

# Integrating Design Thinking into Extreme Programming

**Osama Sohaib<sup>1</sup>, Hiralkumari Solanki<sup>1</sup>, Navkiran Dhaliwa<sup>1</sup>, Walayat Hussain<sup>1</sup>,  
Muhammad Asif<sup>2\*</sup>**

<sup>1</sup> School of Systems, Management and Leadership, Faculty of Engineering and IT,  
University of Technology Sydney, Australia  
[e-mail: Osama.Sohaib@uts.edu.au, Walayat.Hussain@uts.edu.au, Hiralkumari.P.Solanki,  
Navkiran.Dhaliwal@student.uts.edu.au]

<sup>2</sup> Department of Computer Science, National Textile University  
Faisalabad, Pakistan  
[e-mail: \* asif@ntu.edu.pk]

## **Abstract**

The increased demand for information systems drives businesses to rethink their customer needs to a greater extent and undertake innovation to compete in the marketplace. The design thinking (DT) is a human-centered methodology leads to creativity and innovation. The agile applications development such as extreme programming (XP) as a rapid application development approach tends to focus on perfecting functionality requirement and technical implementation. However, it causes significant challenges to building software/applications to meet the needs of end-user. This study integrates design thinking (DT) practices into XP methodology to improve the quality of software product for the end-users and enable businesses to achieve creativity and innovation. The proposed integrated DT@XP framework presents the various DT practices (empathy, define, persona, DT user stories) are adapted into XP exploration phase, prototyping and usability evaluation into XP planning phase. Our work demonstrates the applicability of DT concepts to analyze customer/user involvement in XP projects.

**Keywords:** Agile methods. Agile software development. Design thinking. Extreme Programming. Prototyping. User-centered design. Usability.

## 1. Introduction

The problem with agile application methodologies is the lack of real user involvement and ability to allow for innovation. The main purpose of software engineering discipline is to produce applications/products in a cost-effective way with minimum errors and high usability (Gurusamy et al. 2016). In agile software engineering, current approaches to problems tend to focus on perfecting functionality requirements and technical implementation rather than building applications/software to meet the needs of users (Lindberg et al. 2011). The biggest disadvantage of an agile application development methodology such as extreme programming (XP) is that it focuses too much on analytical thinking and from a technical perspective when it comes to finding solutions to a problem (Lindberg et al. 2011). A technically flawless product does not mean it fulfills the user's needs. The end user experience is not the primary aim of agile methodologies.

On the other hand, design thinking (DT) is a human-centered methodology, rapid prototype-based innovation method, which integrates expertise from design, social sciences, engineering, and business (Lindberg et al. 2011). According to the Lindberg et al. (2011), "DT integrates human, business, and technological factors in problem forming, solving, and design." The main aim of DT is to develop a solution in close relation with stakeholders and target users to ensure desirability, practicality, and viability of the final solution.

Large enterprises and software companies are either implementing design thinking or agile methodologies to tackle complex real-world problems. DT and agile methodologies are similar in that they share a common set of principles (Lindberg et al. 2011). The integration of DT and XP is possible because the main advantage of XP is fast development and the primary focus of DT is to emphasize frequent customer feedback from prototypes, which ultimately ensure usability (Gurusamy et al. 2016; Hussain et al. 2009). Agile software approaches are not usually particular about useful techniques for understanding the users' needs (Hirschfeld et al. 2011). Most software projects fail because of lack of design thinking processes (de Paula et al. 2014). However, software projects benefit from DT process to create innovative software products (Wölbling et al. 2012).

Therefore, the purpose of this study is to integrate design thinking (DT) practices into extreme programming (XP) agile development to help improve the process of managing software projects while ensuring usability. Following the above mentioned, the research question is: How do the design thinking (DT) practices integrate with an agile method of extreme programming (XP)? In this study, the integrated approach will combine the strengths of both approaches to improve the quality product for the customers.

