Design Thinking in Practice

Mary Sánchez-Gordón
Østfold University College

Ricardo Mendoza-González
Tecnológico Nacional de México / IT Aguascalientes

Ricardo Colomo-Palacios
Østfold University College

***Abstract*—Technology companies are under enormous pressure to remain competitive. In this situation, the time that takes to reach the market for an innovative idea about a product or service is crucial. Large corporations in the IT industry like SAP, Google, IBM, Microsoft, Apple and startups acknowledge that Design Thinking—along with other approaches—is one promising avenue to become more innovative and helps to accelerate the process. This article presents a brief overview of Design Thinking as well as what, in practice, technology companies are actually implementing under the rubric of "Design Thinking" and the challenges reported on a survey conducted among professionals in this field.**

■ **INNOVATION** is one of the most challenging aspects for all organizations. In particular, technology companies are under enormous pressure to remain innovative. DT implies the cognitive, strategic, and practical processes to develop design concepts such as proposals for new products, buildings, machines by designers and/or design teams [1]. Despite the increasing interest in Design Thinking (DT), a single, agreed upon definition does not exist.

In particular, according to Tim Brown, CEO of IDEO (www.ideo.com), one of the world's largest design companies, DT is "a methodology that imbues the full spectrum of innovation activities with a human-centered design ethos." While according to David Kelley, one of the founders of IDEO, DT is "a method for how to come up with [...] breakthrough ideas that are new to the world, especially with respect to complex projects, complex problems." In contrast, Roger Martin from the Rotman School of Management emphasizes the thinking element, defining design thinking as "the productive mix of analytical thinking and intuitive thinking." However, the popularity of DT could be understood from an innovation perspective, as a source of inspiration or as "a way of thinking" that non-designers can use. In this way, the design practice and the way designers make sense of their task is captured.

DT is not a new concept; it has been around since the 1970s, when the first view of the design process that underlies "Design Thinking" as it is practiced today appeared [2]. DT is well-known as a problem-solving method and enabler of innovation that has demonstrated its value across industries. In general, DT is seen as a particular social, rather than physical technology, to facilitate innovation speed and success [2]. However, DT is relatively new to the IT industry, especially in the agile context [3].

DT and agile methods are both characterized by iteration, experimentation and a clear focus on users' needs.

However, DT should not be confused with agile methods because they do not necessarily foster an innovative approach. Agile methods tend to focus primarily on activities within production processes while DT offers an innovative way of thinking based on divergence and convergence throughout a mixture of activities (e.g., iteration, and experimentation), skills (e.g., ability to visualize), orientations (e.g., gestalt view; tolerance for ambiguity), and logics (e.g., abductive reasoning) [4]. DT, by its nature, allows constant communication among development teams, stakeholders, and target users including different kinds of methods and tools that gather useful information in order to cover users' needs. In this way, DT contributes to discover new aspects of the market, and triggers creative ideas and solutions (i.e., improved quality of choices available).

Agile methods and DT are not the same, but they can be complementary. Some mixed approaches adopted by practitioners around the world are i) Dschool, and SCRUM, or Lean, ii) IDEO and SCRUM, or Lean, and iii) ISO 9241-210: 2010 and SCRUM, Agile, or Lean [5]. Indeed, the success of the DT process has been proven in many technology companies, from large corporations like SAP, Google, IBM, Microsoft, Apple to startups [6]. Google Ventures provides a set of case studies (see https://sprintstories.com/tagged/case-study) from teams around the world.

However, the DT impact in companies may vary according to (because of) their capabilities and combinations of different DT approaches into their home-grown processes. Taking into account the growing interest in DT approaches, their expected impact in the future, and given that an academic understanding and research of this phenomenon is very limited, this article describes a brief overview of DT. Then, we present the findings from a survey devoted to investigate what, in practice, technology companies are actually implementing under the rubric of "Design Thinking", and the challenges they face in implementing it. The survey results reveal that the main goal is to develop a user-oriented product/service. DSchool, IDEO, and Google Design Sprint are the most reported DT approaches. However, there was no clear distinction among the perceived implementation challenges by practitioners in this survey.

