



HAL
open science

Integration of TRIZ Methodologies into the Digital Product Development Process

Vasilii Kaliteevskii, Matvey Bryksin, Leonid Chechurin

► **To cite this version:**

Vasilii Kaliteevskii, Matvey Bryksin, Leonid Chechurin. Integration of TRIZ Methodologies into the Digital Product Development Process. 22th International TRIZ Future Conference (TFC), Sep 2022, Warsaw, Poland. pp.273-284, 10.1007/978-3-031-17288-5_24 . hal-04449792

HAL Id: hal-04449792

<https://inria.hal.science/hal-04449792v1>

Submitted on 15 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Integration of TRIZ methodologies into the digital product development process

Vasilii Kaliteevskii¹, Matvey Bryksin² and Leonid Chechurin¹

¹ Lappeenranta-Lahti University of Technology, 53850 Lappeenranta, Finland

² Arrival UK, Hammersmith, London, United Kingdom

Abstract. Iterative and waterfall software development processes such as V-model or UP (unified process) are widely used in project development. Nevertheless, modern Agile methodologies are aimed to reduce risks and achieve more predictable results in the digital world. Despite the difference between them, the software development life cycle splits into specific deliverables by breaking a project into smaller segments where the TRIZ toolkit can be effectively applied. Application of the TRIZ theory and core techniques can be valuable for new ideas generation and decision-making process at any point of product development from business model development and product vision definition to user experience design, implementation, and go-to-market strategy. As a result, the product quality grows, continuous delivery is accelerated, and new features are introduced to the final user in a more operative manner. The goal of this article is to demonstrate how TRIZ methodologies integration empowers software development for new products release and continuous improvement.

Keywords: TRIZ, Product Design, Product Management, Product Development, Digital Product

1 Introduction

1.1 Addressing the digital product as a technical system

After being challenged with the dot-com bubble burst, the digital products market continued a substantial growth during the last decades with financial, technological and human resources (IT-specialists) [1]. Although, there is a substantial scientific advance in the direction of mathematical algorithms and technical solutions, there is no such a breakthrough in the direction of the digital product design and strategies. Such scientific pieces aiming user experience and digital product value [2] are rather segregated studies than made in systematic manner. At the same time the number of digital products released increased substantially as well as the number of IT-specialists involved in different digital products design, development and management [3]. In the present research authors are presenting the idea of TRIZ toolset to be applied in the framework of digital product development and management.

Nowadays, TRIZ is widely used for different industries (such as automotive, telecom, aeronautics, etc.) and gives them a competitive advantage opening new opportunities for problem-solving [4, 5]. The theory provides a comprehensive set of inventive

standards that engineers can apply for digital product development as well [6-8]. The powerful advantage of TRIZ tools is that they are written without precisely determining the technical system, making them applicable to any technology product [9]. Software development is a subset of modern system engineering and has the same principles for designing and developing digital products.

Thus, a digital product is considered as a technical system but higher solution flexibility because of the absence of constraints from the physical world in the IT domain. Most digital products are not tangible and can only be visible to the final user via a graphical user interface (GUI). There are different approaches to categorizing digital products, but they all contain the same examples:

- **Application software.** Such a solution provides a digital interface for the final user to deliver a value of a product. Examples: website, mobile app, e-commerce service.
- **Software platform.** Such software includes technology for data processing, event streaming, file storage, analytics, and reporting.
- **System software.** Such software includes operating systems, drivers, and utility programs.

Any digital product development process may follow the standard rules from TRIZ theory to increase performance, quality, business KPIs, and other final product parameters.

1.2 TRIZ tools applied for digital technical systems

One of the core ideas of the TRIZ is that all its principles can be formulated in the terms of the industry where they are to be applied. The flexibility of TRIZ tools makes them independent of the digital product domains. Some of the core terms can have different meanings for software development which are described in this section [10].

Ideality. Ideality principle is a measure of the technical system which quantifies different factors of how much value a solution provides to the final user [4]. An increasing degree of ideality makes the product competitive with other market players, which causes the final product to be more profitable. Product managers can measure the same value from a different angle, such as how much the product meets business goals and KPIs.