The rest of this paper is organized as follows. Background information and literature review about the design thinking are presented in Section 2. Method and finding are explained in Section 3. Integrated DT@XP approach is introduced in Section 4 and discussed in Section 5. Finally, the conclusion is presented in Section 6.

## 2. Background and Related Studies

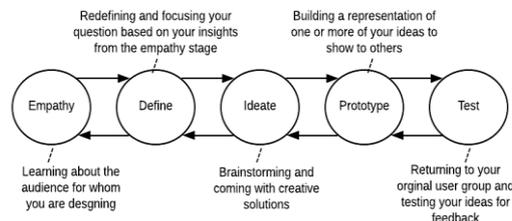
There are similarities between the design thinking (DT) and agile application development methodologies (Lindberg et al. 2011; Hirschfeld et al. 2011). The core features of both approaches are 'user centricity', 'iterative learning and development processes', and 'extensive team communication' (Lindberg et al. 2011). Lindberg et al. (2011) proposed a framework for integrating a core process of DT and agile concepts in

the context of the digital transformation. Adikari et al. (2013) highlighted the implications of DT in agile user experience design. Previous research has addressed the user-centered design approaches in agile software development (de Paula et al. 2014; Sohaib and Khan 2010, 2011; da Silva et al. 2011; 2012; Hussain et al. 2009). However, research on DT and agile software engineering is limited.

## 2.1 Design Thinking (DT)

Design Thinking (DT) is an innovative technique that based on human-centered and rapid prototyping. DT is an approach for creative minds for discovering new opportunities (Adikari et al. 2013; Lindberg et al. 2011; Plattner 2010). DT has gained more popularity in less time because of its conceptual and iterative quality. The main concept of the DT approach is to develop final solution under the guides of stakeholder and target users. DT is a methodology provides a process framework that allows for constant communication between the developing team and the stakeholders and target users. This methodology includes different kind of tool and methods to collect information and data that information related to user needs and creative ideas and also to discover new aspects of the market.

These are the five stages of DT as shown in Fig. 1 (Lindberg et al. 2011; Erickson et al. 2005). The life cycle of DT project starts with empathy, in this phase the target users and their needs are identified. The second phase is called define; here designer needs to define the problem or challenge statement clearly. It is followed by the ideate stage in which the ideas are collected by the team as much as possible. The fourth stage of DT is the prototyping to represent the ideas, and the last stage is testing the ideas.



**Fig. 1.** Design Thinking Process (Lindberg et al. 2011; Erickson et al. 2005)

## 2.2 Extreme Programming (XP)

Extreme Programming (XP) is an agile software development methodology that is a lightweight development methodology and has the capability that responds quickly to changing requirements (Beck 1999, 2000). XP is a continuously evolving methodology since the early 90s (Rittenbruch et al. 2002). According to Beck (1999, 2000), XP has twelve core practices, which are as planning game, small releases, metaphor, simple design, tests, refactoring, pair programming, continuous integration, collective ownership, on-site customer, 40 hours' week, open workspace and just rules. XP principle slogan is "Embrace Change" and the four fundamental values that enable the whole team to be constantly in touch with and responsive to the changing environment are simplicity, communication, feedback and courage (Rittenbruch et al. 2002).

According to Erickson et al. (2005), "Extreme Programming is a discipline of software development based on values of simplicity, communication, feedback, and

courage. It works by bringing the whole team together in the presence of simple practice, with enough feedback to enable the team to see where they are and to tune the practice to their unique situation.” Simplicity in XP is followed in two aspects that are simplicity in process and simplicity in implementation. Simplicity in the process explains that less work for the development team to maintain the XP practices. Simplicity in the implementation is that the software is built merely. It simply means that the software produced is easy to understand with reduced time spent on extension (Beck 1999, 2000; Rittenbruch et al. 2002). Feedback in all level plays a very important role in XP. Constant feedback and evaluation by the user is the major feature of XP (Beck 1999, 2000; Rittenbruch et al. 2002).

