methoden sind in hohem Maße anpassungsfähig und reaktionsfähig im Gegensatz zu konventionellen Methoden, welche versuchen, den gesamten Prozess im Voraus zu antizipieren und vorherzusagen. Das wird durch eine lediglich minimale und kurzfristige Planung sowie einer Entwicklung in kurzen Iterationen erzielt.



Begründung:

2. Erfolgsmessung

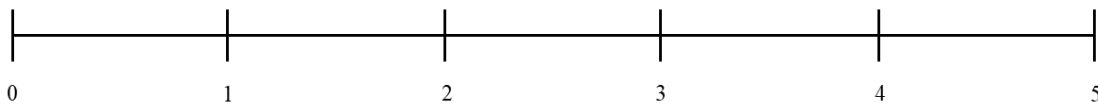
Der primäre Maßstab für den Erfolg agiler Entwicklung sind reale, greifbare Ergebnisse und nicht die Übereinstimmung mit dem Plan. Das zugrundeliegende Verständnis ist dabei: Pläne werden obsolet und falsch, funktionierende Produkte, die den Kunden zufrieden stellen, sind immer richtig.



Begründung:

3. Art des Managements

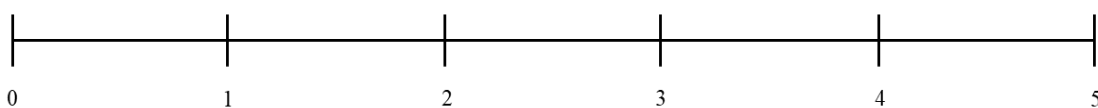
In der agilen Entwicklung ist der Managementstil dezentralisiert und nicht autoritativ, wie es bei den meisten traditionellen Projekten der Fall ist. Konkret bedeutet dies, dass die Projektleiter die Entscheidungsautorität über den eigentlichen Entwicklungsprozess auf die Entwickler übertragen, weil diese am besten wissen, was wann getan werden muss um die Anforderungen des Kunden zu erfüllen. Das Management unterstützt sie dabei und erledigt die wirtschaftlichen Aufgaben des Projekts.



Begründung:

4. Arbeitskultur

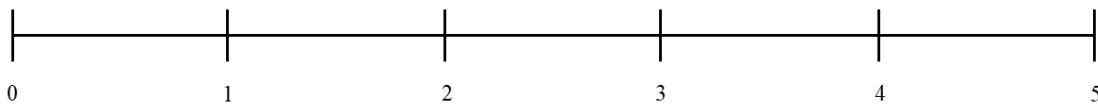
Die agile Arbeitskultur ist hoch kooperativ. Die enge Zusammenarbeit mit dem Kunden trägt dazu bei, immer auf dem richtigen Kurs zu bleiben und das ständige Feedback während des Prozesses verhindert ein Abkommen vom richtigen Weg. Die enge Zusammenarbeit zwischen den Geschäftsleuten und den Entwicklern eines agilen Projekts hält beide Seiten stets auf dem Aktuellen bezüglich sowohl Aspekten der Entwicklung als auch des Geschäftlichen. Die enge Zusammenarbeit zwischen den Entwicklern selbst erhöht das Niveau, auf dem das Team in Bezug auf Tempo und Qualität arbeitet, erheblich.



Begründung:

5. Einstellung gegenüber Änderungen

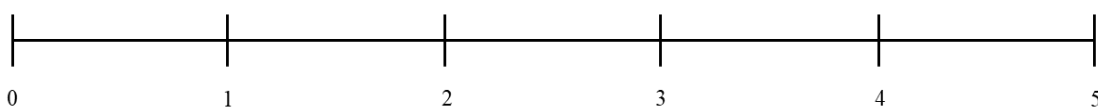
Während traditionelle Herangehensweisen eher veränderungsfeindlich sind, begrüßen agile Methoden Änderungen/Wandel. Dies gilt nicht nur für die frühen Phasen eines Projekts: Stattdessen besteht einer der wichtigsten Grundsätze der Agilität darin, geänderte Anforderungen in allen Phasen eines Projekts zu begrüßen und zu fördern - auch in späten. Agile Teams schaffen somit einen entscheidenden Wettbewerbsvorteil für ihre Kunden, die in einem turbulenten Geschäftsumfeld agieren und daher eine gewisse Flexibilität benötigen und sind gerade deshalb auch selbst erfolgreich.



Begründung:

6. Dokumentation

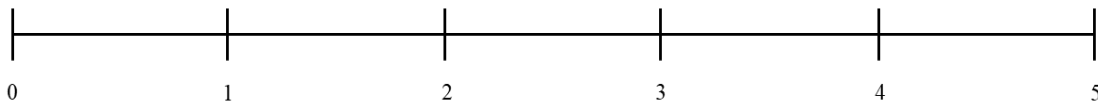
Wie im Agilen Manifest skizziert, sind reale Produkte wichtiger als eine umfassende Dokumentation - mit anderen Worten: Die Dokumentation leichtgewichtig zu halten ist der agile Weg, anstatt großen Wert darauf zu legen alles eines Projekts zu dokumentieren, wie es konservative Projekte tun.. Je umfangreicher Projektdokumentation wird, desto komplizierter ist es, die benötigten Informationen zu finden und alles stets auf dem neuesten Stand zu halten. Durch die Minimierung der Dokumentation, die - vom agilen Standpunkt aus gesehen - Arbeit bedeutet, die dem eigentlichen Produkt keinen Mehrwert bringt, wird der Ressourcenaufwand erheblich reduziert.



