ANÁLISIS DE METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE CENTRADAS EN EL USUARIO: UNA GUÍA RÁPIDA PARA SU ADOPCIÓN

ANALYSIS OF USER-CENTERED AGILE SOFTWARE DEVELOPMENT METHODOLOGIES: A QUICK GUIDE TO ITS ADOPTION

Erick Franco Gaona
Universidad de Guanajuato, México
e.francogaona@ugto.mx

Marco Bianchetti
Universidad de Guanajuato, México
mb@ugto.mx

María Susana Ávila García
Universidad de Guanajuato, México
susana.avila@ugto.mx

Recepción: 12/mayo/2021 **Aceptación:** 17/noviembre/2021

Resumen

Las metodologías de desarrollo de software se han adaptado al avance de la tecnología y a las necesidades de los usuarios. Además, el desarrollo de productos de software diseñados con un enfoque centrado en el usuario se ha considerado una prioridad para garantizar su éxito y adopción. En este trabajo se realiza una revisión bibliográfica del estado del arte para identificar las metodologías que combinan principios de desarrollo ágil de software con métodos de experiencia de usuario. Se presenta un análisis de cómo se han adaptado y adoptado los principios de diseño ágil y centrado en el usuario en el desarrollo de software, así como las experiencias y resultados reportados que destacan las ventajas y desventajas de su implementación. La finalidad es brindar una guía para la elección de la metodología dependiendo de aspectos como las características del equipo de desarrollo y el área de aplicación.

Palabras Clave: Adopción de metodologías, Desarrollo ágil, Diseño centrado a usuarios, Ingeniería de Software, Metodologías ágiles.

Abstract

Software development methodologies have been adapted to the advancement of technology as well as to user needs. Furthermore, the development of user-centered designed products has been considered a priority to ensure their success and adoption. In this work, state-of-the-art literature review is carried out to identify the methodologies that combine agile software development principles with user experience methods. Then, an analysis of how agile and user-centered design principles have been adapted and adopted in software development reported experiences, and results, highlighting their implementation's advantages and disadvantages. The aim is to provide a guide to select a methodology based on aspects, such as the characteristics of the development team and the application area.

Keywords**:** *Adoption of methodologies, Agile development, Agile methodologies, Software Engineering, User-centered design.*

1. Introduction

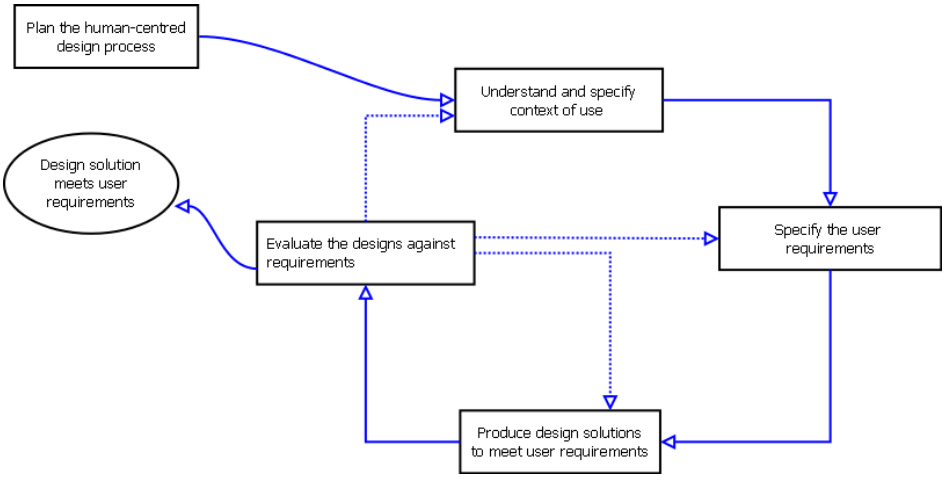
Agile development methodologies and user-centered design have been implemented to manage software projects and improve the quality of the software products [Silva da Silva et al., 2012]. Agile methodologies aim to improve teamwork organization, continuous delivery, and development time that traditional software development methodologies provide [Shankarmani et al., 2012]. The increase in the use of agile methodologies can be confirmed by the annual report that is carried out with around 40,000 participants, who are executives, professionals, and expert consultants in software development. In the 14th edition, it is mentioned that 95% of the companies use agile methodologies for their software developments [State of Agile, 2020]. However, it is a maturing process for the development teams that companies lead and requires time to implement because, as the same survey indicates, not all company teams are agile.