In addition, communication between the team members as well as between the members of the team and the customer plays a crucial role in XP. The user also works with the development team and plays an essential part in planning and development phase (Beck 2000). Communication among the team member is also equally important when there is no documentation. It is achieved through frequent planning and design meetings, daily stand-up meetings and sharing knowledge throughout the development tasks. Furthermore, courage in XP simply means to trust the process, neglecting to think about the future possibilities and outcome and concentrating only on developing and implementing the current or immediate requirement (Beck 1999, 2000; Rittenbruch et al. 2002).

### **3. Method**

This study follows the engaged scholarship approach (Van de Ven 2007), which focuses on the participative form of research for co-production of knowledge between the practitioner and the researcher perspectives to understand and theorize about a real-world problem in a social setting. In the context of this research, we focus on the role of the design thinking (DT) and how they are involved in the agile software development process such as extreme programming. According to Rylander (2009), “a better understanding of design thinking as problem solving and how value is created by designers could help in broadening the knowledge economy rhetoric and theories on knowledge work”.

The motivation for this research is to unite design thinking (DT) and extreme programming (XP) to create a new integrated DT@XP approach to work from the generation of concepts to their implementation for solving wicked problems within the context of agile software development.

DT and agile application development are mainly a human activity carried out by team members; to find out this human being behaviour a qualitative approach is required (Coleman and O'Connor 2007; Hoda et al. 2009). Our study is based on an empirical research approach. The empirical data was collected in semi-structured, open-ended interviews with ten participants at two software companies in Sydney. All interviews took place during August-September 2017. The following five open-ended questions were asked of all participants.

- What is your understanding/experience of DT?
- How important is the customer focus in XP versus the end-user focus in DT?
- How do the agile team collaborate and use the designers?
- Does DT fit into agile software development process?
- Do you have any criticism on DT@XP?

The researcher explained the DT process to all participants, although they were well aware of the DT method. The responsibility of the participants included one project manager (P1), four senior software engineers (P2, P3 and P4, P5), two agile coaches (P6 and P7), two business analysts (P8 and P9) and one software tester (P10). All participants were having more than five years of agile software development experience. The questionnaire was intended to inquire the participants understanding of the integrated approach to agile development and DT. The concepts of DT were explained to all participants. The interviews were recorded and then transcribed. For the qualitative data analysis, an NVIVO software tool was used.

In the qualitative study, literature is used inductively (Strauss and Corbin 1990). For that reason, the focus of this study is also on literature and many online references, including the study done already about the subject. The theory from the literary works as a basis for the suggestion for the proposed integrated approach.

### **3.1 Interviews Findings**

All the participants openly express their understanding of design thinking practices in XP throughout the interviews. P2, P3, and P4 explicitly mentioned that integrating design thinking could prove to be feasible for enhancing the quality of end product. P3 mentioned that user requirements are always evolving; so pre-existed personas can be helpful in generating and shaping the user stories. P1 highlighted that it would be a time-consuming process, which we cannot afford in XP while others support the idea of scenario, based testing. Besides, P-1 and P5 highlighted that the nature of DT thereby simplifies requirement gathering communication. For-example, P3 and P5 stated that XP focusses on the close interaction and face-to-face communication between the program team and the customers. Frequent delivery of new software versions is built to adapt to changes in demand for code writing and timely response to customer's requirement.