Ideal final result (IFR). The principle is a formulation of the final solution that delivers the desired result to the end-user. On the one hand, IFR reaches the highest value of ideality. From another hand, IFR does not require adding extra resources or costs. The basic TRIZ principle says that an ideal system is no system, but the function is performed. Paraphrasing such a rule to the software terminology might be that the ideal application software is a zero user experience interface that provides the capability to reach expected user goals. This principle can significantly improve the quality of product strategy as a base for any new product that introduces a unique value proposition to the market. In other words, any digital product should perform a main useful function with the best-expected outcome for the end customer. Such top priority functionality

defines the order for the development team how to deliver such features in the shortest time.

Su-field. It's a model of a minimal technical system that consists of a pair of actors interacting with each other. The Su-Field requires at least two elements made of substance and a field providing interaction between the substance components. Su-Field analysis is a method of abstract modeling of a technical system. Such a model specifies substance components, fields, and interactions between them. Su-field analysis can be applied to resolve the contradiction between User and System and gain new insights.

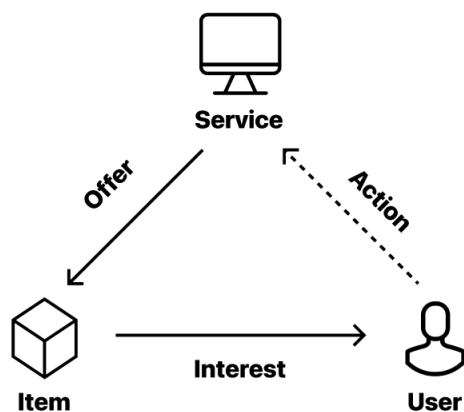


Fig. 1. The Su-field Diagram for a digital service's unique value proposition. Service offers an Item that interests the User to make an Action on the Service.

Su-Field Diagram is an outcome of the Su-Field analysis. Diagram presents a Su-field graphically as a triangle with nodes representing two substances and a field and lines between the nodes representing interactions between the Su-field components. Most software applications require a user interface (UI) for interaction between system and user. The ideal UI is designed to achieve specific user goals. The motivation field can empower the pair of User and System to create a minimal technical system that can be described using a Su-Field diagram from Fig. 1.

Mini-problem and maxi-problem. It's the formulation of an inventive problem that is a result of imposing the following restrictions on this inventive situation: everything remains without any changes or becomes even simpler, but some updates are required. Every decision in software development is a part of continuous integration and improvement. Engineers decide for a specific product component on how they want to

upgrade the system either keep the functionality the same or initiate a new iteration development, depending on the requirements and desired outcome.

2 Software development process and product lifecycle

2.1 Product Discovery and Delivery as main product development stages

Product Lifecycle is the process of product evolution on the entire timeline from its idea through the design and development to the product launch and further updates. The standard industry approach is to slice the lifecycle onto the short implementation iterations [11]. It helps to design and develop the new product in phases. In general, the whole lifecycle from the idea to the final product is split into two main phases – Product Discovery and Product Delivery [12]. Discovery answers how to Build the right product in order to meet the target audience needs. Delivery of how to build the product right. The interfaces between Discovery and Delivery are product backlog and feedback from the market (Fig. 2). They support the linkage between phases and make a smooth transition between them and loopback [13].

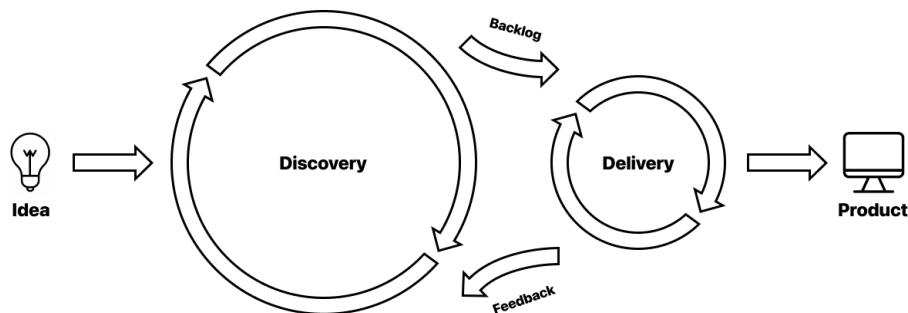


Fig. 2. Product Discovery and Product Delivery – main phases of product continuous improvement.