Begründung:

7. Hauptaugenmerk

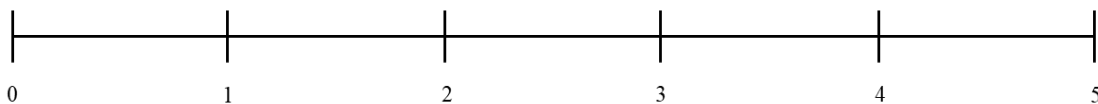
Für Agilisten sind Menschen und deren Interaktionen wichtiger als Prozesse und Werkzeuge. Während also traditionelle Entwicklungsansätze prozesszentriert sind, stehen bei agilen Ansätzen die Menschen im Mittelpunkt. Das zugrundeliegende Verständnis ist dabei, dass Menschen die an einem Projekt arbeiten jegliche Prozesse und Werkzeuge zur Erfüllung ihrer Aufgaben nutzen können, wenn sie qualifiziert genug sind - wenn sie in ihrer Arbeit nicht qualifiziert genug sind, wird jedoch kein spezifischer Prozess oder Werkzeug ihr fehlendes Wissen ausgleichen.



Begründung:

8. Art des Prozesses

Agile Prozesse werden als empirische Prozesse betrachtet, da kein Prozess jemals gleich ist. Vordefinierte Schritte wie in definierten Prozessen würden nicht zu wünschenswerten Ergebnissen führen, die den Kunden zufrieden stellen: Das volatile Umfeld, in dem agile Entwicklung stattfindet und für das sie erfunden wurde, erfordert eine ständige Reaktionsfähigkeit und die Fähigkeit zur Anpassung an die sich ständig ändernden Umstände.

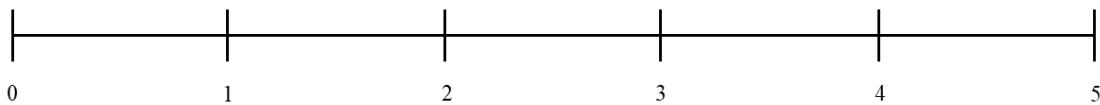


Begründung:

9. Planungs-Stil

Während traditionelle Entwicklungsprojekte im Vorfeld umfassend geplant werden, erfordern agile Projekte nur minimale Planung. Dies beruht auf dem bereits

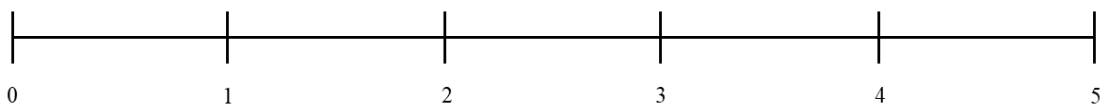
erläuterten hochgradig anpassungsfähigen und veränderungsfreundlichen Charakter agiler Methoden. Der Versuch, den gesamten agilen Prozess im Voraus zu planen, würde bedeuten, dass der Plan aufgrund der unvermeidlichen, andauernd stattfindenden Veränderungen ständig aktualisiert werden müsste - es gibt einfach viel zu viele potenziell veränderliche Variablen, um sie alle berücksichtigen zu können.



Begründung:

10. Einstellung

Agile Teams sind motiviert und streben ständig nach Effizienz und hoher Qualität in ihrer Arbeit. Agilisten wissen, dass ununterbrochene Aufmerksamkeit auf die Qualität ihrer Arbeit ihr Level an Agilität positiv beeinflusst, wie im Agilen Manifest betont. Darüber hinaus kommen sie in regelmäßigen Abständen mit der Absicht zusammen, über ihre Arbeit zu reflektieren sowie gemeinsam Wege zu finden, effizienter zu werden und passen ihr Arbeitsverhalten anschließend entsprechend an. Unterstützt wird dies alles durch ein hohes vorherrschendes Level an Motivation, die durch die agile Art des Managements hervorgerufen wird.



Begründung:

Schlussfragen

- Möchten Sie noch zusätzliche Aspekte/Ideen/Einwände nennen oder gibt es etwas, das Ihrem Gefühl nach vergessen/vernachlässigt wurde?