DT also pay more attention to the role of close interaction in design activities. For example, members of different skills have different views on the project. Therefore, a cross-domain innovation team will be more likely to stimulate more innovation. P6 stated that if the testable prototypes are available after iteration then with the help of some of the user-centered design (UCD) approach to usability and heuristic evaluation can help to find out that the software produced is usable. P7 mentioned it is better to make changes in early stages of development before it is too late and end up in unusable software. P8 and P10 highlighted that one of the challenges in agile methods such as XP is the lack of usable project, one solution would be to apply DT practices. Working with users has been recognized as a key factor to the integration of DT@XP. For example, P7 and P9 mentioned that the key aspect of DT is a user-centric design method, which allows users to fully express their views in the prototyping process, the designer can record the observations to make improvements. In addition, P8 and P10 mentioned that the integration of DT and XP is possible because both approaches emphasize on frequent customer feedback from prototypes.

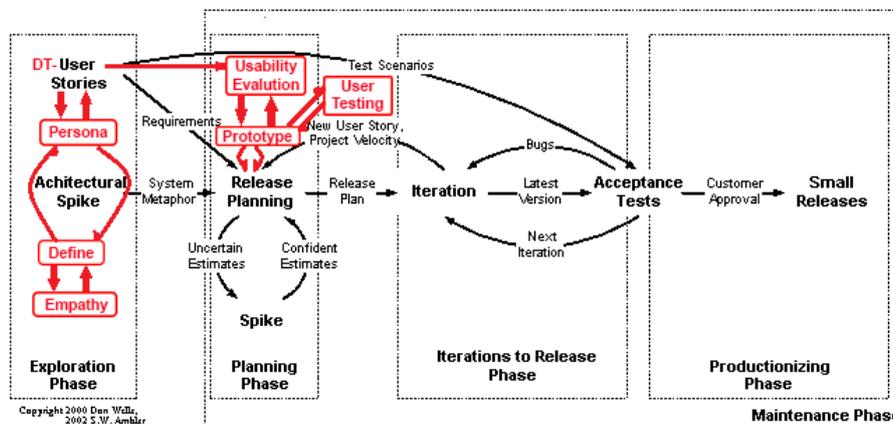
The findings from the interviews show that design thinking is a useful tool if applied in agile methodology. Design thinking practices can enhance the quality of the product design and user experience through iterative software development approaches. Table 1 shows the summary of the main themes emerged from the interviews. The following section explains the integrated DT@XP approach in detail. The interviews findings provide the basis for suggestions for the DT@XP approach.

**Table 1:** Summary of DT and XP practices

DT practices	XP practices	DT@XP
User profiling	User stories	Integrate user stories with persona-based design
Collaborative ideation	Spike solution	Multidisciplinary teams for collaboration and creativity
Prototyping	Incremental design	Prototype development
User feedback	Acceptance test	User-centered design + user acceptance testing
Usability	Working software	Agile usability testing throughout the development process

#### 4. Integrated DT@XP Approach

Since XP does not consider DT practices, the proposed framework integrates DT and XP to compliment the problem-solving abilities of software development teams with the purpose of making their product more users friendly and innovative. Adapting the DT concepts into XP project lifecycle opens enhancement and can be applied in XP iterations (Amber 2016; Wells 2009). In the proposed framework, the various DT practices are adapted into two XP phases called exploration and planning phase. The adapted approach of XP and DT is shown in Fig. 2.



**Fig. 2.** Integrated framework of DT@XP

##### 4.1 DT Integration in XP Exploration Phase

Exploration is the initial phase of XP. In this phase, user stories are created and release for the planning phase. User stories generation is an ongoing process in the XP iterative approach. Therefore, involving the customer in user story creation and gathering also ease the chances of incoming change requests (Inayat et al. 2015). To make a realistic customer expectation from the output of each iteration, we recommend including ‘empathy’ and ‘define’ practices of DT along with user ‘persona’ to create a DT-user story. The

following sections explain DT practices (empathy, define and persona) in the exploration phase of the XP lifecycle.

