

CENTERIS - International Conference on ENTERprise Information Systems /  
ProjMAN - International Conference on Project MANagement / HCist - International  
Conference on Health and Social Care Information Systems and Technologies,  
CENTERIS/ProjMAN/HCist 2018

# Design Thinking Integrated in Agile Software Development: A Systematic Literature Review

Julio Cesar Pereira<sup>a\*</sup>, Rosaria de F. S. M. Russo<sup>a</sup>

<sup>a</sup> *Uninove - Universidade 9 de Julho, Rua Deputado Salvador Julianelli, s/n - 1o. and, São Paulo 01156-080, Brazil*

---

## Abstract

Software development raises the relevance as it enables automation of the processes and provides new functionalities to the final consumers; Agile Software Development is an enabler to accelerate software delivery, manage priorities changes and increase productivity. During the software development life cycle, the development team is expected to fully understand customers' needs, but it is challenged by geographic constraints, narrow project schedule, or short budget. Design Thinking is an approach that promotes the understanding of customer needs considering what is technical and economic feasible. To evaluate how the Design Thinking approach is used integrated with Agile Software Development methodologies, we applied the Systematic Literature Review, which collected, categorized and reviewed 29 articles related to this topic. The results show that most of the integrated models are applied throughout the software life cycle, noting that the model that integrates the DT approach published by the International Organization for Standardization (ISO) and Scrum as an agile methodology composes the most frequently used integrated model. The integrated models resulted in a greater approximation of end users and the development team, improving the quality and usability of the software.

© 2018 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Selection and peer-review under responsibility of the scientific committee of the CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies.

**Keywords:** Information Systems; Software; Agile Methodology; Design Thinking; Project Management.

---

\* Corresponding author. Tel.: +55 11 97208-4720.

E-mail address: [juliopereira.pmp@gmail.com](mailto:juliopereira.pmp@gmail.com)

## 1. Introduction

Information Technology (IT) has become more relevant, as an enabler of product and service, as people gain instant access to the global market, and the high expectations regarding provided services [1]. Thus, organizational strategy considers the massive use of the internet and computer systems, with an emphasis on a competitive advantage against competitors [2,3]. Executives report that the application of IT benefits organizations' revenue growth [4]. In this context, the organization must identify the technology application area and recognize the users' needs [5], combining agility and quality on the deployed software, simultaneously. Software is an element that introduces new technologies or delivers services in organizations [6].

The software deployment activity becomes complex due to the need of direct user participation in software modeling to generate a solution in an integrated manner [7,8]. The usage of Agile Software Development (ASD) methodology in project management allows software release to users quickly by reducing the time between the design and deployment, promoting both partial tests and deliveries with greater agility compared to classic project management methods. Among the practices adopted in the ASD we find (i) face-to-face communication, (ii) iteration planning meeting and retrospectives facilitated by self-organizing cross-functional teams, and (iii) continuous integration with testing facilitates short iterations and releases [9].

In the organizational perspective, ASD promotes collaboration between stakeholders of software projects, with transparency and bigger perceived value by the customer, in addition to improving the product's quality and reducing the time to address possible errors [9,10]. On the other hand, 84% of organizations are still in a low maturity agile level and 40% of agile practitioners indicate this method as a way to manage distributed teams, whereas 22% indicate it as a method to reduce costs in software development projects [11]. To solve this issue, the organization needs to create mechanisms that associate consumers' desires to the product or service being developed which is one of the goals of the Design Thinking (DT) approach. The DT or Human Centered Design (HCD) approach acts from the application of the designer's thought to identify problems and building desirable solutions for customers, that are technologically feasible and viable for the organization's business [12].

Some researchers have adopted a Systematic Literature Review (SLR) method to evaluate the integrated DT approach with agile methods and their application of the combined processes, principles [13], how the integrated method is applied on software design stages and the software usability enhancement [14]. The objective of this study is to evaluate how the DT approach and the ASD method occurs throughout the life cycle of software development projects, what are the variations of this integration, considering the different approaches and models available.