APPROACHES FOR DT

The DT process consists of a set of stages which differ in definition from institution to institution. Although there are different levels of detail and focus in the various approaches of DT, a flexible 3-stages process —data gathering, idea generation and testing— can be discerned along with a common set of tools and prescribed ways of thinking [7].

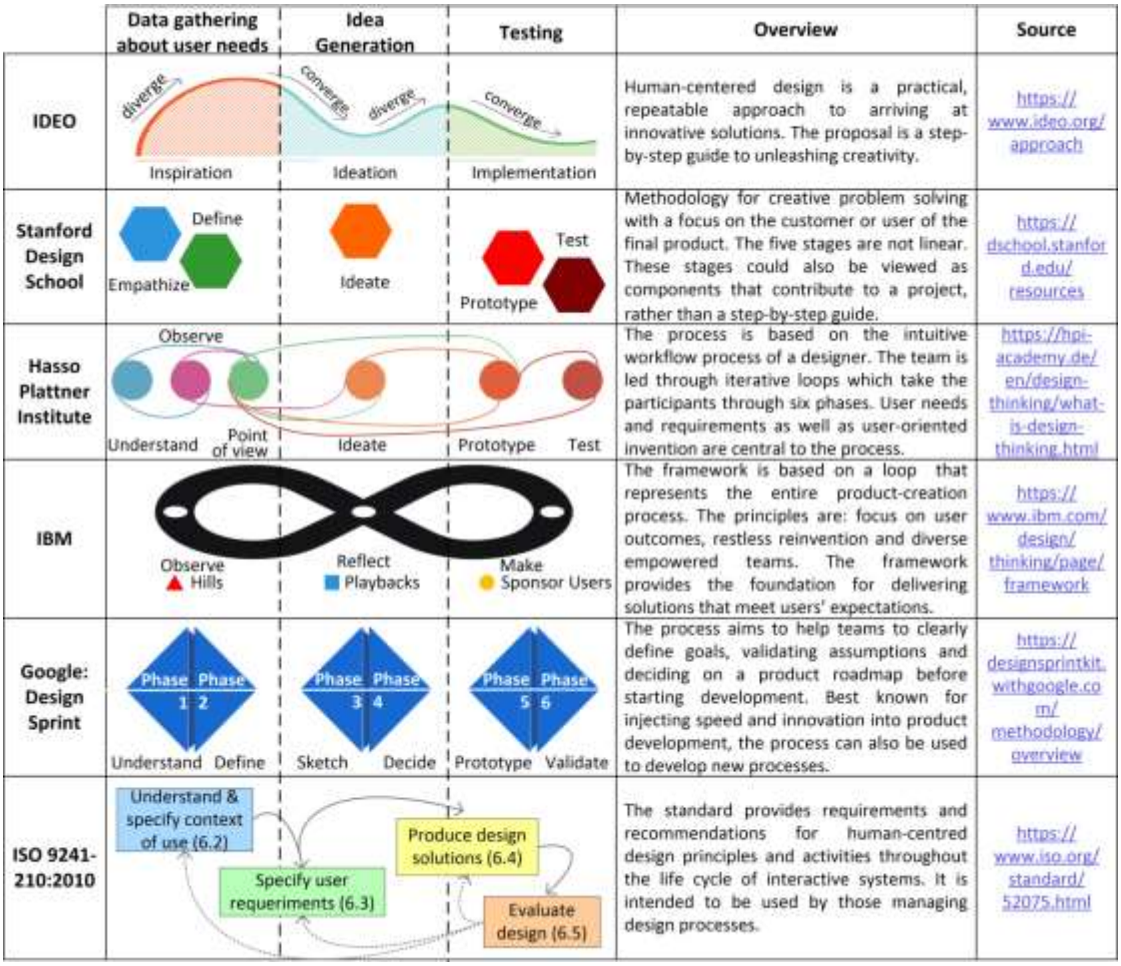
Table 1 shows six popular DT approaches. The first three are well-known in the traditional design domain. These approaches have been proposed by IDEO, Stanford Design School and Hasso Plattner Institute at the University of Potsdam. Indeed, they have served as an inspiration for IBM and Google approaches which emerged in the IT industry. All of them include a wide variety of tools ready to use in order to support the DT process. However, understanding when to use them and under what circumstances requires knowledge and skills (i.e., training). On the other hand, ISO 9241-210:2010 provides requirements and recommendations for human-centred design principles and activities throughout the life cycle of computer-based interactive systems. This part of ISO 9241 is concerned with ways in which both hardware and software components of interactive systems can enhance human–system interaction. Taken all together, they illustrate the ambiguity that characterizes the discourse around DT in IT industry.