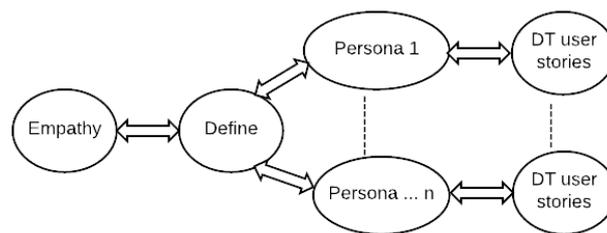
**Empathy.** Empathy deals with the human-centered design process to understand people, their emotional and their needs by observing, engaging, watching and listening user so that you can identify their viewpoint. According to Adikari et al. (2013), “Empathy is one of the most important phases in design thinking to reflect emotional aspects and experience of all the users in context”.

**Define.** ‘Define’ is a process of synthesizing information to discover connection and pattern (Lindberg et al. 2011). The define process aims to explore the problem space and create a meaningful and actionable problem statement after the ‘empathy’. This problem statement is known as a point-of-view which serve as guidelines for the team to focus on understandings of a user needs and it will drive your whole design by an actionable problem statement. It narrows down the area of discussion and requirement gathering. A good point of view can be helpful for 1) delivering attentions and provide a framework to the problem. 2) encourages the group members. 3) provides clear information guidelines for assessing competing ideas. 4) Inspires team members to make choices individually while working parallel.

**User Persona.** The use of persona is a strong communication tool for team and inspires them about user needs (Siricharoen 2012). According to Broschinsky and Baker (2008), XP team lacks in communication factor with their customer and this can successfully solve by implementing personas in XP. In the proposed framework personas are evaluated through ‘empathy’ and ‘define’ practices output that concentrates on customers and their needs.

#### **Design Thinking –User Stories (DT-User Stories).**

XP includes user stories based on customer demands, so adopted DT-user stories are built by personas and are kept on evolving along with ‘Empathy-Define-Personas’ cycle approach to generate user feedback for developing a workable output.



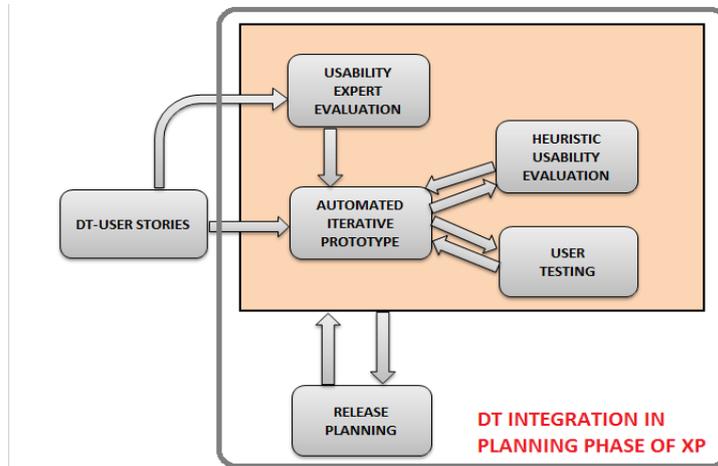
**Fig. 3.** Integration of DT in Exploration

As shown in the Fig. 3, the DT-user stories are created for each persona generated and mapped to user requirements. The DT-user stories will be used as the output of the exploration phase and the input for the planning phase.

#### **4.2 DT Integration in XP Planning Phase**

In the planning phase, future development of the application/software/product is decided. The planning begins with sorting the user stories and end with release plan. The process can also be termed as planning game which is a collaborative planning process for the

team members. As shown in the Fig. 4, the usability evaluation is performed on DT-user stories and prototypes. Finally, real users will test the prototype so that the feedback from the users can be collected and changes can be made accordingly. The final prototype generated for user stories will be passed to the release planning phase.



