
Using Prototypes in Agile Software Development

Ali Hamzah Obaid

Al-Furat Al-Awsat Technical University Babylon Technical Institute Babylon, Iraq
inb.ali210@atu.edu.iq

Abstract

Prototyping has now become an important and essential part of software development processes. As it is considered one of the ideal ways to receive new jobs through changing requirements during the development process due to its speed and efficiency in delivering models that simulate the final model and its ability to be developed and changed according to the opinions of stakeholders.

Agile development and its iterative advantage and adaptability to changing requirements from stakeholders allows business teams to clearly and accurately understand requirements.

This paper explores that the use of prototyping in agile software development has an essential role in early and continuous verification of requirements in the development process through the creation of quick, concrete models of proposed features and functionality. This helps in identifying errors, problems or misunderstandings early in the process. Development and this understanding helps save time and material costs allocated to the development process. Using prototypes and integrating them into agile development helps work teams and stakeholders adapt flexibly, obtain a clear vision of requirements, and identify weak points in the development process by holding meetings and Periodic meetings to obtain a highly efficient product that meets their desires and the desires of end users.

Keywords: Prototype, Agile, Software Development, Methodology, Development Projects.

The Aims

The current study aims to clarify the importance of using prototypes in agile software development through exploratory knowledge.

1. Introduction

Early in the development process, when the solution to the problem is uncertain, software development processes using Agile methodologies are characterized by creativity and design thinking by focusing well on the end-user's or stakeholders' needs to reach problem solutions faster. To create interactive models that simulate the final product, a vision of the customer's requirements can be accessed and evaluated [1].

Agile software development methods that use prototyping facilitate the creation and improvement of early iterations of a product because prototyping allows rapid iteration in partnership with stakeholders and users [2].

Prototypes help in clarifying the final product and verifying requirements by choosing a design method and identifying gaps and problems in real time early in the development process. In the Agile software development process in general [3], Prototypes promote a collaborative and iterative approach to obtaining early feedback and the ability to adapt to changing requirements. Therefore, using Prototypes leads to reduced development time through software solutions that meet the needs of end users from The early stages of development [4]. Prototypes also allow the development team to create multiple versions of the product and improve it in cooperation with users and stakeholders to reach the final version of the product by understanding the requirements and making appropriate decisions for the final design after identifying problems and improving them early in the development process [5].

The use of Prototypes methodologies in agile development enables adaptation to changing requirements by stakeholders by promoting a collaborative, iterative approach, which in turn ensures that the final software solutions meet the expectations of end users from the early stages of design, and significantly improve the efficiency of the development process to reach the best costs, less time in the development process, and reduced software corrections and documentation [6].

Using Prototypes in Agile software development promotes a simpler and more efficient development process, reducing the need for extensive changes and revisions later on. Overall, using Prototypes in Agile software development helps reduce requirements fluctuations and reduces the need for extensive changes and revisions during the development process, resulting in a more successful and satisfying final product [7].

1.1 Background

1.1.1 Agile Software Development

In recent years, many different methodologies and approaches to software development have been introduced. Few methodologies are used today for many reasons. (Nandhakumar and Avison,) [8] point out that traditional software development methodologies are treated as an imaginative innovative design to provide an image of control or to provide a symbolic status. While (Parnas and Clements, [9], Wiegers, 1998, Truex et al, [10]) take an extreme position on traditional methods. Through this simple background, the emergence of agile software development methods are allowed.

Agile software development has its origins specifically in the Agile Manifesto written by (Beck et al., 2001:6), which shows that it is a user-centered approach, and the customer Who participates in development activities during the development process is the focus, which also allows the product to be adapted according to needs,

including requirements and discovery, solution improvement, self-organization, collaborative effort, cross-functional teams, cooperation with customers and end users, and division of tasks [11].

Agile software is a methodology that has gained wide popularity in the software development process due to its ability to facilitate continuous collaboration and effective communication between development teams. The core principles of Agile, flexibility to accept changing requirements, and strong collaboration between developers, stakeholders, and end users, make it well-suited for projects that require adaptability and continuous improvement. Additionally, Agile methodologies align with human-centered approaches to creating the right design, which emphasizes the importance of understanding end users and their technical contexts, and ensuring that the programs that are developed and built are compatible with their views and requirements. Agile argues that software specifications should emerge appropriately and naturally through the development process, delivering prototypes continuously in short periods of time throughout the development process. This allows for improving previous Prototypes and incorporating good practices and patterns to build efficient and high-quality software products. Also, Agile methodologies emphasize the importance of face-to-face interactions between the development team and stakeholders during the development process. This encourages regular communication with clients to adjust priorities and changing requirements to the appropriate time, scope, and project plan. It also enhances teamwork between different development teams, and supports effective collaboration and communication with work teams across different regions and different geographical locations [12].