## 2. Design Thinking

The DT approach generates potential solutions to a problem by providing better products and services, and an increase in productivity and operational improvements [15]. It aims, through the HCD, to capture what is desirable to people, which is technically possible and viable for business, and converting it into both consumer benefit and business value [12]. This approach has some variations in the stages proposed by the existing models as shown in Table 1, although the improvement cycles through repetition permeate all the stages, including the identification of a problem and the rapid exploration of ideas and possible solutions [12].

Table 1. Different phases and activities of DT approaches.

Design Thinking Model	Phases and activities	Reference
IDEO, International Design and Consulting Firm	Inspiration, ideation and implementation	[12]
Institute of Design at Stanford (DSchool)	Empathize, define, ideate, prototype and test	[16]
International Organization for Standardization (ISO): Human-centered design for interactive systems	Understand, specify, produce and evaluate	[17]
Google design sprints	Understand, sketch, decide, prototype and validate	[18]
Austin Center for Design, Educational Program	Ethnography, synthesis and prototyping	[19]
Design Council UK: Double Diamond	Define strategy (understand; define) and execute solution (explore; create)	[20]

One of the key points of the DT approach is the visual representation, so that the developing idea becomes tangible and accepted, ensuring that those involved recognize the result as imagined during the creation [15,20]. It is worth highlighting that the process is iterative and can be repeated whenever new components are identified in the decision-making process of continuance, to achieve the expected result. During the early phase present in the DT approach, as in the empathy phase, it is possible for people to make prototypes to help understand the goal, with a greater depth and to create a better result [21].

DT reinforces a structured thought that helps the creative process make new products and services oriented and related to the customer's experience [22]. Integrative thinking consists not only in comparing two opposite alternatives and choosing one, but in analyzing them together to obtain a greater result when compared to the choice of one or the other in the original option [23]. Notice that the DT approach has a broader notion of thinking beyond the rational and objectivity to intuition and creativity [24].

### 3. Agile Software Development

ASD in project management emphasizes an incremental approach for a project software development [21], more than following sequential pre-defined phases. This movement emphasizes (i) the human aspects involved in software development more than processes and tools, (ii) the software tests continuously performed, and (iii) cooperation between developers and customers [25]. Integrating quality practices and processes that quantify the benefits can improve and speed up development activities and, consequently, increase the level of software's customers satisfaction [26].

For this reason, enforcing the ASD allows an organization to have quick answers to the constant changes in the business scenario [27], with focus on the adaptability when faced with uncertainties and frequent changes [28]. According to previous studies, the main methods of Agile Methodology and the most commonly used are: Extreme programming (XP), Scrum, Feature driven development (FDD), Crystal methods, Dynamic systems development methods (DSDM), Adaptive software development (ASD), Lean development [29]. [30] cites other models such as Scaled Agile Framework (SAFe) and Model-driven development (AMDD). In recent years it is noted that using Scrum alone and the hybrid Scrum and XP model are methodologies predominantly used [31].

Agility can be defined as (i) the measure of performance of the project team; (ii) the performance of agility can be influenced by the ability of the organization to manage changes, and (iii) the ability of the organization to address the impacts caused by internal and external factors [32]. The correct enforcement of the ASD can bring benefits to the organization by providing a broad knowledge exchange and favoring the learning of collaborators, bringing more satisfaction, social skills development, constant feedbacks and the trust of professionals [9]. From the customer's perspective, the most visible and significative benefits are product's value, a better relationship with the customer and increased quality of the product being provided [9].

These benefits come from the close collaboration between parties, ensuring mutual responsibility, clearer requirements due to better comprehension of data being treated and more time being dedicated to finding the solution and not just the collection of requirements [10]. The ASD has characteristics that make it into a process defined through activities including short iterations, continuous testing, self-manageable teams, constant collaboration, and frequent re-planning based on the actual reality instead of following potentially obsolete plans [33]. Adaptability, frequent interaction with customers and readiness for motion and efficiency are complementary aspects of ASD [34] through all phases of a software project life cycle.