The three stages of a process are also shown in **Table 1**. The first stage "data gathering" is concerned primarily with what is —understanding the current situation— and using that understanding as a basis for creating "idea generation" that then lies on the third stage, which is "testing". Finally, these DT approaches provide a useful structure to the innovation process that balances the need for rigor and the need for creativity while enabling a wide range of useful and creative explorations. Although they picture the process as a linear sequence of steps, in practice the DT process could be carried out in a more flexible and non-linear fashion. That means DT often occurs in parallel and is repeated iteratively, without the need of following any specific order.

CHALLENGES IN AGILE METHODS

Classical innovation, strategy or re-engineering projects could be conducted and be merely labeled as DT. Therefore, a first challenge is to understand and master DT principles and foundations. From an implementation perspective, practitioners need to be able to comprehend how ideas have emerged through the DT process [8]. Moreover, given that a realistic assessment of the feasibility of ideas is needed, communication between implementation and DT teams should start early in the life cycle of projects. In particular, it could be hard to maintain an adequate hierarchical communication structure while scaling DT and agile methods from individual teams to company-wide level.

Table 1. Some popular Design Thinking approaches distributed in a three stages process: data gathering, idea generation and testing (adapted from [7]).



	Data gathering about user needs	Idea Generation	Testing	Overview	Source
IDEO	diverge Inspiration	converge diverge Ideation	converge Implementation	Human-centered design is a practical, repeatable approach to arriving at innovative solutions. The proposal is a step-by-step guide to unleashing creativity.	https://www.ideo.org/approach
Stanford Design School	Define Empathize	Ideate	Test Prototype	Methodology for creative problem solving with a focus on the customer or user of the final product. The five stages are not linear. These stages could also be viewed as components that contribute to a project, rather than a step-by-step guide.	https://dschool.stanford.edu/resources
Hasso Plattner Institute	Observe Understand Point of view	Ideate	Prototype Test	The process is based on the intuitive workflow process of a designer. The team is led through iterative loops which take the participants through six phases. User needs and requirements as well as user-oriented invention are central to the process.	https://hpi-academy.de/en/design-thinking/what-is-design-thinking.html
IBM	Observe ▲ Hills	Reflect ■ Playbacks	Make ● Sponsor Users	The framework is based on a loop that represents the entire product-creation process. The principles are: focus on user outcomes, restless reinvention and diverse empowered teams. The framework provides the foundation for delivering solutions that meet users' expectations.	https://www.ibm.com/design/thinking/page/framework
Google: Design Sprint	Phase Phase 1 2 Understand Define	Phase Phase 3 4 Sketch Decide	Phase Phase 5 6 Prototype Validate	The process aims to help teams to clearly define goals, validating assumptions and deciding on a product roadmap before starting development. Best known for injecting speed and innovation into product development, the process can also be used to develop new processes.	https://designsprintkit.withgoogle.com/methodology/overview
ISO 9241-210:2010	Understand & specify context of use (6.2)	Specify user requeriments (6.3)	Produce design solutions (6.4) Evaluate design (6.5)	The standard provides requirements and recommendations for human-centred design principles and activities throughout the life cycle of interactive systems. It is intended to be used by those managing design processes.	https://www.iso.org/standard/52075.html

The importance of adequate user involvement to understand user needs is well known in software engineering. This involvement is key in order to develop a software product or service that satisfies user needs. However, that user involvement frequently requires more than a software development process [9] whether traditional, agile or hybrid. Although the Agile Manifesto emphasizes "customer collaboration," its approach does not imply that the team will work towards solving the correct problem. Indeed, agile methods have already included user feedback as part of the requirement process and user stories to capture all the faces of user needs. For instance, Cohn [10] proposed to introduce a Customer Team and adopt observation techniques, conduce user interviews, apply questionnaires, create personas and develop user interfaces prototypes. At first glance, similar techniques are applied in DT. However, DT puts more emphasis on interdisciplinary creative collaboration and team diversity while divergent thinking seems to be avoided in Agile methods in order to keep the overall view on what to do next [11]. After all, DT is not only a toolkit but also a cognitive process or a mindset for innovation in which empathy is an essential element. As not all IT projects need or provide the opportunity for the same level of DT activities, the right selection of methods is another challenge, especially in the early stages of their development.