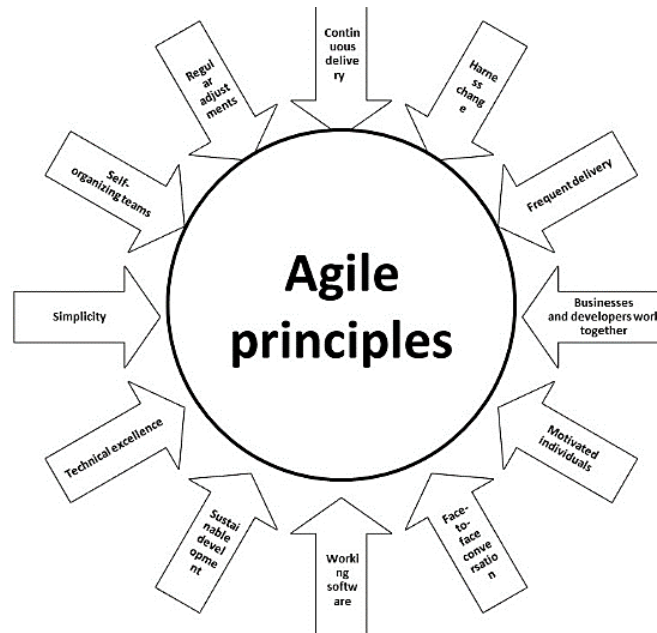


Figure (1): The 12 Agile Principles of the Agile Manifesto

1.1.2 Prototype Methodology

Prototypes methodology is an iterative approach to software development by creating a working copy to develop the proposed or required system. Different methods can be used in the development process. The prototype is developed according to methodologies based on the basic assumptions, concepts, and processes in the product life cycle. The development process is carried out according to techniques that are compatible with the knowledge available to the organization at that time. Prototypes methodologies allow for early feedback during the development process and validation of the proposed system's functionality and usability during the design period [13].

The prototype methodology is an approach to software development before the full version of the system is developed. It focuses on creating multiple mini-versions to demonstrate functionality and receive feedback from stakeholders. It is frequently

applied in circumstances where requirements and needs are not defined or are likely to change over a period of time of system development.

In Prototypes methodologies, during the planning and analysis stages, multiple tools are used, such as taking the opinions of experts and users and looking at other similar projects, in order to make estimates of the material costs and time costs of the effort required in the next stages of the development life cycle [14]. These estimates are usually made in detail. Then, the first model methodology moves to the design and development stages after taking into consideration the organization's requirements to create a working model for the required system based on the organization's initial requirements and the specifications required for the design. In this stage, the development and design team works closely to produce a prototype that simulates the requirements, features of the product, and its basic functions to reach the final product. As the development process progresses, new features and improvements are added based on the desire of end users or stakeholders to produce a more realistic version of the final product. Collaboration and exchange of ideas during the development process is a characteristic of prototypes due to their iterative nature in creating multiple versions of the product, and this is what encourages stakeholders to communicate with work teams to determine the functionality and needs of the end user at each stage of delivery [14]. The final model undergoes a series of tests and evaluations in order to ensure that the product is free of any errors or problems in design or programming, which can be fixed early before the product is delivered [16].