### 4. Research Methodology

This study aims to evaluate how the DT approach along with the Agile Software Development (ASD) are integrated. For this purpose, we use the systematic literature review method, whose objective is to evaluate and make sense of great volume of information in order to address issues regarding a particular subject [35]. We use the eleven steps established by [35] in this research and detailed above.

1. The definition of the research question and objective cited below;
2. Establish a validation committee composed by two researchers, one expert on DT and the other on Agile Project Management methodologies;

3. The protocol for research consisted of tools to access information in the Web of Science (WoS) and Scopus databases; words being researched ("design thinking" or "human cent\* design" and "agile" and "information system" or "software" and "project\*" in the title, abstract or keywords); scope areas (Computer Science, Engineering, Mathematics, Business, Management and Accounting, Decision Science, Social Science, Materials Science, Energy and Economics, Econometrics and Finance); the article should combine DT and at least one of ASD; it must have been published in the last ten years; and the characteristics of the integration of them. All these characteristics configure the inclusion criteria of the present research;

4. In the bibliographic research, we removed some records because there were conference proceedings papers without author references and some articles were identified in both databases. After this step, we selected articles for the abstract reading;

5. In this step, we removed articles referring to different subjects rather than software development, resulting in a subset. Among the subjects related in those removed articles are the approaches in education, organizational structure, information security, cloud computing, reference architecture, or professional behavior;

6. We removed articles not referring to the use of the methodology with DT integrated to ASD during the software development life cycle, resulting in the list of articles finally selected. Thus, we describe the criteria for excluding articles in this research in steps 4 to 6;

7. The data extraction included: the authors, the referred journal, conference, or book, the year of publication, the subject of the study, the abstract, the exclusion criteria of this study, the type of problem studied, type of software developed, the DT approach used, the ASD method applied, if a new model is suggested, study type (practical, conceptual, or model evaluation) and software development life cycle (SDLC) phases which the model was applied;

8. When critically reviewing the primary articles, we needed more information to verify the ASD methodology, simultaneously applying DT during a software development life cycle;

9. The synthesis of the cases selected was complemented by the literature to support the conclusions;

10 and 11. We wrote an evaluation of the limitations and biases of the studies and a final report.

## 5. Systematic Literature Review Results

### 5.1. Selected articles

We found 33 articles in the WoS and 111 in the Scopus database. We removed 21 records because the conference proceedings lacked author references and we identified other 7 articles duplicated in both databases research. As a preliminary result, we selected 116 articles for the abstract reading. In step five, mentioned on the previous section, we removed 51 articles for referring to different subjects rather than software development, resulting in 66 articles. Of this total, we removed 37 articles for not referring to the use of the methodology with DT integrated to ASD during the software development life cycle, resulting in 29 articles classified and shown in Table 2.

Table 2. References selected in systematic literature review.

Id	Year	DT approach	ASD Methodology	Type of software (SW)	Suggested model	Study Type	SDLC
[36]	2017	IDEO	Scrum	Digital game	Yes	1	2
[37]	2017	DSchool	Scrum	Sales portal and other apps	Yes	1	1
[38]	2017	DSchool	Unspecified	Not applicable	Yes	2	2
[39]	2016	Unspecified	SAFe	Not applicable	Yes	2	2
[40]	2016	IDEO	Lean	Mobile game	Yes	1	1
[41]	2015	Unspecified	Unspecified	Web app	No	1	1
[42]	2011	Unspecified	Unspecified	Integrated enterprise solution	Yes	1	1
[43]	2016	ISO	Scrum	Not applicable	Yes	2	1
[44]	2016	ISO	Kanban	Internet Portal	No	1	1
[45]	2016	ISO	Scrum	Web app	Yes	1	2
[46]	2016	ISO	AMDD	Construction planning	Yes	1	1