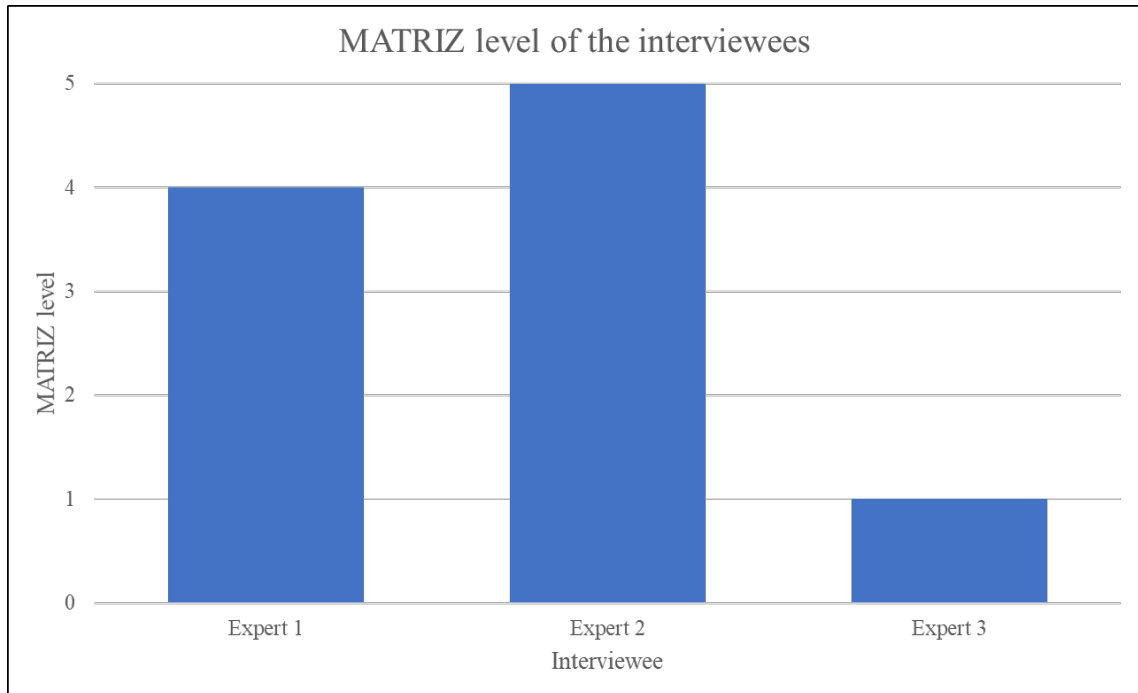
Ende

- Erneuter Dank für das Zeitnehmen
- Darf ich Sie bei eventuellen Rückfragen erneut kontaktieren?
- Verabschiedung

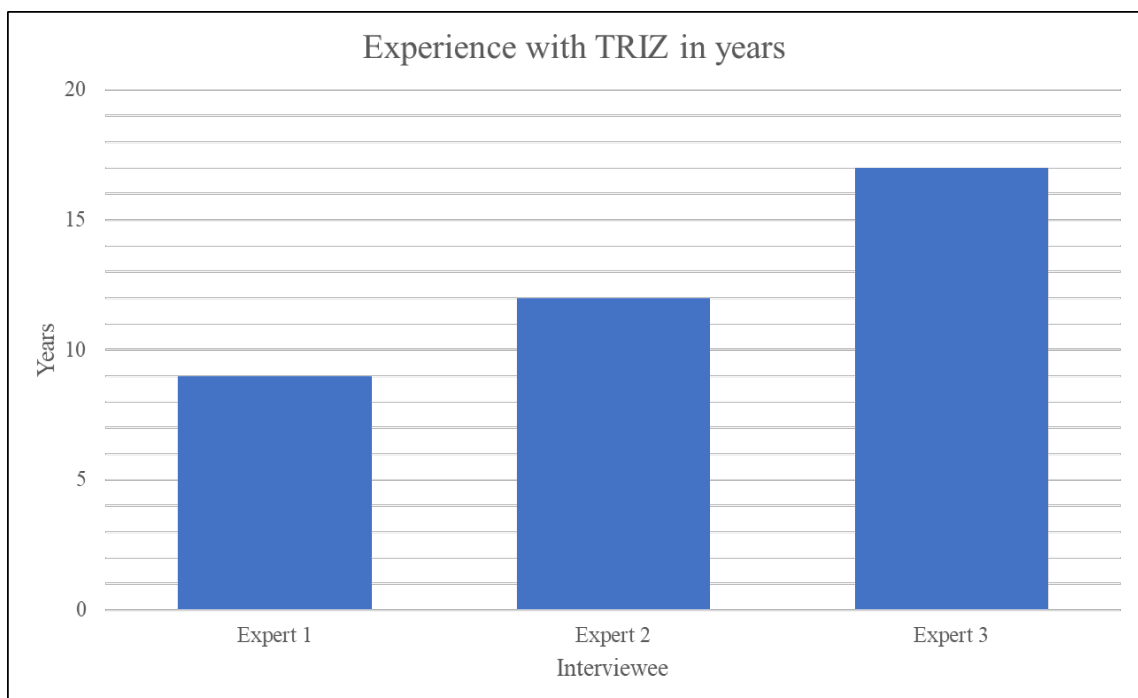
Appendix B: Evaluation of the filled-out guidelines of the expert interviews