The principles of these methodologies, proposed in 2001 by a group of experts headed by Kent Beck, also creator of the agile Extreme Programming methodology [Beck et al., 2001], apply to any agile methodology:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The standard ISO IEC 9241-210 aims to provide a framework to make the systems usable and useful, and in this standard are also users and their needs driving the development [ISO ,2010]. The document describes the human-centered interactive systems design process as illustrated in figure 1. The methodology considers the iteration of the following activities:

- Plan the human-centered design process, where the stakeholders are described, and their characteristics, the goals of the users and the environment of the system.
- Understand and specify the context of use, where the conditions to apply the system requirements are specified.



Source: [ISO, 2010: 11]

Figure 1 Human-Centered Design Methodology Diagram.

- Specifying the user needs, where the system requirements are obtained from stakeholders.
- Produce design solutions to meet user requirements, where prototypes are created to get feedback.

- Evaluate the designs against requirements, until the solution meets the user requirements.

The principles of human-centered design also provided by the same ISO are [ISO,2010]:

- "The design is based upon an explicit understanding of users, tasks, and environments". Due to this, the entire team must know the needs of the stakeholders and their work environment.
- "Users are involved in the design and the development". This principle also helps agile development because involving the stakeholder in all development prevents products that are not useful from being designed, and less time is wasted.
- "The design is driven and refined by user-centered evaluation". Feedback can be obtained involving stakeholders each time a new product or product feature is available through evaluations that can be presentations on the use of the system.
- "The process is iterative". The process is carried out multiple times to improve the user experience so that the user's feedback helps to improve the product.
- "The design addresses the whole user experience". By improving the system with each iteration, the design improves so that the user experience is optimal. In this way the end-users will have an intuitive, efficient, and effective application.
- "The design team includes multidisciplinary skills and perspectives". The team must be multidisciplinary to divide the tasks and that each member contributes his knowledge and contributes to improving the performance of the others.

Despite the benefits offered by agile and human-centered design methodologies, there are proposals for combining them. In the first instance, agile principles do not allow for the implementation of user-centered design [Rojas & Macías, 2015].

Schwartz [2013] presents an analysis and comparison of this type of methodologies, including:

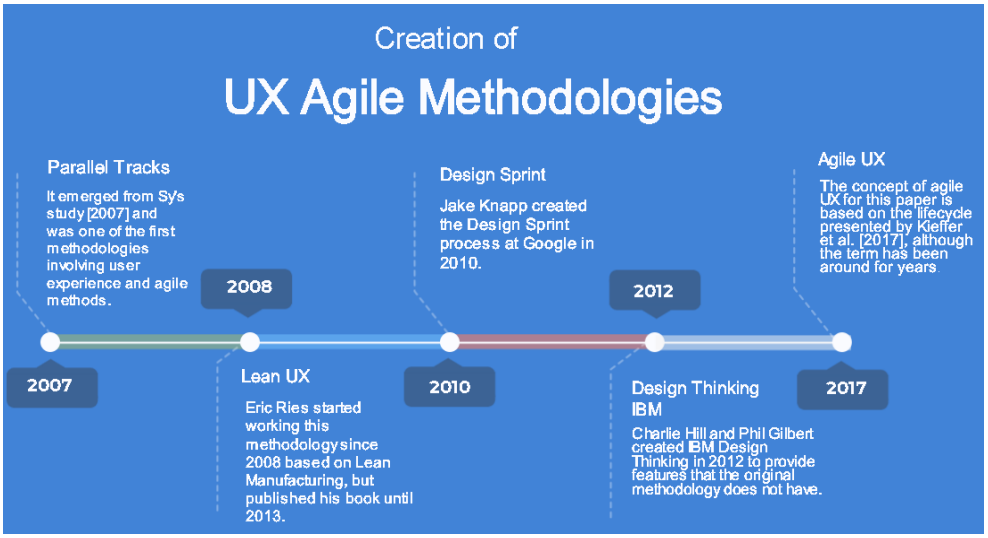
- Parallel Tracks.
- Parallel Levels.
- Sequence of an iterative design phase and an iterative development phase.
- Big upfront design.
- Usage centered design. The results, presented in a comparative table, show how most of these methodologies do not comply with various aspects of human-centered design or agile methods except for Parallel Tracks.

In this work, the methodologies that will be analyzed and that combine UX and Agile principles are:

- Parallel Tracks.
- Lean UX (User Experience).
- Design Sprint.
- Design Thinking.
- Agile UX (User Experience).