Product Discovery. The very first strategic phase after a new product idea introduction is Product Discovery [12]. Most of the risks are concentrated in the ideation phase because the cost of an error in the strategic decision can lead to the whole product failure. The Discovery phase starts with generating and prioritizing existing ideas to solve the problem or achieve defined business goals. The Discovery backlog sets the requirement and right focus for the market analysis and needs research. The product manager can develop the product strategy through continuous concept improvements and hypothesis validation cycles that provide a product backlog and roadmap to guide the development team on how to deliver the right product. Such standard outputs in the software industry

are required as minimum knowledge about the product to plan a transition from strategic phase to execution.

Product Delivery. The next phase after Product Discovery is the implementation and execution phase, where the requirements of the desired product are implemented in the software [14]. The main steps are developing products according to specification, running beta or pilot tests, defining a go-to-market strategy, creating a launch plan, and finally releasing the product to the market. During all these incremental steps, it is crucial to gather feedback and monitor major KPIs to adjust product strategy in short periods rapidly [15].

Even though Product Discovery and Delivery are the most critical phases, the standard industry practice is to extend the product lifecycle to the Adoption and end-of-life phases. The broader product lifecycle includes the phase of existing product evolution after introduction to the market. The growth and optimization phases are the sources of product-market fit analysis and valuable feedback for the new iteration of strategy and analysis [16]. The end-of-life phase is a decision point about whether to iterate or decline the product.

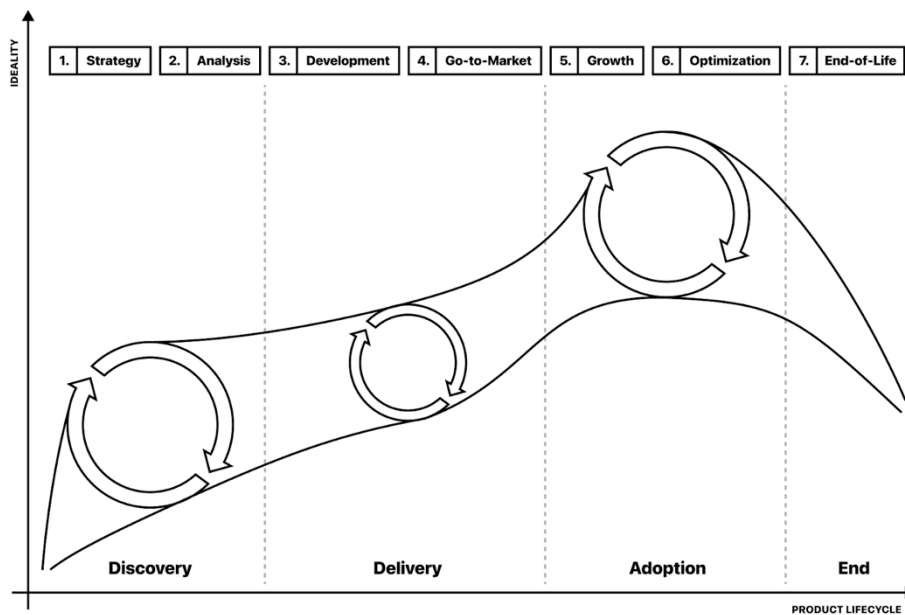


Fig. 3. Phases of product management lifecycle and feedback iterations between them.

2.2 Digital product lifecycle phases overview

Strategy. Undoubtedly the business modeling phase is the most critical starting point for any new digital product. Business goals and objectives are the main drivers of creating a long-term and mid-term product vision without implementation details. As a part of the Product Discovery phase, the major activity and focus are on creating and

prioritizing ideas based on the cost to develop and value to customers or business stakeholders. Development of an initial proposition or draft product concept is required to facilitate detailed analysis and testing.

Analysis. The starting point for detailed product analysis and improvement cycles should be the defined product strategy. Product opportunity assessment requires a deep understanding of user segments and needs through market research and hypothesis analysis. During this phase, it is critical to discover the right value proposition and the problem being solved. Some rapid prototyping techniques are common techniques to validate multiple ideas through continuous feedback loops. Using paper sketches and interactive prototypes increases the quality and speed of gathering early feedback. User research methodologies help validate a market need through customer development. Research processes described in Google Design Sprint, Design Thinking, and Lean UX methodologies help to iterate quickly on the one or two-week iterations to deliver the accurate results of the analysis [17, 18].

The main objective of the Analysis phase is to understand an opportunity for development and estimate costs and risks. In case of success and the decision to move forward, the product backlog and roadmap become the bridge between strategy and delivery. Specification and formulated requirements are the results of the analysis required to create a technical backlog and product plan development.