Evaluation of the opening questions

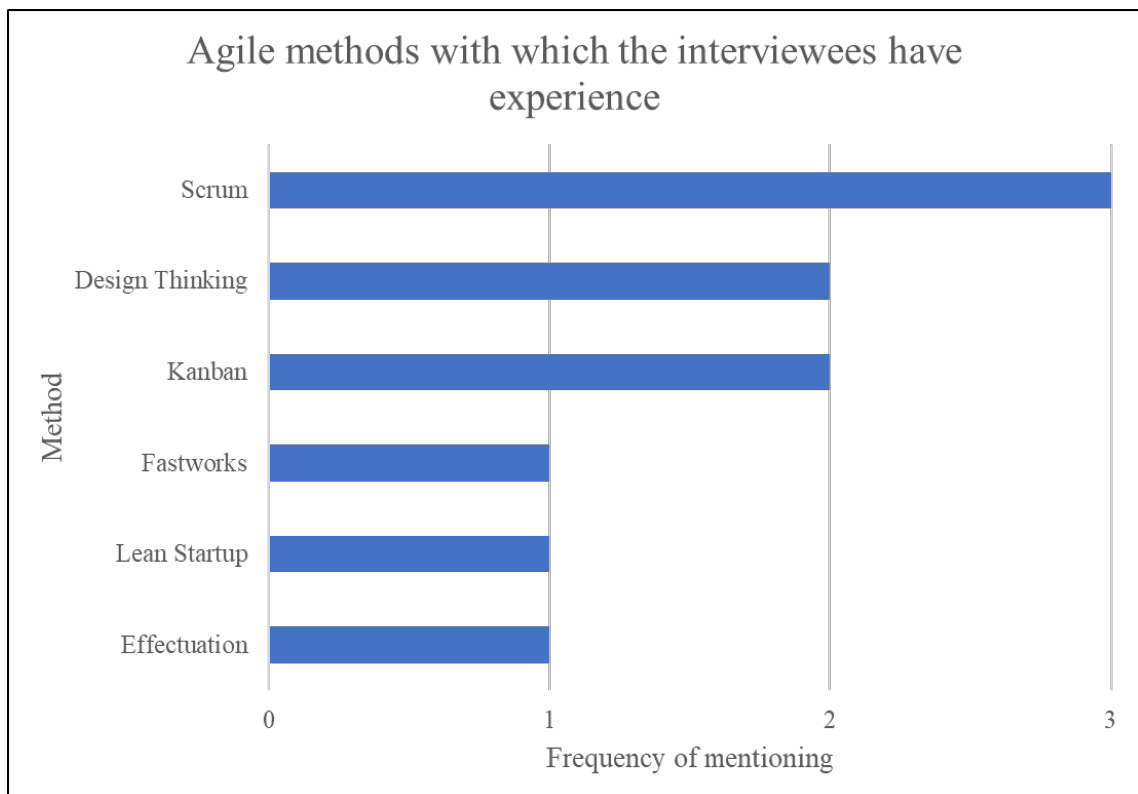
- What MATRIZ level do you have?



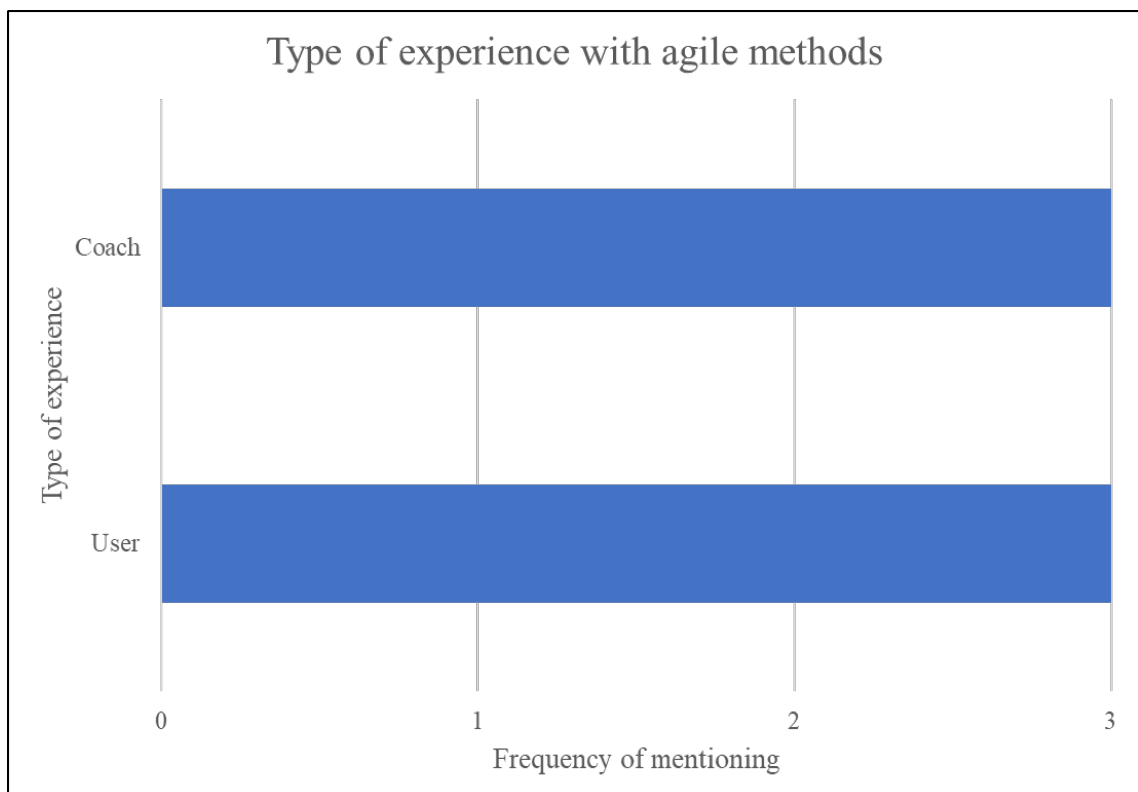
- How long have you been working with TRIZ?