There are some popular DT approaches, but certainly, there are very little practical examples of adoption of DT in the academic literature (e.g., [12]–[18]) and also there is a limited understanding of what happens when

DT is adopted in an agile software development project [5], [18]. For instance, the positive and negative implications of its use in real projects are unclear. Thus, the emergence of DT gives rise to several questions: How to measure impact of DT in Agile methods? How will the time spent on understanding the problem before the actual development process starts affect the total project timeline and cost? To what extent will there be fewer change requests due to the adoption of DT? Finally, neither DT nor Agile methods provide support on how to track growth and how to scale a software product or service after its launch [19].

Figure 1. Participants' background based on company size, globally distributed projects and work experience.

DT IN PRACTICE

Motivated by the growing attention towards DT in the IT industry, we carried out a survey among practitioners to provide proper empirical figures that would reveal DT approaches and perceived implementation challenges in agile methods. Here, we present an overview of the preliminary results from that survey carried out among 45 practitioners, of which 30 (66.67%) have used some DT approach (see **Figure 1**). Therefore, we focus on the last group of practitioners to get an overview of DT in practice (see details in https://doi.org/10.6084/m9.figshare.9205793.v4). Practitioners are from six countries —Mexico (17), United States (6), Brazil (3), Italy (2), Canada (1), and Chile (1). They had wide experience in software development projects —77% reported between 6 and 20+ years of experience in the area—, and 70% worked in large companies while 87% were involved in regionally, nationally, or globally distributed projects.