For this work, we are based on Schwartz [2013], but we chose the most current methodologies and showed key aspects that can help to choose between one of them depending on factors such as the size of the team. Some methodologies reported in Schwartz's study have not been widely reported in the literature, so we consider for our Parallel Tracks analysis. Methodologies such as Lean UX, Agile UX, Design Sprint, and Design Thinking were chosen for their use reported in the literature as shown in the analysis below, their extensive documentation, current validity, and support in books or guides. A timeline showing the emergence of each of these methodologies is shown in figure 2.

This work aims is to analyze and evaluate methodologies that integrate user-centered design practices and agile development methods, review their integration in some reported works, and provide a quick guide to assist in the choice of one of these methodologies in a software development project.



Source: [Own elaboration]

Figure 2 UX Agile Methodologies Timeline.

2. Methods

Analysis of agile UX methodologies

In this section, the selected methodologies are introduced, highlighting their core elements and principles, then the way these have been adopted is described.

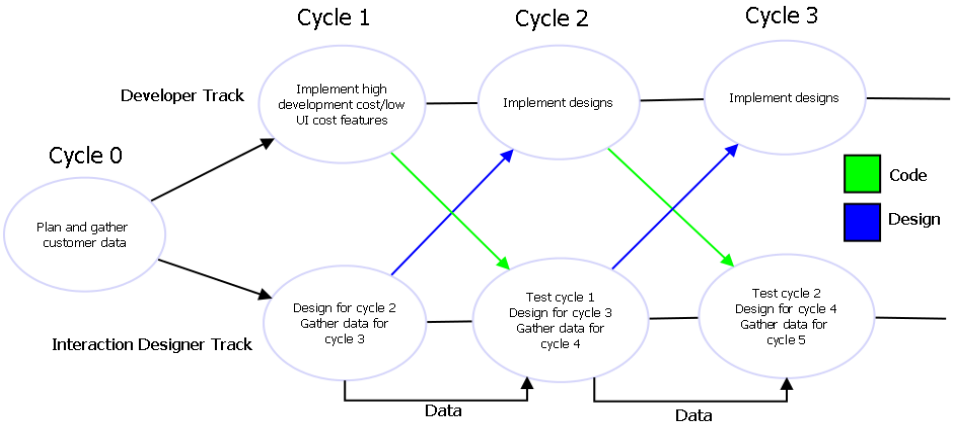
Parallel Tracks

Parallel Tracks allows to manage and organize work between developers and designers. Sy [2007] proposes to work on parallel tracks after the initial planning iteration, also called zero iteration, where user information is collected, and a project plan is made. The methodology allows usability experts to get ahead of developers, collect user data, analyze it, and propose design solutions. Given a design solution, designers and developers work one or two iterations time shifts.

The figure 3 shows the Parallel Tracks development cycle where it can be deduced that during iteration *i* the designers gather information from users and context for iteration *i* + 2. Work on the designs for iteration *i* + 1 to assist developers with the implementation of the designs for iteration *i*. Finally, they evaluate the software developed during iteration *i*-1.

The methodology has been studied and used in some projects. For instance, Silva da Silva et al. [2012] aimed to analyze the use of the proposed framework and the

integration of UX and Agile methodologies is a real project in a medium-sized company. Some of the issues reported were that UX designers could not collaborate closely with developers and work in advance because they were working on multiple projects simultaneously.



Source: [Sy, 2007: 118]

Figure 3 Parallel Track methodology workflow.

Fox et al. [2008] focused on testing the integration of agile development with UX while developing low fidelity prototypes. The authors reported that the integration of these methodologies fit well with the entire teamwork. Furthermore, the dependence that the design team has on the development team and vice versa was observed.

Lean UX

Gothelf & Seiden [2016] proposed the Lean UX methodology to eliminate waste from the design process. First, a Lean UX process is carried out to create only the product designs needed for the team's learning. Then, agile principles are applied to drive cross-functional collaboration that includes non-designers into the design process and designers, developers, and others. Finally, a change in mentality should be observed in team members when Lean UX principles and a model based on experimentation to evaluate ideas is implemented. The principles of this methodology are defined in table 1.