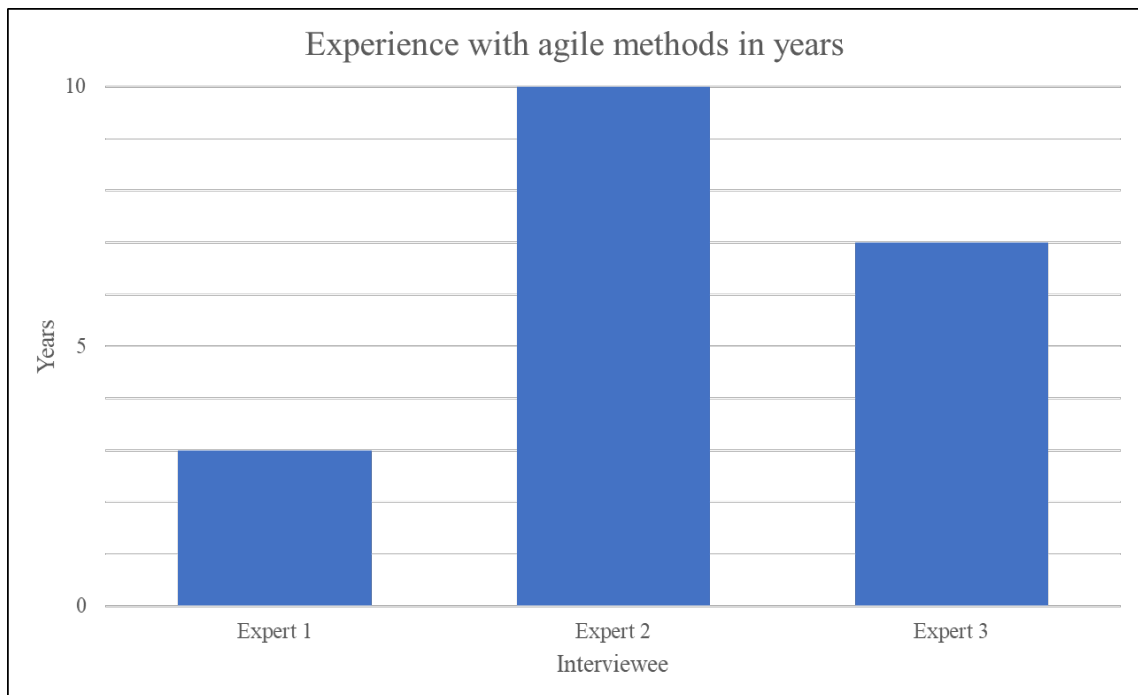
- Do you have experience with agile methods? If so, with which ones?



- In what way do you have experience with agile methods? (Coach, workshop participant, user, basic theoretical knowledge)



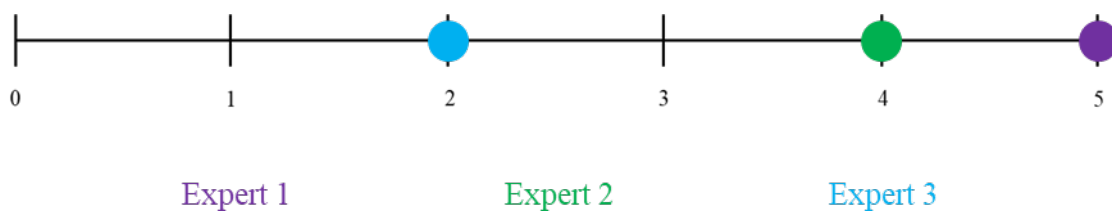
- How long have you been working with agile methods?



- What characterizes agile methods for you?
 - Expert 1: Mainly their flexibility, their ability to adapt to changed circumstances, as well as the iterative and incremental development approach.
 - Expert 2: Agile methods provide a certain framing, similar to the crash barriers of a multi-lane road: they prevent vehicles from straying from the road but leave the choice of what lane to use to the driver.
 - Expert 3: If the agile values and principles are met, a method or process can be considered agile. Some characteristics are result-oriented instead of process-oriented, pragmatic, fast-paced, iterative—provided with constant feedback on which is built up incrementally.