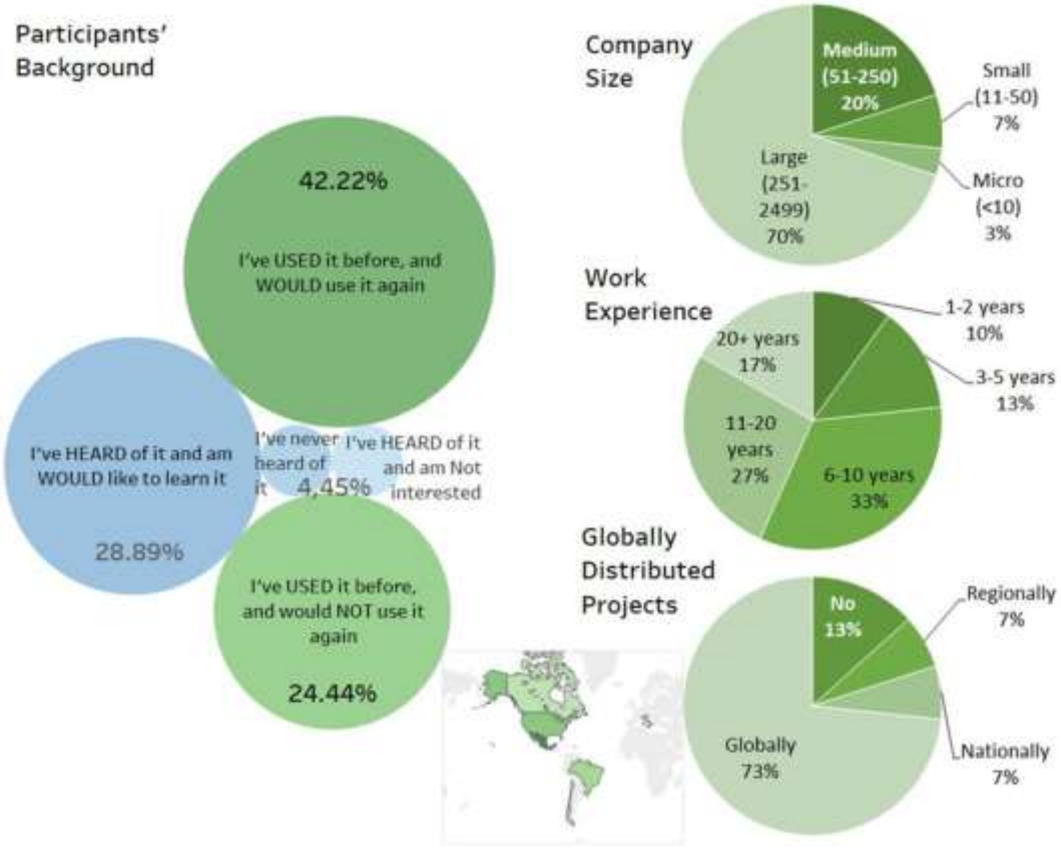
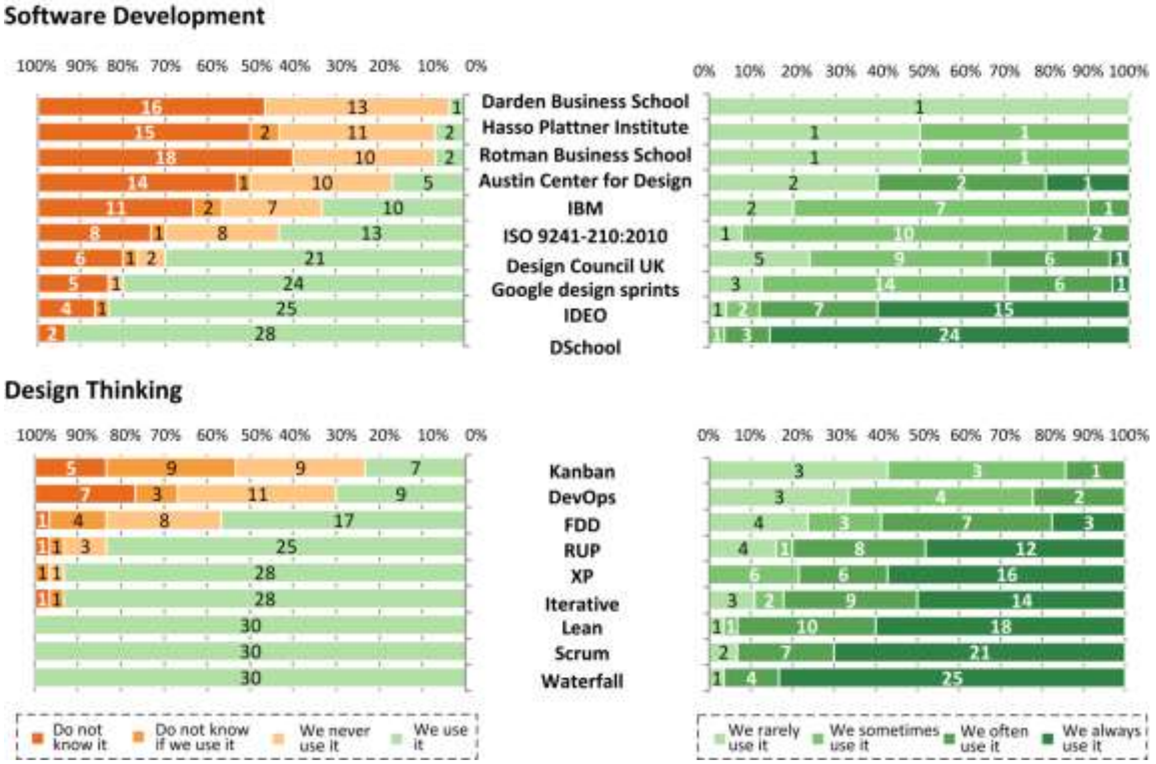


Figure 2 shows a set of software development frameworks and methods implemented by the respondents. Although the most used were Waterfall, Scrum, Lean, and XP, all of the respondents stated that they used practices from different approaches (i.e., hybrid approaches). **Figure 2** also depicts the use of DT methods and

frameworks. The most reported were DSchool followed by IDEO, and Google Design Sprint. However, one interesting point is that 50% of respondents stated that they integrated DT and software development approaches in existing new product development (NDP) work —voluntary, on individual basis— while 20% used DT in a workshop format to support/coach innovation projects. In fact, 23% reported that a special innovation function used DT. Regarding how DT approaches are applied (tailored) in regular projects, the answers were distributed equally in three groups. One group of practitioners reported that they do not consider a particular tailoring approach of DT in their projects. On the contrary, another group defined a tailoring approach that continuously guides the application of DT. While another reported that at the beginning of a project, the project lead/designer tailors the DT process based on experience.