Id	Year	DT approach	ASD Methodology	Type of software (SW)	Suggested model	Study Type	SDLC
[47]	2017	DSchool	Scrum	Not applicable	Yes	2	2
[48]	2016	DSchool	Unspecified	Not applicable	Yes	2	1
[49]	2017	Unspecified	Scrum	Web portal	Yes	1	1
[50]	2017	ISO	Agile-like and SAFe	Safety and mission critical system	Yes	1	1
[51]	2016	ISO	Scrum	Not applicable	Yes	2	2
[52]	2015	DSchool	XP, Lean	Data storage app	No	1	1
[53]	2015	ISO	XP	Game app	No	2	1
[54]	2015	ISO	Scrum and Lean	Educational web-based system	No	1	1
[55]	2014	Unspecified	XP	Web-based system	No	1	1
[56]	2014	ISO	Scrum	Unspecified	No	3	4
[57]	2013	Unspecified	Scrum	Technological assistive device	No	1	1
[58]	2011	Unspecified	XP	Not applicable	Yes	2	3
[59]	2011	ISO	XP	Website	No	1	2
[60]	2011	Unspecified	AMDD	Not applicable	Yes	2	1
[61]	2010	Unspecified	Scrum	Educational web-based system	Yes	2	4
[62]	2010	Unspecified	Scrum	Mobile app	Yes	2	1
[63]	2009	Unspecified	Unspecified	Unspecified	Yes	2	2
[64]	2008	Unspecified	XP	Content manager app	Yes	1	1

NOTE: Study Type 1 is a practical case, 2 is conceptual, 3 is a model evaluation.

NOTE2: How was DT integrated with ASD during SDLC: 1 – throughout the entire software development life cycle, 2 - is applied only in the design or requirement phases of the software, 3 - applied only in the phases of acceptance and deployment tests, 4 - unspecified.

The articles selected were published in 2008 (1), 2009 (1) and 2010 (2), 2011 (4), 2013 (1), 2014 (2), 2015 (4), 2016 (8), and 2017 (6). Despite the research considering the past ten years, the striking presence in the last four years show that the discussion of the theme is quite recent. The majority are empirical articles (16) based on practical cases, only 12 cases suggest models without validating and one of them makes a simulation of the model.

## 5.2. Design Thinking integrated with Agile Software Development Models

We found different implementing models of the DT approach integrated with the ASD methodology in which the DT approach proposed by the ISO had the highest incidence of integration with agile methods in ten articles, followed by DSchool and IDEO models, with five and two articles, respectively, as can be noticed in Fig. 1(a). In 12 articles DT approach were unspecified, because the authors refer to the use of a user-centered design approach, without mentioning DT explicitly. The findings reflect the variety of their application in the development information systems. It “provides requirements and recommendations for human-centered design principles and activities throughout the life cycle of computer-based interactive systems” [17].

We observed in this research that Scrum is the ASD methodology most often integrated with the DT approach, being referenced in 11 studies (38%), as can be noticed in Fig. 1(b). In most studies using Scrum as an ASD established a model of integrated application of DT with agile methods as a practical application. The research findings show that the practical case studies using the integrated method is applied during all the software project life cycle, not only in the initial phase.

Disregarding the articles that do not specify the DT approach and the agile method in the integrated model applied in software development, the ISO DT approach appears integrated with Scrum in 5 articles, one of which is integrated with Lean, simultaneously. Two other articles cite the integration of the DT approach of ISO with XP. The DT approach disclosed by Dschool appears in two articles integrated with Scrum, as can be seen in Fig. 1(c).