Development. Getting product requirements delivered is the next phase of the product lifecycle after the confirmation of the product strategy. It starts with designing and delivering a solution to meet the requirements. Some Agile practices are widely used across development teams to achieve more predictable results in the middle of development. The scrum approach for release management helps to set fixed time iterations for designing, engineering, developing, and launching product updates that bring additional value to the market. In this phase of development, inaccurate sprint planning becomes the main point of failure and causes missing deadlines. If the product is delivered on time and passes verification of use cases and test scenarios, the product team ensures that requirements have been met. In that case, it becomes a ready-for-deployment artifact with a specified build date. The release plan is required to set a strategy for how a brand new product or product update will be introduced to the end-user. A soft launch strategy is the best option to pilot upcoming releases with early adopters or loyal audiences.

Go-to-Market. In the earliest stages of launching a product to the market, the launch team researches market conditions, and go-to-market strategy is planned for the new product or a service. After defining the market and target audience, the final value proposition is developed through multiple iterations of experiments and test campaigns. Then business defines the pricing strategy to fulfill company needs. Moreover, choosing the right distribution channels and distribution model followed by promotion is the most vital step before deploying a go-to-market strategy. When a competitive advantage is created, the marketing team decides how they will promote the final product and what kind of marketing campaigns are the most profitable for the go-to-live phase.

Growth. Scaling products to reach a bigger audience and increase profits is vital for any product that found its market fit and is ready for broader expansion. As the next step, value proposition and sale offers are continuously shaped with each sale by the reality of the market and feedback from customers. New functions such as customer support and customer marketing are introduced to execute marketing plans and monitor metrics of market adoption.

Optimization. As part of the natural product evolution, the maturity of the product grows with more time presented on the market. The product value proposition becomes stable, and only micro-optimizations are returned as updates of the well-developed product strategy. The product maintains saturation during the optimization phase, and scale speed goes down the same as profit growth. Most of the product enhancements may use data analysis and rely on big data to grow business metrics like ROI or LTV. Additional product features can be introduced as brand new products to support sales and extend the value proposition to the existing customer segments or new markets. Those pivots start the new iteration of the product lifecycle and can dramatically impact the future product strategy.

End-of-life. In case of a decision to prevent keeping the product alive, stop ongoing optimizations, and decline the existing product, it goes into the end-of-life phase. Decommissioning of the product contains the procedures of planning sunset or product replacement. Performing post-mortem is the final phase of the product before its possible future reincarnation.

3 TRIZ integration into product development process

Despite some less traditional applications of TRIZ for such spheres as business and management [19], as well as for the programming (i.e. algorithmic) [20] the idea of TRIZ application for digital product development for the whole software lifecycle is rather new [6]. In the present chapter the approach to utilize TRIZ for software development looking back to product Discovery and Delivery phases is described.

Thus, as a consequence of the rapid growth of the IT industry, digital products are the most flexible systems for any market changes. The unique feature of digital products from the other technical systems is the higher speed of evolution caused by the short development cycles. Agile methodologies aim to achieve more predictable results with a smaller period and resources [21]. Modern Agile principles applied for software development make it flexible for regular updates and robust for rapid market changes. Software products follow the general Trends of Technical Systems Evolution, such as trends of static, cinematic, and dynamic [22]. The complexity of systems grows with the product scale. TRIZ techniques can be applied at any time, starting with product discovery, innovation, and analysis and finishing with go-to-market, optimization, and end-of-life.

Strategy. The strategy phase contains a high degree of uncertainty, starting from an idea and finishing with a new invention, and may utilize multiple TRIZ tools for achieving more accurate results. Su-field Analysis is a powerful tool that can be applied for business modeling or product value proposition design [23, 24]. Su-field diagram from Fig. 1 describes a general model of interaction between User and Service as a contradicted pair and Item as a field that creates interest for User to make an action such as purchase on the Service. This general model can be applied to any e-commerce service such as Amazon, Netflix, Spotify, or niche products that propose a specific value or Service to the final customer. General Su-field is a powerful tool that product managers can also apply for Social Community design, including the niche Social Networks design [25]. Ableton's sound-producing community is considered as a case to express the value of the application of the Su-Field model in digital product development (Fig. 4). It brings local artists, skilled users, and newcomers at regular intervals to locations all over the world to create an additional boost of interest in their product. Ableton creates an environment for different user groups and maintains the organizational part and sponsorship. Still, experienced users help with the basics as a part of social networking and the desire for collaboration. Moreover, almost everybody can become a community facilitator and organize their local meet-ups to grow the community. Thus, being an outcome of the heuristic approach the same model was achievable with Su-Field analysis which is a more formal tool in such a case compared with more classical but less systematic Brainstorming, Design Thinking and other rather heuristic toolsets.