Evaluation of the key questions

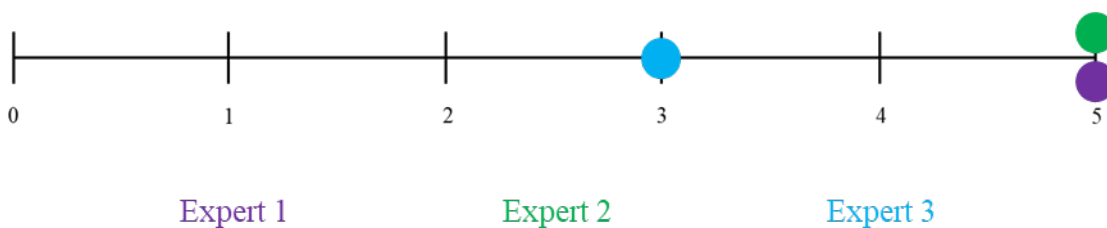
1. Development approach



Justification:

- **Expert 1:** This criterion fully applies to TRIZ. TRIZ essentially works iteratively as well: You start with a problem that you analyze and then you think about how to solve it—similar to the Product Backlog in Scrum. Afterwards you work through the problem-solving process and generate different ideas from which you choose one and create a prototype. During the whole process there are secondary problems which have to be solved, also. The only difference between agile methods and TRIZ is that there is no prescribed systematology in TRIZ—although, if you work with the innovation checklist you could say that you first define the overall problem and in the following step you decide how to work on it.
- **Expert 2:** This criterion mostly applies to TRIZ. TRIZ is a toolbox—you are free in which order you use certain tools and whether you use them at all.
- **Expert 3:** This criterion applies a little to TRIZ. TRIZ itself does not know an iterative procedure and is by no means as adaptive and responsive as agile methods are. However, it offers such a broad toolbox that a certain degree of flexibility is achieved simply by choosing the appropriate tool.

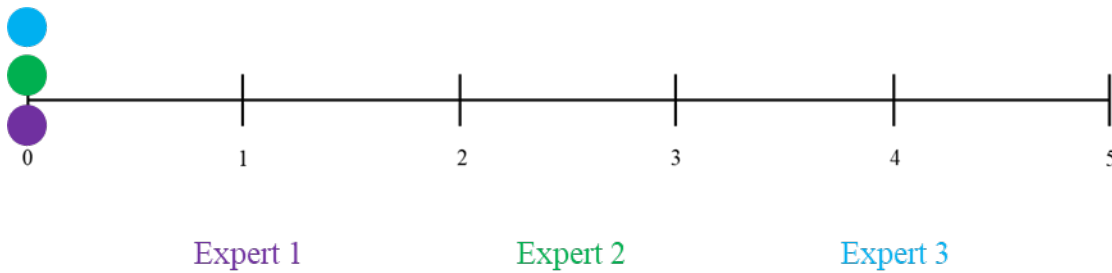
2. Success measurement



Justification:

- **Expert 1:** This criterion fully applies to TRIZ. The decisive factor is the result at the end of the process, not whether a previously created plan has been followed during the process.
- **Expert 2:** This criterion fully applies to TRIZ. With TRIZ it is definitely about the result at the end of the problem-solving process, not about the agreement with a plan.
- **Expert 3:** This criterion partially applies to TRIZ. TRIZ generates extremely inventive ideas and does not focus on conforming to a plan. In practice, however, the user is often overwhelmed by the ideas resulting from the process. This then often leads to not implementing the idea and therefore not creating a real, tangible output—this is also one of the main points of criticism of the method. TRIZ does not pay attention to the result standing at the end of the process but is satisfied as long as an original idea generated.

3. Management style



Justification:

- **Expert 1:** This criterion is not applicable to TRIZ. Management is not a subject of TRIZ.
- **Expert 2:** This criterion is not applicable to TRIZ. Management is not a subject of TRIZ.
- **Expert 3:** This criterion is not applicable to TRIZ. Management is not a subject of TRIZ.

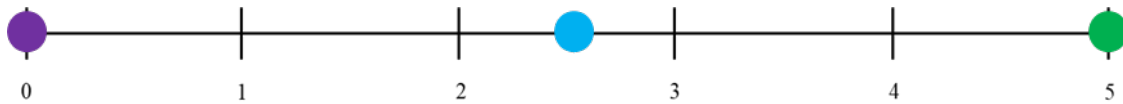
4. Culture



Justification:

- **Expert 1:** This criterion mostly applies to TRIZ. TRIZ recommends a particularly close cooperation of the developers. However, it is also possible to work alone with TRIZ instead of as part of a team (but not so well).
- **Expert 2:** This criterion fully applies to TRIZ. TRIZ is a highly cooperative method and is used best with a team—using it alone brings relatively little benefit. Experience has even shown that the best results are achieved when working together across company borders which, however, entails challenges in terms of patent law.
- **Expert 3:** This criterion applies a little bit to TRIZ. With the innovation checklist, TRIZ has a tool at the beginning with which the needs of the users are queried, but in the further course of the process the developers are left alone. However, the developers cooperate with each other, therefore the work culture as a whole cannot be rated 1.