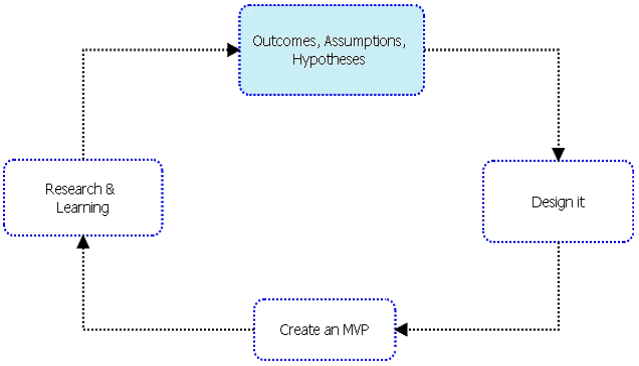
As shown in figure 4, this methodology creates assumptions and hypotheses to choose the objective that best contributes to the least possible risk. Once a

hypothesis is selected, a design meeting is held with the stakeholders to reach an agreement on the minimum viable product that will be designed and developed. Finally, a deliverable is obtained, and the necessary tests are carried out to learn what has been done well and what should change.

Table 1 Principles of Lean UX.

Team organization	Guide culture	Guide the process
• Multifunctional teams • Small, dedicated, located • Self-sufficient and empowered • Team focused on problems	• Moving from doubt to certainty • Results, not exit • Eliminating waste • Shared understanding • No know-it-all • Permission to fail	• Work in small batches to mitigate risk • Continuous discovery • GOOB: the new focus on the user • Externalizing work • Analysis renewal • Get out of the deliverables business

Source: [Gothelf & Seiden, 2016]



Source: [Gothelf & Seiden, 2016: 22]

Figure 4 Lean UX Process.

In industry, this methodology is being implemented progressively to familiarize project teams with it. Liikkanen et al. [2014] reported a case study conducted in a Finnish company where this methodology was introduced. A change of mentality was needed on the team, since the code was usually developed separately from the design, and on the clients who were already used to participating in design meetings at the end of the process. González et al. [2015] used a methodological approach to investigate interaction design and the usability of an interactive interface to solve mathematical operations. The authors used some ideas from User-Centered Design (UCD), User Experience Design (UXD), and agile methodologies. These methodologies led to correct errors that affected flexibility and efficiency, and met user's expectations.

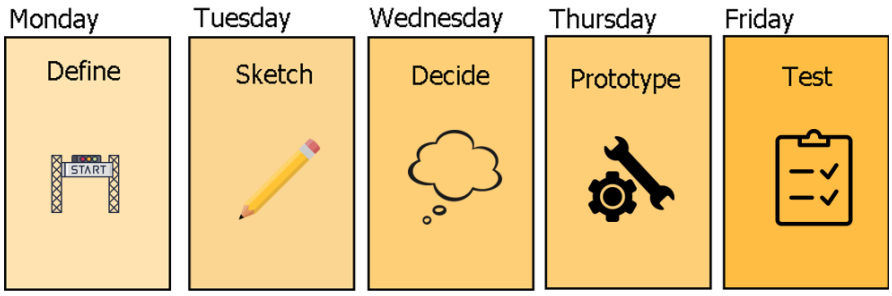
Design Sprint

Google proposed the Design Sprint methodology for teams of any size to solve design problems in 5 days and is mainly recommended for start-up companies [Ries, 2011]. In addition, it allows to quickly prototype and validate ideas with end-users. The idea of Sprints originates from agile concepts like in SCRUM. Another analogy with SCRUM is the presence of a Sprint Master team leader who has tasks very similar to a SCRUM Master. This person identifies the design challenge for the sprint, brings the team together, and takes it through all the Sprint stages [Knapp et al., 2016]. Design Sprint is based on the Lean Startup principles described by Ries [2011] as follows:

- Entrepreneurs can be anywhere. Anyone who is trying to innovate can help the team come up with good ideas.
- Entrepreneurship is organization. A start-up is not just a product but rather an institution, so it requires a new type of management for its context.
- Customer validated learning. Clients should always be close to development to contribute their ideas and feedback.
- Build - Measure - Learn. We should not wait to get the perfect product; it just needs to be good enough.
- Create innovation. When we create a solution that adds more value than existing solutions, we are already able to innovate.

In figure 5, the Design Sprint methodology is based on five days of work to provide design solutions and deliver a good value product at the end. It can be implemented as many times as necessary. On the first day the goals and the plan to achieve those goals are defined. On the second day, a sketch is defined by the work team to reach a common design through storyboards. On the third day, a decision is made based on the results of the previous two days. It is decided whether in this sprint the entire solution or only a part of it will be validated. On the fourth day, low fidelity prototypes should be sketched, focusing on validating the idea. The prototype is tested with real users to refine and correct it, deciding whether to start the development. This methodology has been used in recent years for different types of developments.