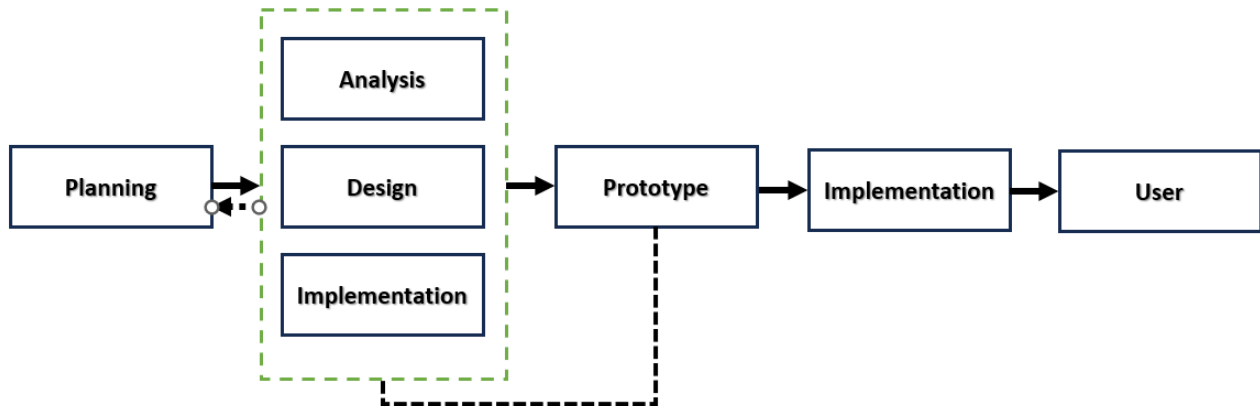


Figure (2): Prototype Methodology

2. Integrating Prototyping Methods into Agile Software Development

Lack of understanding of requirements is one of the most important problems facing software development teams, which may cause the delivery of a product that is not good and does not meet the ambitions of stakeholders. Given the advantages of prototypes in continuous and effective communication with stakeholders, this communication provides a better understanding of requirements and access to designing a product that satisfies all parties. Early feedback allows for the design of a product that meets the expectations of stakeholders [17].

Incorporating prototypes into agile development enhances the transparency of requirements through the exchange of ideas between development teams and end users. This collaboration enhances the early identification of additional functionality and avoids problems that may occur during the product delivery process [18].

Continuous improvement allows the production of a highly efficient product through early integration in understanding requirements based on consensus between work teams and end users through effective cooperation and understanding.

In order to achieve the best results in the final product, the process of iteration and experimentation is carried out between the developer and the user through continuous evaluation by collecting comments and suggestions during the development period from the client. Through Continuous evaluation: The product is continuously improved to achieve the best results through the feedback provided by users. The use of the prototype methodology is one of the common methods in agile development due to the flexible and iterative approach to software development that aims to deliver a highly efficient product in short increments. The agile development methodology focuses on collaboration, adaptability, and user satisfaction [19].

When integrated into software development, Agile is the process of quickly building a basic version of a product to gather feedback, test product functionality, and define requirements. This approach allows for rapid iteration and improvement, creating a product that meets the customer's requirements, needs, and expectations as shown in Figure (3).

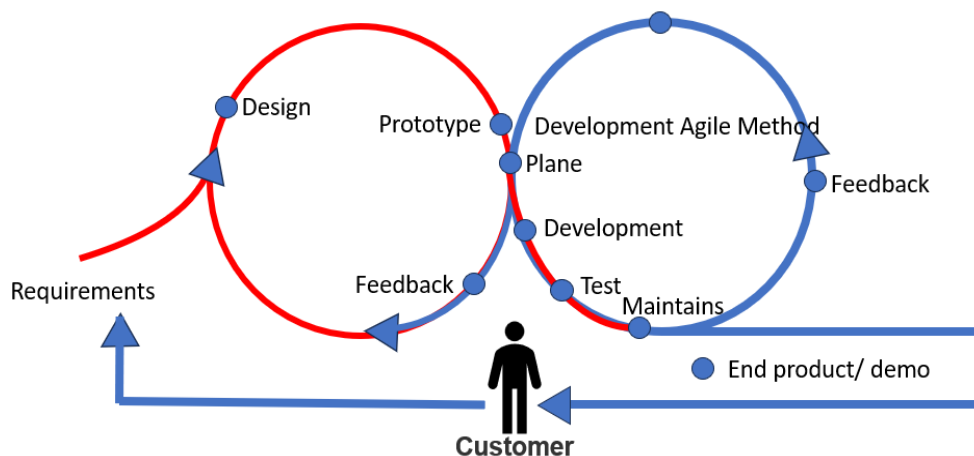


Fig. (3): Merging Prototyping in Agile Software Development.

Merging Prototypes with Agile helps to reuse the development components and reduce the restrictions imposed on the development process, Through the process of documenting the development components of Agile [20].