**Fig. 4:** DT integration in the planning phase

### Automated Prototyping

A prototype is one of the ideation techniques which explore solution space in design thinking process (Lindberg et al. 2011; Gabrysiak et al. 2011). Integrating prototype in XP can enhance software product quality. According to Silva et al. (2015), the usability evaluations are mainly performed by experts and that should be performed on an ongoing basis from the start in three phases (early, mid-release and late) of design stage (Silva et al. 2015). Within the agile software development, paper prototypes and low-fidelity prototypes are used to validate business goals with the UX designer and business analysts. However, the hi-fidelity prototypes are used to validate the target users (Silva et al. 2015). Prototypes are effective in an iterative approach to the design and evaluation of the system (Rittenbruch et al. 2002). Thus, developer and designer can get user feedback in the early stage of development and hence the iterative software produced will be usable (Soutome et al. 2013).

However, Kamimori et al. (2015) propose a method of generating an automated web prototype. The proposed web prototype employs live interactive widgets (LIWs) so that users can validate the models automatically. The models are developed in UML-based notation so that developers can apply the prototypes to their existing coding development process easily. Kamimori et al. (2015) define a “LIW as a Web UI component that immediately responds to the inputs of a user”. We propose adopting the web automated prototype (Kamimori et al. 2015) in the XP planning phase would help developers to obtain the artefacts efficiently for validating functional usability requirements.

**Prototype Usability Evaluation.** Usability can be defined as the extent to which a product produced can be proved to achieve the intended goals of the user with effectiveness and efficiency (Jokela and Abrahamsson 2004; Holzinger et al. 2005). Even though usability evaluation is performed at the early stage of software development, still usability problems are found in the later stage because of the use of traditional usability testing techniques which are not suitable for the growing complexities of the software projects (Sivaji et al. 2013). Thus, we propose heuristic usability evaluations to evaluate

usability at the early stage of the agile. Heuristic evaluations at the early stage of development can be beneficial because improvement can be made as part of the iterative design process (Sohaib and Khan 2011). The usability heuristic evaluation method is the combination of exploratory heuristic evaluation and lab-based usability testing (Sivaji et al. 2013).

**User Testing.** Testing is the last part of design thinking cycle with the condition that the cycle is iterative (Newton and Riggs 2016). The user will test the prototype and provide feedback which can be valuable in the early stage of development. The mapping of the prototype with user stories and personas should also be performed to ensure that the user requirements are accomplished. Prototype testing provides certain benefits. 1) feedback from users. 2) validate user requirement and helpful to understand real user requirement. 3) scenario-based testing provides exact task achievement. 4) acceptance testing time can be reduced. 5) and to ensures that the requirements of the target user group are meeting.

## 5. Discussion

Integrating DT into the XP can contribute to delivering constant continuous deliveries to the customers faster than the traditional approach. Today's businesses are driving processes, which will lead to market success without considering the needs of the customer while designers are constantly looking for innovative ideas to break through the market. This research will help software product development team to foster creativity in agile development methodologies.

DT and XP has so many similarities such as user-centricity, iterative development processes, and extensive team communication (Lindberg et al. 2011). XP concentrate on continuous incremental refinement of the whole process whereas DT explores radically new solution path. It also avoids divergent thinking and limits down to trial and error approach of prototyping. Participatory design is one of the important issues both in DT and agile software development (Kautz 2010).

Even though there is a strong emphasis on team collaboration in XP, but the people involved are trained thinker and the real thinking style is not implemented. One effort to resolve such problem is by introducing specialist role like user-interface designer mainly work on user-friendly digital graphics interfaces (Mandel 1997). Interaction designer who takes care of the dynamic aspects of human-computer interaction, user-centered designer generates and validate software systems design decision and user-experience design the whole experience (Lindberg et al. 2011).

Prototyping can be a useful tool in design thinking because they are considered as mock-up which supports elaboration and evaluation of product concept to find out the right or wrong way. Prototyping is often encouraged as realization rather than iteration when there is a problem of time constraint. Researchers suggest that several rapid prototypes yield valuable outcomes than allocating that time to a single iteration (Lindberg et al. 2011). An iterative prototype helps designer refine their idea and discover issues and opportunity (Dow and Klemmer 2011). Usability testing of the prototype with the real users can identify the lots of usability problems in the early stages (Hussain et al. 2009). Prototypes can also be validated through mapping scenarios that are obtained during the design thinking process (Gabrysiak et al. 2011).