5. Attitude towards change



Expert 1

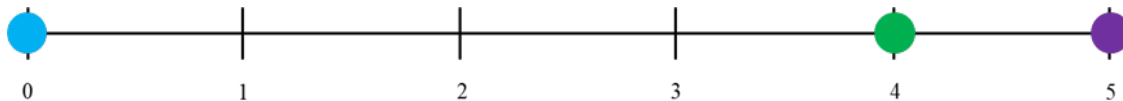
Expert 2

Expert 3

Justification:

- **Expert 1:** The criterion is not applicable to TRIZ. Attitude towards change is not a subject of TRIZ.
- **Expert 2:** This criterion fully applies to TRIZ. TRIZ is a tool to create change and transformation and tells the user in many ways what can be changed. Thus, TRIZ is not at all change averse, but much rather change friendly.
- **Expert 3:** This criterion applies a little to TRIZ. The typical TRIZ user is open to changes. However, classic TRIZ is essentially a waterfall process and hence not really open to change—at the beginning the changes are queried with the innovation checklist but then a linear process is initiated. Later in the process it is very difficult to deviate from the chosen path to respond to changed circumstances.

6. Documentation



Expert 1

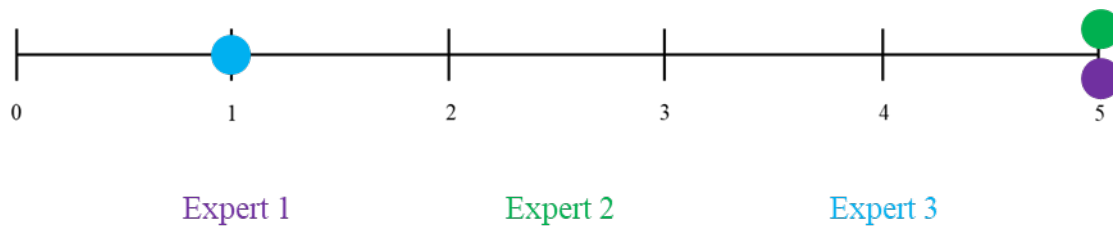
Expert 2

Expert 3

Justification:

- **Expert 1:** This criterion fully applies to TRIZ. The documentation of the problem-solving process is secondary and by no means stands in the foreground—the result at the end of the process is what matters.
- **Expert 2:** This criterion mostly applies to TRIZ. TRIZ is not excessively focused on documentation—some tools do require it, though.
- **Expert 3:** The criterion is not applicable to TRIZ. The manner of documentation is not dictated by TRIZ itself—it depends much more on the personal style of every single TRIZ user.

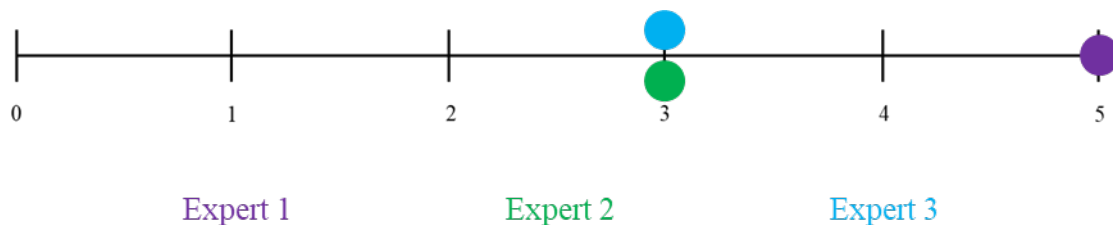
7. Emphasis



Justification:

- **Expert 1:** This criterion fully applies to TRIZ. TRIZ typically asks for contradictions and tries to resolve them—this can only be successful if the user/developer is qualified enough and has the appropriate expertise.
- **Expert 2:** This criterion fully applies to TRIZ. The developers are definitely in the focus of TRIZ because they generate the ideas—therefore, the method is people-centred. As with agile methods, with TRIZ it should be avoided under all circumstances that incompetence becomes part of the team.
- **Expert 3:** This criterion does not apply to TRIZ at all. The human being plays no role in TRIZ or in other words: TRIZ does not care about the human being at all. The method is very technocratic. At the time when TRIZ was developed, Stalinism prevailed in the Soviet Union and the needs of the individual human being did not play a role—this can still be observed with TRIZ today.

8. Process

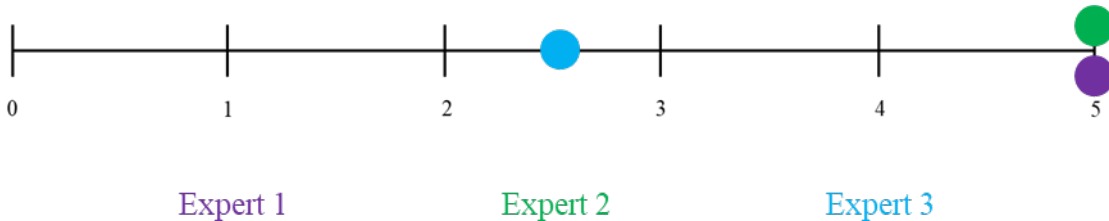


Justification:

- **Expert 1:** This criterion fully applies to TRIZ. As with agile methods, a certain framing is given in which the process takes place, but each individual problem-solving process is different.
- **Expert 2:** This criterion partially applies to TRIZ. TRIZ contains strict instructions and defined steps from the point in time on at which one has made a choice as to which tool to use. Before this point in time, however, each problem-solving process is different.
- **Expert 3:** This criterion partially applies to TRIZ. The TRIZ user must always adapt the tools he chooses for the project, he does not just stubbornly follow the same steps over and over again and must adapt the approach according to the specific characteristics of each problem-solving

process—however, he typically falls into a standard procedure relatively quickly.