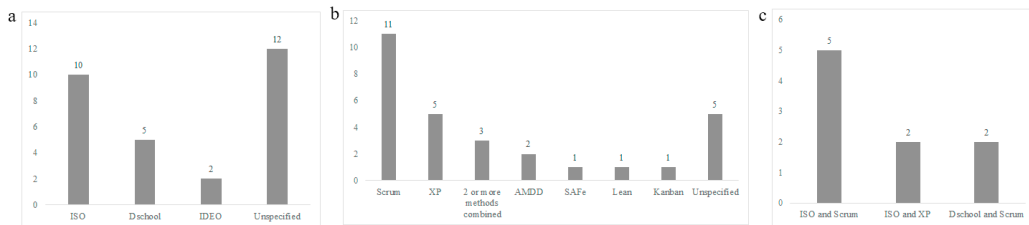


Fig. 1(a) Distribution by DT approach; Fig. 1(b) Distribution by ASD; Fig. 1(c) Frequent integrated models

Although there is variation between the SDLC stages and how the organizations build their software, it has the following stages: communication, requirement gathering, feasibility study, system analysis, software design, coding, testing, integration, implementation, operation and maintenance and disposition [29]. Eight articles explored the DT approach only in the initial stages comprehended in design and requirements phases, indicating as benefits: the design solutions include first ideas of user interfaces [51], the evaluation of different alternatives can be performed in an early stage of development [45,60], and clearer requirements [47] based on customer input [42,52].

On the other hand, most of the articles mentioned the use of the DT approach integrated with ASD throughout the software's life cycle (18 articles), seeking not only to capture the needs of customers in the early stages but ensuring the usability of software, avoiding the implementation of inflexible systems [46]. As noticed in these articles, the model contributes through the whole process [36,40,43], favoring the identification of issues that need to be solved and that reflect customer's satisfaction in the software received [37,48,57] and to achieve project success [64]. Therefore, the use of the DT approach with ASD methods suggests a concern with the better alignment between customer and developer expectations [38,41,59,62] along the entire software life cycle, including better usability [54,55,58] when the criteria for acceptance of the software was defined before the implementation phase [56].

Although in this context there are particularities among the developed software (mobile apps, organizational software, web-based software etc.), several authors exposed the model applied. The users of the developed systems are satisfied and their needs are fulfilled [44] as a result of the products and services better suited to customer needs [47,55,61], the changes are more properly managed when challenges or requirements are discovered [39,49], followed by frequent tests to see how functionality is being implemented [50]. Users also find the software easier to use and require less support to be able to use them [63]. The integrated model composed by the DT approach and ASD has not been associated with a specific software or application type.

## 6. Conclusion

The use of a DT approach promotes communication between software development teams and customers along the entire software development project. We note that in some cases the quality of the software significantly increased and that the satisfaction of the users or customers of the software was perceived. The aspects of the DT approach are characterized by empathy, definition and fast prototyping. This is crucial for a narrowing between the project team and the recipients of the project outcome, in this case, the software developed, checking both non-technical and technical factors. The finding of a limited number of papers that presented rigorous empirical findings can support the recommendation for future studies about the adoption of DT integrated to ASD, especially because the authors only mention the model. Potential future research could be the application of the integrated DT approach with ASD by means of a detailed and prescriptive method.

## References

- [1] Sheppard B, Edson J, Kouyoumjian G. More than a feeling: Ten design practices to deliver business value 2017. <https://www.mckinsey.com/business-functions/mckinsey-design/our-insights/more-than-a-feeling-ten-design-practices-to-deliver-business-value> (accessed March 20, 2018).
- [2] Davenport TH. Reengineering Work through Information Technology. 1993. doi:10.5465/AME.1993.9411302338.
- [3] Owens D, Khazanchi D. From Strategic Intent to Implementation: How Information Technology initiatives take shape in organizations. 51st Hawaii Int. Conf. Syst. Sci., 2018, p. 4783–92.