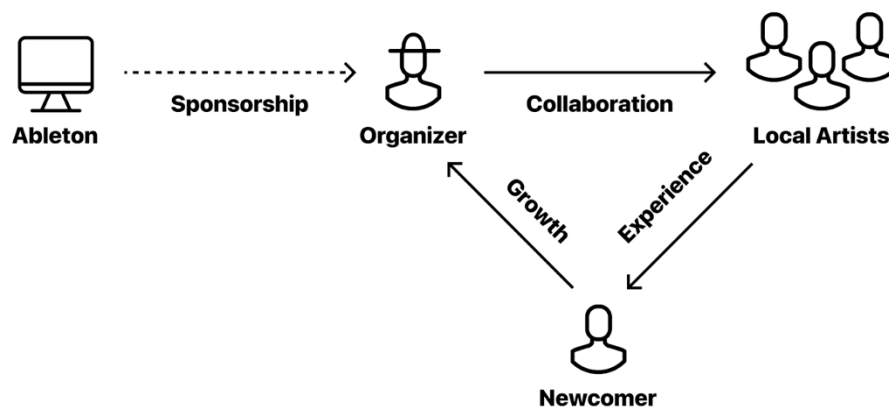


Fig. 4. The Ableton community described using the Su-Field Diagram as a contradiction between Organizers, Local Artists, and Newcomers.

Analysis. Creating and prioritizing hypotheses is a challenging problem. However, digital product development is more nowadays about generating and testing hypotheses rather than development of the determined solution after continuous market and product research. This is due to the fact that market conditions are now changing promptly, and implementing a hypothesis turns out to be economically more effective than prolonged market research and subsequent development. Ability to launch digital products on big markets with a relatively small team due to the present advanced technology stack made

the product development path to the short iterations starting with a set of hypotheses. And this is why it is important to find a more formal and systematic approach for hypotheses catching and formulation. Authors addressing TRIZ for the Analysis part of the product development as a toolset that have been developing for the analogous problem in classical engineering during the last half a century.

Thus, in case of wrong prioritization, the cost of failure depends on the cost of solution implementation and the amount of changes to fix the issue. That is why the integration of TRIZ tools into the Analysis phase can be considered in order to reduce risks of failure and help to increase the quality of decisions made.

One of the IFR applications for the product strategy and vision is the definition of the main function such as a user story or job story [26]. To describe the function in the user's terminology, the format of the job story proposes to describe the Situation, Motivation, and Expected Outcome. The ideal job story as a function definition is written based on the user's needs and doesn't specify any details about the technical solution.

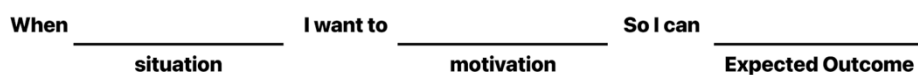


Fig. 5. Template for Job story mapping to define a user function.

IFR is a type of function that a product manager should define as the target solution and part of the product vision. After that, any iteration solution can be simplified using function-cost function and analysis. As an example, one solution can have a higher level of Ideality and higher cost, but another is lower Ideality and cost. Product managers should make such decisions and trade-offs according to available resources and constraints. The level of Ideality can be an additional parameter for function scoring and prioritization. It can be introduced into the RICE model formula that includes Reach, Impact, Confidence, and Efforts values as the main inputs for priority scoring [27]. Integration of the Ideality level to the RICE model gains an additional visionary insight for backlog prioritization which is highly important to startups with futuristic plans.

Development. The main challenge for the development phase is the design architecture that matches the product backlog requirements. Multiple architectural patterns such as microservice architecture or multi-agent systems are used as modern practices on how to build sustainable systems. The TRIZ concepts such as Subsystems or Supersystems, Monosystem, Bisystem, and Polysystem can be used as standard decision options for each technical solution. Engineers can integrate the TRIZ principles about system evolution into the system design phase to make a decision about component redesign and scalability.