[Martinez et al., 2017] applied this methodology combined with mixed methods and usability tests to evaluate a diabetes control panel in a patient portal. The methodology was repeated until most of the participants completed the assigned tasks during the tests, users expressed high levels of satisfaction, and did not identify new usability problems.



Source: [Knapp et al., 2016: 10]

Figure 5 Design Sprint methodology workflow.

[Fay et al., 2019] described the work conducted in a three-day workshop that was held to design new user interfaces for future subsea control rooms to process higher volumes of data collected from next-generation sensors. The Design Sprint methodology was used to develop the UIs, to leverage the experience and recommendations of the expert participants to inform future interface design requirements and novel concepts.

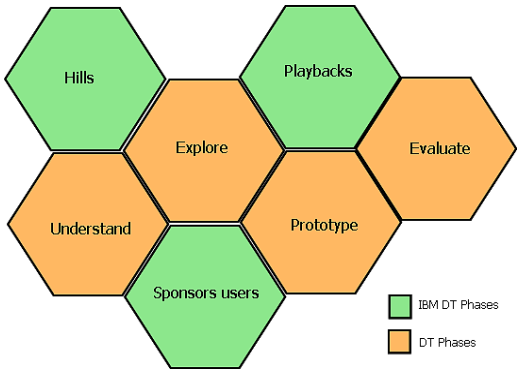
Design Thinking

The Design Thinking (DT) methodology proposed by Brown [2008] is based on empathizing with users and thus being able to define, devise, prototype, and test solutions. User participation and feedback are essential throughout the whole process.

IBM proposes a new approach called IBM Design Thinking [Lucena et al., 2017], prioritizing the following principles:

- Focus on user outcomes.
- Bestless reinvention.
- Diverse empowered teams.

This methodology aims to extend DT to capture user needs in fast-paced incremental software developments, such as cloud-based software. It includes three practices exclusive to the framework: *Sponsor Users, Playbacks, and Hills*. The integration of these practices to DT is shown in figure 6, where the original flow of DT and the three additional steps that IBM proposes can be observed.



Source: [Lucena et al., 2017: 4]

Figure 6 IBM Design Thinking framework.

Lucena et al. [2017] explain the phases of the methodology as follows:

- *Understand phase*: Activities are carried out to understand users and their issues within their environment.
- *Hills*: An individual Hill is a clear goal achievable in one iteration or a finite set of iterations like a user story in SCRUM. Therefore, a Hill should be written to satisfy a specific and clearly defined user problem.
- *Explore*: It focuses on generating new ideas to avoid obvious solutions and thus increase the innovation potential. A common technique used in this phase is brainstorming.
- *Sponsors users*: They participate in all phases of development. Product design and management team members interview the sponsor's users at the beginning of the project.
- *Prototype*: It is the iterative generation of prototypes designed to answer questions to solve the design problem. Prototypes in DT are generally mock-ups that support the development and evaluation of product concepts to discover which shapes are correct or incorrect.

- *Playbacks*: These are checkpoints carried out by the project team and Sponsors Users at the end of each project development phase, to review the project's status and plan the next steps.
- *Evaluate*: In this stage, users are asked for feedback on the prototype created. User experience assessment (UX) techniques could be used to test the prototype.

This methodology is studied by Lindberg et al. [2011]. They address how DT can help to improve innovation in the development of information technologies and what individual and organizational factors facilitate or promote this. They describe the contribution of DT to engineering and how it relates to similar IT development approaches.

Carroll & Richardson [2016] explained the need to develop e-pharmacy software using DT principles to help software developers identify healthcare requirements and extend and enrich traditional software requirements gathering techniques.