3. Steps to Develop Agile Software Using Prototypes

There are many important steps in creating Prototypes in Agile software development after collecting requirements from the customer. The development team identifies and distinguishes these requirements, collects functions, and includes them in the prototype. After that, the design team creates a rough design for the user interface, specifying the basic design of the product and its most important elements through requirements-based planning using prototype tools. This model focuses on basic functions and is called a low-level prototype [21]. It helps validate stakeholders' requirements at an early stage. Once the comments are collected in the low-level model, a higher-accuracy model is built with more functions, elements, and interactive features.

The prototype allows to deal efficiently with unstable conditions, especially with fluctuating requirements to achieve early course correction (Menéndez and Silva, 2016). Building the prototype facilitates the users and stakeholders to fully visualize the program through the concrete representation of requirements and the possibility of dealing with comments. The necessary modifications are made by the development team. Due to the ability of Prototypes to deal with the requirements accepted by stakeholders, it is an essential technique in agile development with recommended categories including business, ease of use, performance, and capacity of the prototype (Münch et al., 2013).

In general, the use of Prototypes in rapid development is extremely important to effectively manage requirements and volatile and unstable conditions.

The iterative steps in Prototypes used in Agile development ensure that the software is continuously improved based on end-user feedback. Figure (4) includes the steps on how to use Prototypes in the Agile software development process.

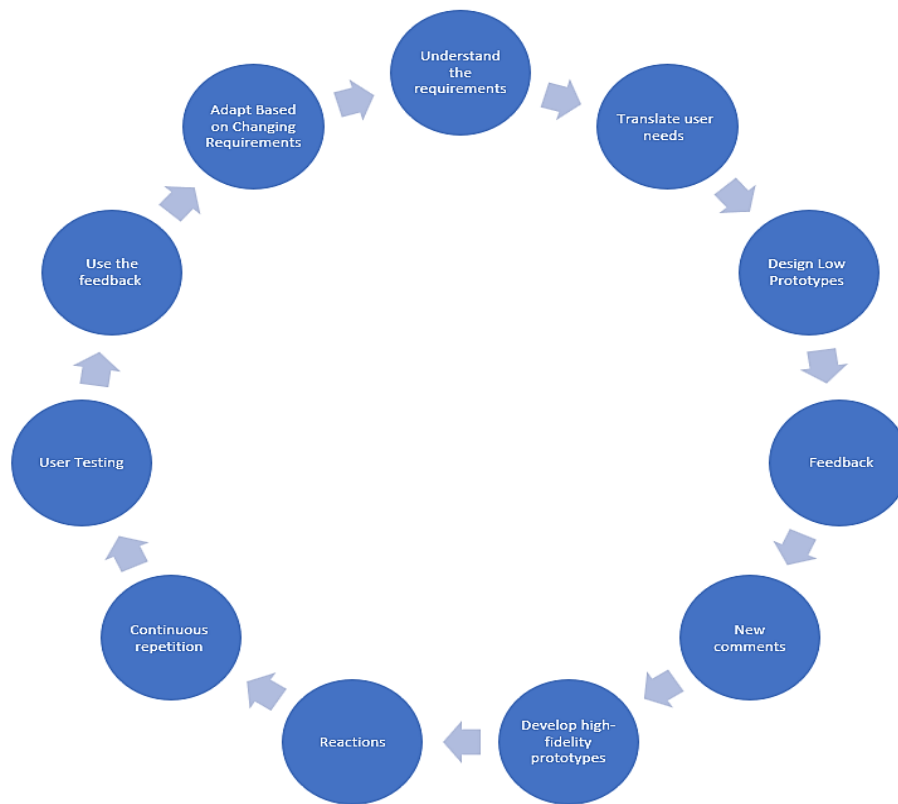


Figure (4): Steps to develop agile software using prototypes

4. User Feedback on Prototype Methodology and its Integration into Agile Development

User feedback is an essential element in the development process. It helps uncover and identify product problems, allowing for improvement and providing a good vision for the final product. This feedback provided by prototypes can be

incorporated into Agile software development and collected through early prototypes allows teams to Making adjustments to meet their needs. To ensure that the product meets the ambitions of stakeholders. Prototypes allow crucial decisions to be made through iterative observations and feedback. The iterative feedback loop helps ensure that the product meets the customer's aspirations. The participation of end users in expressing their opinions through participation in the development process contributes to making decisions in the final form of product design, defining the necessary functions, and directing work teams to deliver a product that is more appropriate to the requirements [22].