Go-to-Market, Growth. All phases from the post-product launch phase have the same principles of product evolution. After introducing a product to the market, the next question to answer is what will be our next iteration after receiving real feedback from the fields. There are multiple TRIZ strategies that can be integrated into those phases.

One of them is mini and maxi problem-solving. Using the result of the go-to-market deployment and monitoring KPIs helps to make the right decision on how to pivot the product forward. If profits are running low, the radical option using maxi-problem formulation can be a complete redesign and can be selected as a future improvement strategy. Another option is to solve the mini-problem, keeping the product almost the same with small optimizations.

Optimization. The standard approach for product optimization is the mini-problem formulation and solution based on micro-optimizations. The radically opposite solution to this approach is the strategy of transitioning to Macro-level or Micro-level, which is the standard TRIZ practice. Transition to the Micro-level method provides a new inventive solution by replacing the operational principle behind subsystems. In other words, a solution on the micro-level doesn't change the function itself but changes the way how the technical solution works inside.

On the other hand, transitioning to a Macro-level method provides a solution to evolving a technical system to the next stage of evolution. The first solution is increasing the number of products or interactions between existing components. It is a valid approach for the products under the umbrella brand or the microservices. The second solution is to unite all smaller product lines into a single product. One of the well-known examples is the Wechat super app that aggregates the most common functions like messages, news, and payments into one product. And finally the third solution can be bringing additional features to the existing system that increases the properties of the product and creates additional value for the final customer.

End-of-Life. At first glance, the final phase of the digital product does not have any challenges and problems to solve because of its terminal specifics. That is why in this paper, we have no examples of TRIZ integration into the End-of-Life phase. We keep this section open as an opportunity for future TRIZ integration research papers.

4 Conclusion

Being developed from the early 50s of the last century TRIZ is now saturated with different case-based knowledge and theory in the field of construction and engineering for problem solving and design problems including those challenges arising in the context of the Industry 4.0. However, due to the growth of the digital products market, more inventive challenges in the design appear in the field of such areas as Internet-of-Things, Big Data, Cloud Computing, Blockchain software and user web- and mobile-interface applications. In such areas there is a whole set of inventive problems arise in the field of customer development, product discovery and product delivery, product market-fit finding and go-to-market strategy formulation. Despite its effectiveness in problem-solving in the field of classical engineering, TRIZ has not received much popularity in the circles of product managers, software developers and UX/UI designers of digital products. Although there is a number of studies and cases suggesting the TRIZ exploitation in business, management and digital products development, the theory of

TRIZ application in such areas is still on a very early stage of adoption. Thus, the approbation of such an effective tool for engineering in the area of digital product development paves the way for new innovative development approaches in a growing digital sphere opening up new challenges and opportunities. Hence, despite the efficiency of the TRIZ suggested tools there are still challenges to be overcome, such as the lack of a problem formalization model and heuristic nature of the applied steps determination compared to the more formal ARIZ approach [28]. That makes it challenging to link the inventive principles directly with model contradictions and leads to further research in the field of digital product development. One of the possible solutions for the relation identification of functional requirements with design parameters for the digital product design stage may be considered as Functional Decomposition and Morphology framework [29] and recommended to be addressed for future research to be adopted for software design and development.

Despite the mentioned challenges, the main research direction in this article of inventive toolset integration in digital makes it possible to apply proven innovation methodology to digital product delivery. The proposed toolset for the digital domain is supposed to facilitate the software development process from the early design stage to the scrum iterations for faster delivery of the key product features and faster hypotheses approbation. Being a continuation of the previous study [6], the present article uncovers the potential of TRIZ methodology at each stage of the digital product development throughout the digital product life cycle.