9. Planning style



Justification:

- **Expert 1:** This criterion fully applies to TRIZ. In TRIZ you only ever plan the next step—consequently, the planning style can definitely be considered to be minimal.
- **Expert 2:** This criterion fully applies to TRIZ. While conventional projects are completely planned from start to finish in advance, projects with TRIZ are only ever planned very short-term—just like it is the case with agile methods.
- **Expert 3:** This criterion applies a little to TRIZ. Many TRIZ users excessively plan the entire approach at the beginning of each problem-solving process.

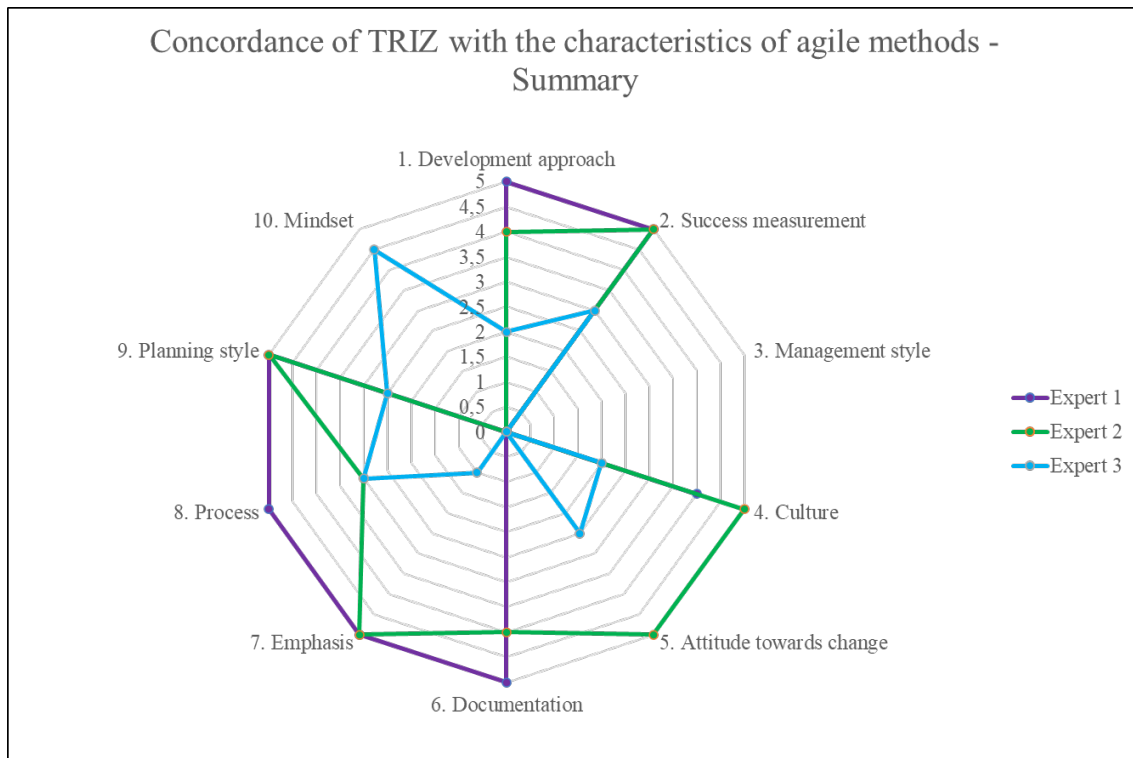
10. Mindset



Justification:

- **Expert 1:** This criterion is not applicable to TRIZ. Cultural factors are not a subject of TRIZ.
- **Expert 2:** This criterion is not applicable to TRIZ. Cultural factors are not a subject of TRIZ.
- **Expert 3:** This criterion largely applies to TRIZ. Striving for efficiency and high quality is very characteristic behaviour for TRIZ. TRIZ is very much geared towards efficiency, and those who use TRIZ do so because of the high quality of the solutions that result from it.

- Summary of the experts' opinions regarding the concordance of TRIZ with the characteristics of agile methods:



Declaration of Academic Integrity

Student: David Fabian Leiner

University of Applied Sciences Landshut

Faculty of Electrical and Industrial Engineering

Course of studies: International Engineering and Management (Bachelor of Engineering)

Hereby, I declare that I have composed the presented thesis independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted to another authority, nor has it been published yet.

Date, Place

David Fabian Leiner

Release Certificate

Student: David Fabian Leiner

Hereby, I declare that this bachelor's thesis can be added to the stock of the university library and

without retention period

or after a retention period of

1 Year

2 Years

3 Years

5 Years

10 Years or longer

may be made accessible via the university library.

Date, Place

David Fabian Leiner