- [4] Curran C, Garrett D, Puthiyamadham T. A decade of digital - Keeping pace with transformation. 2017.
- [5] Mustonen-Ollila E, Lyytinen K. Why organizations adopt information system process innovations: a longitudinal study using Diffusion of Innovation theory. *Info Syst J* 2003;13:275–97. doi:10.1046/j.1365-2575.2003.00141.x.
- [6] Achi A, Salinesi C, Viscusi G. Innovation capacity and the role of information systems: a qualitative study. *J Manag Anal* 2016;3:333–60. doi:10.1080/23270012.2016.1239228.
- [7] Piccoli, Ives. Review: IT-Dependent Strategic Initiatives and Sustained Competitive Advantage: A Review and Synthesis of the Literature. *MIS Q* 2005;29:747. doi:10.2307/25148708.
- [8] Shenhar AJ, Dvir. Reinventando o gerenciamento de projetos: a abordagem diamante ao crescimento e inovação bem sucedidos. 1st ed. São Paulo: M. Books; 2010.
- [9] Solinski A, Petersen K. Prioritizing agile benefits and limitations in relation to practice usage. vol. 24. Springer US; 2016. doi:10.1007/s11219-014-9253-3.
- [10] Larson D, Chang V. A review and future direction of agile, business intelligence, analytics and data science. *Int J Inf Manage* 2016;36:700–10. doi:10.1016/j.ijinfomgt.2016.04.013.
- [11] VersionOne. 12th Annual State of Agile Report 2018.
- [12] Brown T. Design Thinking. *Harv Bus Rev* 2008.
- [13] Brhel M, Meth H, Maedche A, Werder K. Exploring Principles of User-Centered Agile Software Development: A Literature Review. *Inf Softw Technol* 2015;61:163–81.
- [14] Silva TS da, Martin A, Maurer F, Silveira M. User-Centered Design and Agile Methods : A Systematic Review. *Agil. Conf., IEEE;* 2011, p. 77–86. doi:10.1109/AGILE.2011.24.
- [15] Gruber M, Leon N de, George G, Thompson P. Managing by design. *Acad Manag J* 2015;58:1–7. doi:http://dx.doi.org/10.5465/amj.2015.4001.
- [16] Plattner H. An introduction to design thinking: Process guide. Palo Alto, CA: Stanford Institute of Design; 2009. doi:10.1007/978-1-4302-6182-7\_1.
- [17] DIS I. 9241-210: 2010. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems. International Standardization Organization (ISO). Switzerland: 2010.
- [18] Google. Design sprint kit 2018. <https://designsprintkit.withgoogle.com/methods/> (accessed April 22, 2018).
- [19] Kolko J. Introduction to Design Strategy. *Ac4d - Austin Cent Des* 2018. doi:10.1016/B978-012064155-0/50001-X.
- [20] Design Council. The Design Process: What is the Double Diamond? 2018. <https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond> (accessed April 22, 2018).
- [21] Bogdan C, Gulliksen J, Sauer S, Forbrig P, Winckler M, Johnson C, et al. Human-Centered and Error-Resilient Systems Development. Springer International Publishing; 2016.
- [22] Brown T. Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation. New York: Harper Collins; 2009.
- [23] Martin RL. The design of business - why design thinking is the next competitive advantage. Boston: Harvard Business Press; 2009.
- [24] Keck DW, Cottingham IJ. Using Innovation as a Catalyst for Integrating Business and Engineering Education. *IEEE Front. Educ. Conf.*, 2016.
- [25] Abrahamsson P, Salo O, Ronkainen J, Warsta J. Agile software development methods: Review and analysis. Espoo, Finl Tech Res Cent Finland, VTT Publ 2002:478. doi:10.1076/csed.12.3.167.8613.
- [26] Baseer KK, Reddy ARM, Bindu CS. A Systematic Survey on Waterfall vs. Agile vs. Lean Process Paradigms. *I-Manager's J Softw Eng* 2015;9:34.
- [27] Baweja S, Venugopalan N. Agility in Project Management. *PM World J Agil Proj Manag* 2015;IV:1–14.
- [28] Mishra D, Mishra A. Complex software project development : agile methods adoption. *J Softw Maint Evol* 2011;23:549–64. doi:10.1002/smr.
- [29] Nigam C, Gupta S. Agile Methodology for Software Development. *Inst Innov Technol Manag - J Inf Technol* 2017;3:54–64.
- [30] PMI. Agile Practice Guide. 2017.
- [31] Vallon R, da Silva Estácio BJ, Prikladnicki R, Grechenig T. Systematic literature review on agile practices in global software development. *Inf Softw Technol* 2018;96:161–80. doi:10.1016/j.infsof.2017.12.004.
- [32] Conforto EC, Amaral DC, da Silva SL, Di Felippo A, Kamikawachi DSL. The agility construct on project management theory. *Int J Proj Manag* 2016;34:660–74. doi:10.1016/j.ijproman.2016.01.007.
- [33] Highsmith JA. What Is Agile Software Development? *J Def Softw Eng* 2002;15:4–9. doi:10.1109/MC.2007.73.
- [34] Alzoabi Z. Agile Software: Body of Knowledge. *Bus. Intell. Agil. Methodol. Knowledge-Based Organ. Cross-Disciplinary Appl.*, Hershey PA: Business Science Reference; 2013.
- [35] Petticrew M, Roberts H. Systematic reviews in the social sciences: A practical guide. Oxford, England: Blackwell Publishing; 2006.
- [36] Higuchi MM, Nakano DN. Agile Design: A Combined Model Based on Design Thinking and Agile Methodologies for Digital Games Projects. *Rev Gestão e Proj* 2017;8:109–26. doi:10.5585/10.5585.