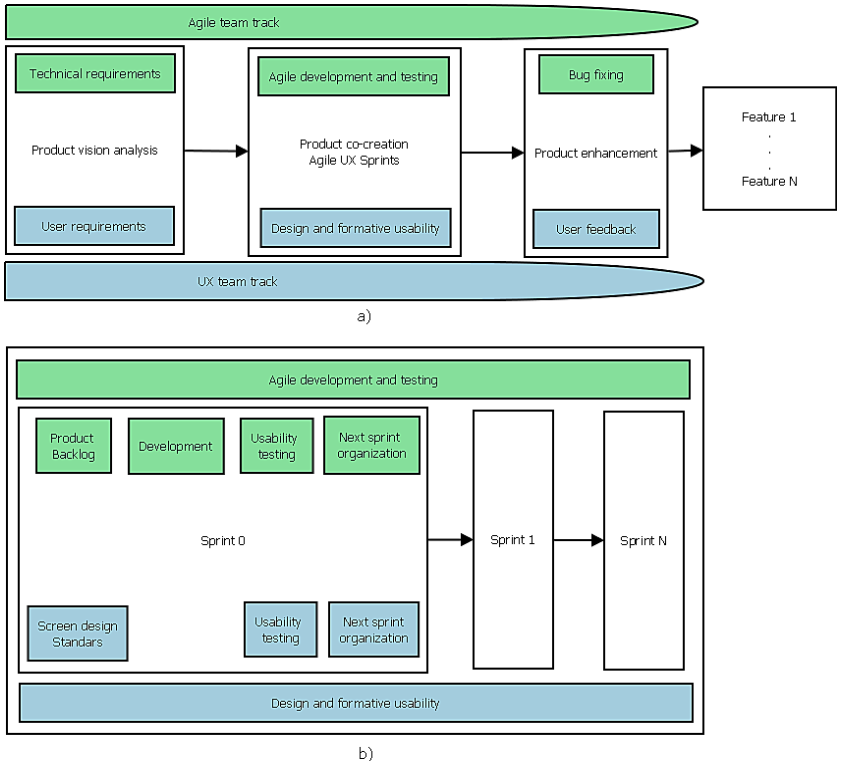
Agile UX

Agile UX facilitates and synchronizes the development and user experience teams [Kieffer et al., 2017]. As described by Kieffer et al. [2017] this methodology is based on the SCRUM process [Tridibesh, 2017], so it takes its development process in sprints and makes it work in conjunction with user experience concepts. The principles of this methodology are described as follows [Kieffer et al., 2017]:

- Describe the iterative, incremental nature of the UX and agile activities organized in parallel synchronized tracks. Furthermore, user engagement throughout the process is highlighted.
- An initial time should be considered to carry out activities to understand user needs and specify functional requirements. The team starts with a shared vision of the product to develop. This state is similar to the Sprint 0 required in parallel tracks.
- Features are defined to guide the development of the product prototype.
- Continuous delivery of working software should be carried out, including implementing rapid formative usability to meet UX goals.

- Documentation of findings during the initial analysis, design, and usability should be carried out to guide the UX deliverable.

In figure 7a the workflow of this methodology is illustrated. As it can be seen, the work is developed in parallel between the agile and the UX teams. Several sprints are carried out iteratively, each one resulting in a new characteristic until the final product is obtained. In addition, figure 7b shows how the development of an adapted sprint works to meet the goals of user-centered design.



Source: [Kieffer et al., 2017: 579]

Figure 7 a) Agile UX development cycle. b) Agile UX Sprint.

A sprint 0 is defined where the backlog or product requirements are obtained, the planning of the next Sprint, and the initial design process. In this way, each Sprint has the same stages as Sprint 0.

Perez-Medina et al., [2019] reports the importance of using the principles of this methodology and some standards such as ISO IEC 9241-210 to develop a web platform for the rehabilitation of domestic motors. The results reported that the

platform was useful, effective, efficient, easy to use, and its interfaces were acceptable. In general, the participants were satisfied with the use of the platform.

Samwi et al. [2020] developed a website to contact university students with companies and avoid long job search processes. The development was based on Scrum with a user-centered design to allow their participation. A mixed-method approach was applied to ensure the integrity of the information. This method included key informant interviews, observations, and requirements workshops for obtaining portal requirements. User-determined requirements were used to guide the initial interface designs that were later turned into clickable pages using a pencil.

3. Results

In table 2, we can compare the methodologies analyzed in this work using the analysis criteria proposed by Schwartz [2013].

Table 2 Comparison of agile methodologies with user experience.