References

1. DeLong, J.B. and Magin, K., 2006. A short note on the size of the dot-com bubble. https://www.nber.org/system/files/working_papers/w12011/w12011.pdf
2. Nylén, D. and Holmström, J., 2015. Digital innovation strategy: A framework for diagnosing and improving digital product and service innovation. *Business Horizons*, 58(1), pp.57-67.
3. "The number of ICT specialists in the EU grew by 50.5 % from 2012 to 2021, almost 8 times as high as the increase (6.3 %) for total employment." Eurostat "ICT specialists workforce continued to grow in 2021" 05/05/2022. <https://europa.eu/!Jfdpy9>
4. Salamatov, Y. and Souchkov, V., 1999. TRIZ: the right solution at the right time: a guide to innovative problem solving (p. 256). Hattem: Insytec.
5. Zlotin, B., Zusman, A., Altshuller, G., & Philatov, V. (1999). *Tools of classical TRIZ*. Southfield, MI: Ideation International.
6. Kaliteevskii, V., Bryksin, M., & Chechurin, L. (2021, September). TRIZ Application for Digital Product Design and Management. In *International TRIZ Future Conference* (pp. 245-255). Springer, Cham.
7. Nählara, H., & Gronauerb, B. (2017). TRIZ in the Context of Digitalisation and Digital Transformation. In *Proceedings of the MATRIZ TRIZfest-2017 International Conference* (pp. 475-482).
8. Boikaa, S., Kuryana, A., & Ogievicha, D. (2017). Applications of TRIZ in business systems. *TRIZfest-2017*, 52.
9. Wu, Y., Zhou, F., & Kong, J. (2020). Innovative design approach for product design based on TRIZ, AD, fuzzy and Grey relational analysis. *Computers & Industrial Engineering*, 140, 106276.

10. Korolyov, V. A. (2013). Glossary of TRIZ-OTSM terms. Retrieved December 01, 2013, from <http://www.triz.org.ua/enc/>
11. Casner, D., Souili, A., Houssin, R., & Renaud, J. (2018, October). Agile' TRIZ framework: towards the integration of TRIZ within the agile innovation methodology. In *International TRIZ Future Conference* (pp. 84-93). Springer, Cham.
12. Blank, S., 2020. *The four steps to the epiphany: successful strategies for products that win*. John Wiley & Sons.
13. Patton, J., 2017. *Dual track development is not duel track. Dual Track Development Is NotDuelTrack*.
14. Greenway, A., Terrett, B. and Bracken, M., 2021. *Digital transformation at scale: Why the strategy is delivery. Do Sustainability*.
15. Sutherland, J. and Sutherland, J.V., 2014. *Scrum: the art of doing twice the work in half the time*. Currency.
16. Dennehy, D., Kasraian, L., O'Raghallaigh, P., & Conboy, K. (2016). *Product market fit frameworks for lean product development*.
17. Banfield, R., Lombardo, C. T., & Wax, T. (2015). *Design sprint: A practical guidebook for building great digital products*. " O'Reilly Media, Inc."
18. Gothelf, J. (2013). *Lean UX: Applying lean principles to improve user experience*. " O'Reilly Media, Inc."
19. Chechurin, L. and Borgianni, Y., 2016. Understanding TRIZ through the review of top cited publications. *Computers in Industry*, 82, pp.119-134.
20. Chechurin, L., 2016. *Research and Practice on the Theory of Inventive Problem Solving (TRIZ)*. Switzerland: Springer.
21. McCormick, M. (2012). *Waterfall vs. Agile methodology*. MPCS, N/A.
22. Cascini, G., Rotini, F., & Russo, D. (2011). Networks of trends: systematic definition of evolutionary scenarios. *Procedia Engineering*, 9, 355-367.
23. Terninko, J. (2000). Su-field analysis. *TRIZ Journal*, 2, 23-29.
24. Osterwalder, A., Pigneur, Y., Bernarda, G., & Smith, A. (2014). *Value proposition design: How to create products and services customers want*. John Wiley & Sons.
25. Giuffre, K. (2013). *Communities and networks: using social network analysis to rethink urban and community studies*. John Wiley & Sons.
26. Patton, J., & Economy, P. (2014). *User story mapping: discover the whole story, build the right product*. " O'Reilly Media, Inc."
27. Trieflinger, S., Münch, J., Bogazköy, E., Eißler, P., Schneider, J., & Roling, B. (2021, June). How to prioritize your product roadmap when everything feels important: a grey literature review. In *2021 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)* (pp. 1-9). IEEE.
28. Borgianni, Y., Fiorineschi, L., Frillici, F.S. and Rotini, F., 2021. The process for individuating TRIZ Inventive Principles: deterministic, stochastic or domain-oriented?. *Design Science*, 7.
29. Fiorineschi, L., Frillici, F.S. and Rotini, F., 2018. Enhancing functional decomposition and morphology with TRIZ: Literature review. *Computers in Industry*, 94, pp.1-15.