Figure 2. Frequency of use of Software development approaches and DT approaches.

Additionally, the main goal that participants aimed to address with DT was to develop a user-oriented product/service, followed by gaining an increased adaptability, improved quality of choices, and increased implementation of ideas. However, it should be noted that few respondents (less than 5) were in a neutral position facing the last three goals. Furthermore, enhanced likelihood of successful implementation as well as decreased risk and cost of failure were pursued by most of the practitioners (80%). Finally, around 50% of participants paid greater attention in fostering team alignment and collective learning, as well as building engagement and trust.



All of the practitioners also reported that they measured the impact of DT. The most popular measures were questionnaires and surveys from participants (e.g., practitioners, employees, and consumers) in NDP processes, followed by the level of satisfaction based on customer feedback, and traditional Key Performance Indicators (KPIs) such as financial performance, market success, and revenue outcome of DT projects. Although almost all respondents (97%) claimed that they were generally satisfied with their current DT approach, 73% stated that if they could, they would change/improve it while 37% would not use it again.

In spite of that, there was no clear distinction among the perceived implementation challenges by participants. The findings reveal that half of the respondents somewhat disagree that value of DT is difficult to measure using metrics (e.g., KPIs). Apart from that, 50% also disagree or somewhat disagree that (i) DT principles/mindsets clash with organizational culture; (ii) DT ways of working are hard to integrate in existing process and structures; (iii) resulting ideas and concepts are difficult to implement; and (iv) it is difficult to build diverse teams. Likewise, 76% disagree or somewhat disagree that DT skills are difficult to learn and vocabulary from the design world and artifacts are less accepted. However, it is worth stressing that all the challenges aforementioned can diminish the beneficial effect and jeopardize valuable DT adoption.

CONCLUSION

While DT has brought innovation in many sectors and the literature review leads to the conclusion that DT has significant added value for IT consulting and developing practices, IT industry operates in a different paradigm than from which DT is derived. Therefore, despite the increasing interest for this approach in IT field it should not be considered as a “silver bullet.” Succeeding with agile methods and DT is not a matter of course. Rather, the approach needs to be adapted to changing needs during a project lifecycle and its adoption should be carefully evaluated depending on the company’s business area and the target application domain of the project. Therefore, there is a need for systematic investigations in this area.

Furthermore, although we survey a relatively small number of practitioners, they agreed that DT skills, ideas, and concepts are easy to learn. Thus, the challenge seems to be not only an effective application of the DT and agile practices in a way that is appropriate to the context but also a change of mindset and the development of right attitude for building empathy with the users and approaching this kind of thinking. Consequently, the tentative conclusion is that the difference from other approaches lies in the attitude or mindset, more than the specific practices and tools, and most significantly, DT is emerging as a very appealing approach to developing the capability for IT-enabled business innovation in organizations.

■ REFERENCES

[1] B. Penzenstadler, “When Does Design Help Thinking, and When Does Design Thinking Help?,” *IEEE Softw.*, vol. 37, no. 2, pp. 6–9, Mar. 2020, doi: 10.1109/MS.2019.2958263.

[2] J. Leidtka, “Exploring the impact of design thinking in action,” University of Virginia Darden School of Business, 2018.

[3] W. M. D. Ruchira Prasad, G. I. U. S. Perera, K. V. Jeeva Padmini, and H. M. N. Dilum Bandara, “Adopting Design Thinking Practices to Satisfy Customer Expectations in Agile Practices: A Case from Sri Lankan Software Development Industry,” in *2018 Moratuwa Engineering Research Conference (MERCon)*, Moratuwa, 2018, pp. 471–476, doi: 10.1109/MERCon.2018.8422006.

[4] P. Micheli, S. J. S. Wilner, S. H. Bhatti, M. Mura, and M. B. Beverland, “Doing Design Thinking: Conceptual Review, Synthesis, and Research Agenda,” *J. Prod. Innov. Manag.*, vol. 36, no. 2, pp. 124–148, 2019, doi: 10.1111/jpim.12466.