- [37] Lucena P, Braz A, Chicoria A, Tizzei L. IBM design thinking software development framework. vol. 680. 2017. doi:10.1007/978-3-319-55907-0\_9.
- [38] Darrin MAG, Devereux WS. The Agile Manifesto, design thinking and systems engineering. 11th Annu. IEEE Int. Syst. Conf., 2017. doi:10.1109/SYSCON.2017.7934765.
- [39] Gamble MT. Can Metamodels Link Development to Design Intent? 1st Int. Work. Bringing Archit. Des. Think. Into Dev. Dly. Act., 2016, p. 14–7.
- [40] Paula DFO De, Araujo CC. Pet Empires: Combining Design Thinking, Lean Startup and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate Environment. Cham: Springer; 2016. doi:10.1007/978-3-319-40548-3.
- [41] Newman P, Ferrarioy MA, Simm W, Forshaw S, Friday A, Whittle J. The Role of Design Thinking and Physical Prototyping in Social Software Engineering. 37th Int. Conf. Softw. Eng., vol. 2, IEEE Press; 2015, p. 487–96.
- [42] Bosch J, Bosch-Sijtsema PM. Introducing agile customer-centered development in a legacy software product line. *Softw - Pract Exp* 2011;41:871–882.
- [43] Forbrig P. When do projects end? - The role of continuous software engineering. vol. 261. 2016. doi:10.1007/978-3-319-45321-7\_8.
- [44] Schön E-M, Winter D, Uhlenbrok J, Escalona MJ, Thomasczewski J. Enterprise Experience into the Integration of Human-Centered Design and Kanban. 11th Int. Jt. Conf. Softw. Technol., 2016, p. 133–40.
- [45] Forbrig P, Saurin M. Supporting the HCI Aspect of Agile Software Development by Tool Support for UI-Pattern Transformations. *Human-Centered Error-Resilient Syst. Dev.*, Springer, Cham; 2016, p. 17–29.
- [46] Fischer H, Senft B. Human-Centered Software Engineering as a Chance to Ensure Software Quality Within the Digitization of Human Workflows. *Human-Centered Error-Resilient Syst. Dev.*, Springer, Cham; 2016, p. 30–41.
- [47] Steinke GH, Al-deen MS, Labrie RC. Innovating Information System Development Methodologies with Design Thinking. 5th Int. Conf. Appl. Innov. IT, 2017, p. 51–5.
- [48] Gurusamy K, Srinivasaraghavan N, Adikari S. An Integrated Framework for Design Thinking and Agile Methods for Digital Transformation. *Int. Conf. Des. User Exp. Usability*, Springer, Cham; 2016, p. 34–42.
- [49] Ardito C, Baldassarre MT, Caivano D, Lanzilotti R. Integrating a SCRUM-Based Process with Human Centred Design: An Experience from an Action Research Study. *Conduct. Empir. Stud. Ind. (CESI)*, 2017 IEEE/ACM 5th Int. Work., IEEE; 2017, p. 2–8. doi:10.1109/CESI.2017.7.
- [50] Sebok A, Walters B, Plott C. Integrating Human-Centered Design and the Agile Development Process for Safety and Mission Critical System Development 2017:1086–90. doi:10.1177/1541931213601876.
- [51] Forbrig P. Continuous software engineering with special emphasis on continuous business-process modeling and human-centered design. *ACM Int. Conf. Proceeding Ser.*, vol. 07–08–Apri, 2016. doi:10.1145/2882879.2882895.