Feature		Parallel Tracks	Lean UX	Design Sprint	Agile UX	Design Thinking
UCD Activities	Specify context of use	X	X	X	X	X
	Specify users' needs	X	X	X	X	X
	Design	X	X	X	X	X
	Evaluate	X	X	X	X	X
UCD Principles	Design based on explicit understanding of users	X	X	X	X	X
	Users involved	X	X	X	X	X
	Design driven and refined by user-centered evaluation	X	X	X	X	X
	Iterative process	X	X	X	X	X
	Process addresses the whole user experience	Not ensured	X	X	X	X
	Team includes multidisciplinary skills	X	X	X	X	X
Agile Principles	Individual and interactions over processes and tools	Not ensured	X	X	X	X
	Working software over comprehensive documentation	Not ensured but promoted	X	X	X	X
	Customer collaboration over contract negotiation	Not ensured	X	X	X	X
	Responding to change over following a plan	X	X	X	X	X

Based on: [Schwartz, 2013: 350]

A comparison was made with the main activities established by ISO IEC 9241-210 before mentioned user-centered design and the principles of this and the agile

manifesto. In the analysis presented in table 2, we have included the Parallel Tracks methodology also analyzed by Schwartz [2013] and another four methodologies. The main difference of table 2 with the results presented by Schwartz [2013] is that we are considering another four methodologies.

4. Discussion

As shown in table 2, we can see that most of the methodologies meet practically all the metrics evaluated. At first glance, any of them could be viable for agile development with user experience. So, how do we know which one to use? It depends on human and economic resources, time, team experience, etc. Table 3 shows the aspects to be considered when adapting each methodology to different applications and work teams.

The parallel tracks methodology has concepts that are not optimal for new companies or teams. Since two separate teams (one for each track) and a sprint 0 are required, the project's cost may increase. This methodology can be used for new or legacy projects, but the multidisciplinary team must include developers, designers, leaders, and testers.

Table 3 Characteristics of each methodology.

Evaluation criteria	Parallel Tracks	Lean UX	Design Sprint	IBM Design Thinking	Agile UX
Team size (Number of team members)	7 to 10 for track (2 tracks)	5 to 7	5 to 7	7 or more	7 to 10 for track (2 tracks)
Requires Sprint 0	Yes	No	No	No	Yes
Development type	New or legacy software	New software or test ideas	Test ideas	New or legacy software	New or legacy software
Team experience level	High	New or High	New	High	High
Monetary cost	High	Low	Low	Low	High
Multidisciplinary team	The team must have a leader, designers, developers, and testers	The team should have developers, project managers, UX designers, QA, and Marketing specialists	The team should have people from all departments involved in a project	The team should have developers, designers, a co-creator, a coach, an advocate, and a leader	The team must have a leader, designers, developers, and testers

Source: [Own elaboration]

Lean UX does not require sprint 0 and is recommended for new teams without much experience due to the learning curve. Everyone works together, contributing ideas and making decisions. It can be used for new projects or to test ideas, which reduces costs since it avoids developing software that does not contribute as a valuable product.

Design Sprint does not require starting development as it is to test ideas. If the idea seems to work, the development begins, and it is recommended for new teams at a low cost since the process takes a week. IBM Design Thinking requires a larger team with more experience for new or legacy projects. However, the cost can be reduced because unnecessary developments and wasted time are avoided with a team that knows the methodology. Finally, Agile UX is similar to parallel tracks but improved since it has all the criteria of UX and agile. The team must have two groups, one for development and the other for design, and there must be a sprint 0. The difference with parallel tracks is that Agile UX involves stakeholders throughout the process, taking SCRUM concepts.

5. Acknowledgements

Erick Franco Gaona was supported by CONACyT, grant 762047.

6. Bibliography and References

[1] Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., 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. Agile Manifesto. https://agilemanifesto.org/

[2] Brown, T. (2008). Design Thinking. Harvard Business Review, 1–10. https://doi.org/10.1145/3347709.3347775

[3] Carroll, N., & Richardson, I. (2016). Aligning healthcare innovation and software requirements through design thinking. Proceedings - International Workshop on Software Engineering in Healthcare Systems, SEHS 2016, April, 1–7. https://doi.org/10.1145/2897683.2897687

[4] Fay, D. J., Roberts, A., & A StantonNeville, N. (2019). Interfaces with Legs? Documenting the Design Sprint of Prototype Future Submarine Control Room User Interfaces. Advances in Human Factors of Transportation, 964, 545–556. https://doi.org/10.1007/978-3-030-20503-4

[5] Fox, D., Sillito, J., & Maurer, F. (2008). Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. Proceedings - Agile 2008 Conference, 63–72. https://doi.org/10.1109/Agile.2008.78

[6] González, C., Toledo, P., & Muñoz, V. (2015). Agile human centered methodologies to develop educational software. DYNA (Colombia), 82(193), 187–194. https://doi.org/10.15446/dyna.v82n193.53495