[5] J. C. Pereira and R. de F. S. M. Russo, “Design Thinking Integrated in Agile Software Development: A Systematic Literature Review,” *Procedia Comput. Sci.*, vol. 138, pp. 775–782, Jan. 2018, doi: 10.1016/j.procs.2018.10.101.

[6] H. Park and S. McKilligan, “A Systematic Literature Review for Human-Computer Interaction and Design Thinking Process Integration,” in *Design, User Experience, and Usability: Theory and Practice*, 2018, pp. 725–740.

[7] J. Liedtka, “Perspective: Linking Design Thinking with Innovation Outcomes through Cognitive Bias Reduction,” *J. Prod. Innov. Manag.*, vol. 32, no. 6, pp. 925–938, 2015, doi: 10.1111/jpim.12163.

[8] F. Häger, T. Kowark, J. Krüger, C. Vetterli, F. Übernickel, and M. Uflacker, “DT@Scrum: Integrating Design Thinking with Software Development Processes,” in *Design Thinking Research: Building Innovators*, H. Plattner, C. Meinel, and L. Leifer, Eds. Cham: Springer International Publishing, 2015, pp. 263–289.

[9] L. Corral and I. Fronza, “Design Thinking and Agile Practices for Software Engineering: An Opportunity for Innovation,” in *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, New York, NY, USA, 2018, pp. 26–31, doi: 10.1145/3241815.3241864.

[10] M. Cohn, *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley Professional, 2004.

[11] T. Lindberg, C. Meinel, and R. Wagner, “Design Thinking: A Fruitful Concept for IT Development?,” in *Design Thinking: Understand – Improve – Apply*, C. Meinel, L. Leifer, and H. Plattner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3–18.

[12] M. M. Higuchi and D. N. Nakano, “Agile Design: A Combined Model Based on Design Thinking and Agile Methodologies for Digital Games Projects,” *Rev. Gest. E Proj. - GeP*, vol. 8, no. 2, pp. 109–126, Sep. 2017, doi: 10.5585/10.5585.

[13] N. Carroll and I. Richardson, “Aligning Healthcare Innovation and Software Requirements Through Design Thinking,” in *Proceedings of the International Workshop on Software Engineering in Healthcare Systems*, New York, NY, USA, 2016, pp. 1–7, doi: 10.1145/2897683.2897687.

[14] K. Gurusamy, N. Srinivasaraghavan, and S. Adikari, “An Integrated Framework for Design Thinking and Agile Methods for Digital Transformation,” in *Design, User Experience, and Usability: Design Thinking and Methods*, 2016, pp. 34–42.

[15] F. L. Eickhoff, M. L. McGrath, C. Mayer, A. Bieswanger, and P. A. Wojciak, “Large-scale application of IBM Design Thinking and Agile development for IBM z14,” *IBM J. Res. Dev.*, vol. 62, no. 2/3, pp. 1:1-1:9, Mar. 2018, doi: 10.1147/JRD.2018.2795879.

[16] A. Duenser, D. Smith, J. Freyne, E. Gilleran, K. de Salas, and H. Cinis, “Engaging Assessments: Interface Design of a Referral Support Tool for Adults and Children,” in *Proceedings of the 30th Australian Conference on Computer-Human Interaction*, New York, NY, USA, 2018, pp. 42–52, doi: 10.1145/3292147.3292166.

[17] A. Sebok, B. Walters, and C. Plott, “Integrating Human-Centered Design and the Agile Development Process for Safety and Mission Critical System Development,” *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 61, no. 1, pp. 1086–1090, Sep. 2017, doi: 10.1177/1541931213601876.

[18] E. D. Canedo and R. Parente da Costa, “The Use of Design Thinking in Agile Software Requirements Survey: A Case Study,” in *Design, User Experience, and Usability: Theory and Practice*, 2018, pp. 642–657.

[19] F. Dobrigkeit, D. de Paula, and M. Uflacker, “InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup,” in *Design Thinking Research : Looking Further: Design Thinking Beyond Solution-Fixation*, Cham: Springer International Publishing, 2019, pp. 199–227.