User comments and their identification of software defects are of great benefit in feedback for product development and allow development teams to identify defects and errors in product functionality, which contributes to resolving problems early in the delivery process. An Agile software development timeline with prototyping To provide high-quality products, it is of utmost necessity to deliver the product on time and meet customer expectations [23,24].

5. Prototyping Strategy in Agile Development

Prototyping allows for continuous feedback, meeting requirements, and collaborating with customers throughout the development process, ensuring alignment with customer needs and requirements (Kaur et al., 2015). In addition, agile methodologies allow Adaptability and flexibility by making adjustments and responding quickly to changing requirements.

By following an agile software development timeline approach using prototyping, development teams can iterate on the basic designs and functionality of a product, incorporate end-user and customer feedback, and incrementally improve product quality (Williams, 2010). Figure (4) represents three sprint phases, each of which represents a time-bound iteration in agile development using prototyping. The first

phase of the sprint focuses on low-cost (Low-Fi) and high-fidelity (High-Fi) prototyping activities. The second sprint phase focuses on developing the first sprint of prototypes, while the third sprint phase focuses on improvement processes and integrating requirements and comments to improve the product. The arrows indicate the flow of work in progress in the prototypes to the development and improvement process, which illustrates the iterative and incremental process of agile development of continuous and organized feedback loops that each cycle contributes to agile development. Prototyping fits into the timeline of Agile development. The scope of development projects is prioritized based on the tasks, their importance, and the actual need for them. The requirements for the first sprint are directly linked to the requirements desired by the client based on end-user feedback. In this sprint, an initial version of the product is created using prototypes, building, iterations, debugging, improving, innovating, and writing code. Iteration in production ensures that the prototype is improved and all requirements are validated. Once the prototypes are improved and requirements are validated, the process of Production and release of the program begins and the product enters the maintenance phase, and any changes or other necessary requirements that are determined by the end users are made. This is what characterizes the second and third sprint phases.

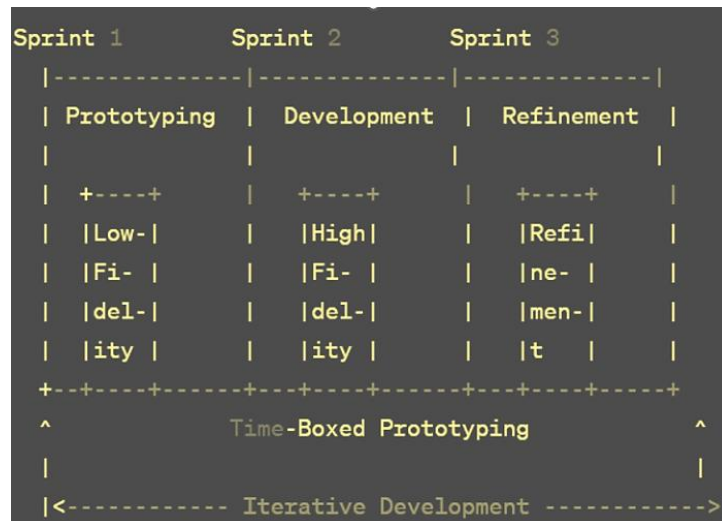


Figure (5): Agile software development timeline with prototyping

Conclusion

The use of prototyping in Agile development is to learn and improve the product. It allows us to effectively test ideas, make necessary adjustments, and ensure that the final product meets the user's needs.

Using the prototyping methodology in Agile software development contributes to achieving many advantages, the most important of which is its contribution to the success of the project and its effectiveness in meeting requirements to reach the best quality through continuous and early feedback. Prototyping allows early feedback to be collected from customers and end users to improve the compatibility of the final product. The agile approach emphasizes continuous collaboration between work teams and stakeholders and involving them in the development process and evaluation of the final product, which helps in communication and understanding, which is what prototypes work on with visual and tangible representations of the product, which in turn help end users visualize the design and functionality of the

software, which contributes to reducing the costs allocated to the development process.