[7] Gothelf, J., & Seiden, J. (2016). Lean UX Designing Great Products with Agile Teams (C. Cole (ed.); 2da ed.). O'Reilly Media, Inc. http://tdavis.davissoft.com/Lean_UX.pdf

[8] International Organization for Standardization (ISO). (2010). Ergonomics of human-system interaction. Part 210: Human-centred design for interactive systems (ISO (ed.); 1ra ed., Issue 1). https://doi.org/10.5100/jje.30.1

[9] Kieffer, S., Ghouti, A., & Macq, B. (2017). The Agile UX Development Lifecycle: Combining Formative Usability and Agile Methods. Proceedings of the 50th Hawaii International Conference on System Sciences (2017), 577–586. https://doi.org/10.24251/hicss.2017.070

[10] Knapp, J., Zeratsky, J., & Kowitz, B. (2016). SPRINT How to solve big problems and test new ideas in just five days (1st ed.). Simon & Schuster.

[11] Liikkanen, L. A., Kilpiö, H., Svan, L., & Hiltunen, M. (2014). Lean UX - The next generation of user-centered Agile development? Proceedings of the NordiCHI 2014: The 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, 1095–1100. https://doi.org/10.1145/2639189.2670285

[12] Lindberg, T., Meinel, C., & Wagner, R. (2011). Design Thinking: A Fruitful Concept for IT Development? In Design Thinking (pp. 3–18). Springer. https://doi.org/10.1007/978-3-642-13757-0

[13] Lucena, P., Braz, A., Chicoria, A., & Tizzei, L. (2017). IBM design thinking software development framework. 7th Brazilian Workshop on Agile Methods, 1–12. https://doi.org/10.1007/978-3-319-55907-0_9

[14] Martinez, W., Threatt, A. L., Rosenbloom, S. T., Wallston, K. A., Hickson, G. B., & Elasy, T. A. (2017). A Patient-Facing Diabetes Dashboard Embedded in a Patient Web Portal: Design Sprint and Usability Testing. JMIR Human Factors, 5(3), 1–17. https://doi.org/10.2196/humanfactors.9569

[15] Perez-Medina, J. L., Gonzalez, M., Pilco, H. M., Beatriz Jimenes Vargas, K., Acosta-Vargas, P., Sanchez-Gordon, S., Calle-Jimenez, T., Esparza, D., & Rybarczyk, Y. (2019). Usability study of a web-based platform for home motor rehabilitation. IEEE Access, 7, 7932–7947. https://doi.org/10.1109/ACCESS.2018.2889257

[16] Ries, E. (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically (1st ed.). Currency New York.

[17] Rojas, L. A., & Macías, J. A. (2015). An agile information-architecture-driven approach for the development of user-centered interactive software. ACM International Conference Proceeding Series, 1–8: https://doi.org /10.1145/2829875.2829919.

[18] Samwi, E. R., Michael, K., & Moebs, S. (2020). The Role of a Web Portal to Facilitate Higher Learning Institutions Students' Field Attachment in Tanzania. Journal of Information Systems Engineering and Management, 5(4), em0130. https://doi.org/10.29333/jisem/9140

[19] Schwartz, L. (2013). Agile-User Experience Design: an Agile and User-Centered Process? ICSEA 2013, The Eighth International Conference on software engineering advances, 8, 346–351: http://www.thinkmind.org /index.php?view=article&articleid=icsea_2013_12_20_10176.

[20] Shankarmani, R., Pawar, R., S. Mantha, S., & Babu, V. (2012). Agile Methodology Adoption: Benefits and Constraints. International Journal of Computer Applications, 58(15), 31–37. https://doi.org/10.5120/9361-3698

[21] Silva da Silva, T., Selbach Silveira, M., Maurer, F., & Hellmann, T. (2012). User Experience Design and Agile Development: From Theory to Practice.

Journal of Software Engineering and Applications, 05(10), 743–751. https://doi.org/10.4236/jsea.2012.510087

[22] State Of Agile. (2020). 14th annual STATE OF AGILE REPORT. In Annual Report for the STATE OF AGILE (Vol. 14, Issue 14). https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report

[23] Sy, D. (2007). Adapting usability investigations for agile user-centered design. Journal of Usability Studies, 2(3), 112–132.

[24] Tridibesh, S. (2017). A Guide to the Scrum Body Of Knowledge (3rd ed.). SCRUMstudy.