- [52] Ximenes BH, Alves IN, Araújo CC. Software project management combining Agile, Lean startup and design thinking. vol. 9186. 2015. doi:10.1007/978-3-319-20886-2\_34.
- [53] Cano SP, González CS, Collazos CA, Arteaga JM, Zapata S. Agile Software Development Process Applied to the Serious Games Development for Children from 7 to 10 Years Old. *Int J Inf Technol Syst Approach* 2015;8:64–79. doi:10.4018/IJITSA.2015070105.
- [54] González-González CS, Toledo-Delgado P, Muñoz-Cruz V. Agile human centered methodologies to develop educational software | Metodologías ágiles centradas en personas para desarrollar software educativo. *DYNA* 2015;82:187–94. doi:10.15446/dyna.v82n193.53495.
- [55] Isa WARWM, Lokman AM, Aris SRS, Aziz MA, Taslim J, Manaf M, et al. Engineering rural informatics using agile user centered design. 2014 2nd Int Conf Inf Commun Technol ICoICT 2014 2014:367–72. doi:10.1109/ICoICT.2014.6914093.
- [56] Lárusdóttir M, Cajander Å, Gulliksen J. Informal feedback rather than performance measurements - User-centred evaluation in Scrum projects. *Behav Inf Technol* 2014;33:1118–35. doi:10.1080/0144929X.2013.857430.
- [57] Prior S, Waller A, Black R, Kroll T. Use of an agile bridge in the development of assistive technology. *Conf. Hum. Factors Comput. Syst. - Proc.*, 2013, p. 1579–88. doi:10.1145/2470654.2466210.
- [58] Sohaib O, Khan K. Incorporating discount usability in extreme programming. *Int J Softw Eng Its Appl* 2011;5:51–62.
- [59] Lester CY. Combining agile methods and user-centered design to create a unique user experience: An empirical inquiry. *ACHI* 2011 - 4th Int Conf Adv Comput Interact 2011:16–21.
- [60] Losada B, Urretavizcaya M, De Castro IF. An integrated approach to develop interactive software. *Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 2011;6949 LNCS:470–4. doi:10.1007/978-3-642-23768-3\_60.
- [61] Bourimi M, Barth T, Haake JM, Ueberschär B, Kesdogan D. AFFINE for enforcing earlier consideration of NFRs and human factors when building socio-technical systems following agile methodologies. vol. 6409 LNCS. 2010. doi:10.1007/978-3-642-16488-0\_15.
- [62] Xiong Y, Wang A. A new combined method for UCD and software development and case study. 2nd Int Conf Inf Sci Eng 2010:1–4. doi:10.1109/ICISE.2010.5690032.
- [63] Adikari S, McDonald C, Campbell J. Little design up-front: A design science approach to integrating usability into agile requirements engineering. *Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 2009;5610 LNCS:549–58. doi:10.1007/978-3-642-02574-7\_62.
- [64] Hussain Z, Lechner M, Milchrahm H, Shahzad S, Slany W, Umgeher M, et al. Agile user-centered design applied to a mobile multimedia streaming application. vol. 5298 LNCS. 2008. doi:10.1007/978-3-540-89350-9-22.