Reference

- 1- Davison, R. M. (2020). Research contributions: The role of the iconoclast. *Information Systems Journal*, 30 (2), 215-219.
- 2- Martin, R. C. (2003). *Agile software development: principles, patterns, and practices*. Prentice Hall PTR.
- 3- Obaid, A. H. (2017). Tools for conceptual-algorithmic prototyping in solving design problems in the development of systems with software. In *INTERACTIVE SYSTEMS: Problems of Human-Computer Interaction* (pp. 276-282).
- 4- Münch, J., Fagerholm, F., Johnson, P., Pirttilahti, J., Torkkel, J., & Järvinen, J. (2013). Creating minimum viable products in industry-academia collaborations. In *Lean Enterprise Software and Systems: 4th International Conference, LESS 2013, Galway, Ireland, December 1-4, 2013, Proceedings* (pp. 137-151). Springer Berlin Heidelberg.
- 5- Obayes, K. A., & Hamzah, A. (2022). Using of prototyping in develop an employee information management. *Measurement: Sensors*, 24, 100557.
- 6- Sarma, A. D. (2019). One-pot synthesis of requirements elicitation for operational bi (obi) system: in the context of the modern business environment. *Journal of Information Systems Engineering and Business Intelligence*, 5 (2), 131. <https://doi.org/10.20473/jisebi.5.2.131-145>.
- 7- Menéndez, D. A., & da Silva, P. C. (2016). A requirement elicitation process for BI projects. *Lecture Notes on Software Engineering*, 4 (1), 20.
- 8- Nandhakumar, J., & Avison, D. E. (1999). The fiction of methodological development: a field study of information systems development. *Information technology & people*, 12 (2), 176-191.
- 9- Parnas, D. L., & Clements, P. C. (1986). Correction to “a rational design process: How and why to fake it”. *IEEE Transactions on Software Engineering*, (8), 874-874.
- 10- Knosalla, K. (2009). Is the world agile? A review of the IT knowledge base and debate on agile methodologies in the field of information systems development. *From the Head of Group*, 20.
- 11- Al-Husseini, K. A. O., & Obaid, A. H. (2020). Interaction between project tasks and risk management tasks in software development. *Periodicals of Engineering and Natural Sciences*, 8 (4), 2300-2308.

-
- 12- Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software development*, 9 (8), 28-35, ISO 690
 - 13- Jiménez, M., Piattini, M., & Vizcaíno, A. (2009). Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering*, 2009.
 - 14- Al-Saqqa, S., Sawalha, S., & Abdelnabi, H. (2020). Agile Software Development: Methodologies and Trends. *Int. J. Interact. Mob. Technol.*, 14, 246-270.
 - 15- Comella-Dorda, S., Lohiya, S., & Speksnijder, G. (2016). An operating model for company-wide agile development. *McKinsey Quarterly*, May, 1-10.
 - 16- Dennis, A., Tegarden, D. P., & Wixom, B. H. (2003). *Systems analysis and design: an object-oriented approach with UML*. Wiley.
 - 17- Chamberlain, S., Sharp, H., & Maiden, N. (2006, June). Towards a framework for integrating agile development and user-centred design. In *International Conference on Extreme Programming and Agile Processes in Software Engineering* (pp. 143-153). Berlin, Heidelberg: Springer Berlin Heidelberg.
 - 18- Higuchi, M. M., & Nakano, D. N. (2017). Agile design: A combined model based on design thinking and agile methodologies for digital games projects. *Revista de Gestão e Projetos*, 8(2), 109-126.
 - 19- Schuh, G., Doelle, C., & Schloesser, S. (2018). Agile Prototyping for technical systems—Towards an adaption of the Minimum Viable Product principle. *DS 91: Proceedings of NordDesign 2018*, Linköping, Sweden, 14th-17th August 2018.
 - 20- Tanvir, S., Safdar, M., Tufail, H., & Qamar, U. (2017). Merging prototyping with agile software development methodology. *Circulation in Computer Science. Int Confr Eng Comput Info Technol*, 50-54.
 - 21- Huber, T. L., Winkler, M. A., Dibbern, J., & Brown, C. V. (2020). The use of prototypes to bridge knowledge boundaries in agile software development. *Information systems journal*, 30(2), 270-294.
 - 22- Dönmez, D., & Grote, G. (2018). Two sides of the same coin—how agile software development teams approach uncertainty as threats and opportunities. *Information and Software Technology*, 93, 94-111.
 - 23- Leite, M., & Braz, V. (2016). Agile manufacturing practices for new product development: industrial case studies. *Journal of Manufacturing Technology Management*, 27(4), 560-576.

- 24- Obaid, A. H. (2017). Tools for conceptual-algorithmic prototyping in solving design problems in the development of systems with software. In INTERACTIVE SYSTEMS: Problems of Human-Computer Interaction (pp. 276-282).