Adikari et al. (2013) proposed a framework to enhance the user experience design by integrating three design approaches (design thinking, designing for user experience and

agile software development), and this has resulted enhancing the quality of design and user experience through agile software development approaches. According to (Wölbling et al. 2012), software industry can benefit from DT approach in order to create innovative software products. Hassi and Lakso (2011, 2011) proposed framework identifies elements of DT, which are characterized in three dimensions: practices, thinking styles, and mindsets. Carlgren et al. (2016) illustrate how these DT elements interact with each other.

## 6. Conclusion and Future work

In conclusion, we presented an integrated framework based on design thinking (DT) and extreme programming (XP) concepts. The main contribution of this paper is highlighted in the new approach in the context of agile software engineering and complement and enhances the software product development approaches. This study contributes to the knowledge of design thinking research and agile software development.

DT is a comparatively new approach to software development and has yet to be extensively studied. This integrated framework was developed based on qualitative analysis and literature on design thinking and agile software methodologies (Extreme Programming). Attention was given to previous research and findings were considered while developing the proposed framework. However, the proposed integrated approach needs to be evaluated in real software projects. In addition, future work includes an extension of our proposed DT@XP approach. Furthermore, DT has yet to be extensively studied with the agile software development such as Scrum. There is a significant room for research about integrating DT and agile methodology such as XP and Scrum.

## References

- Adikari S, McDonald C, Campbell J (2013) Reframed contexts: design thinking for agile user experience design. In: Marcus A. (eds) *Design, User Experience, and Usability. Design Philosophy, Methods, and Tools*. DUXU 2013. Lecture Notes in Computer Science, vol 8012. Springer, Berlin, Heidelberg. pp. 3-12. [https://doi.org/10.1007/978-3-642-39229-0\\_1](https://doi.org/10.1007/978-3-642-39229-0_1)
- Amber S. W (2018) AM throughout the XP lifecycle. Ambysoft Inc. <http://www.agilemodeling.com/essays/agileModelingXPLifecycle.htm>. Accessed 02 Nov 2017
- Beck K (1999) Embracing change with extreme programming, *Computer*, vol. 32, no. 10, pp. 70-77. <https://10.1109/2.796139>
- Beck K (2000) *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Boston
- de Paula D.F.O, Menezes B.H. X. M, Araújo C. C (2014) Building a Quality Mobile Application: A User-Centered Study Focusing on Design Thinking, User Experience and Usability. *International Conference of Design, User Experience, and Usability*, Springer International Publishing, pp 313-322
- Broschinsky D, Baker L (2008) Using persona with XP at LANDesk Software, an Avocent company. In: *Proceeding of the Agile 2008 conference (AGILE'08)*. pp. 543-548. IEEE. <https://doi.org/10.1109/Agile.2008.91>
- Carlgren L, Rauth I, Elmquist M (2016) Framing Design Thinking: The Concept in Idea and Enactment. *Creativity and Innovation Management*, 25(1). pp 38–57. <http://doi.org/10.1111/caim.12153>
- Coleman G, O'Connor R (2007) Using grounded theory to understand software process improvement: A study of Irish software product companies. *Information and Software Technology*, vol. 49, Issue 6, pp. 654-667. <https://doi.org/10.1016/j.infsof.2007.02.011>

- Dow S.P, Klemmer S.R (2011) The efficacy of prototyping under time constraints. In: Meinel C., Leifer L., Plattner H. (eds) *Design Thinking. Understanding Innovation*. Springer, Berlin, Heidelberg, pp 197–218. Springer. pp. 111-28.
- da Silva S. T, Silveira S M, Maurer F, Hellmann T (2012) User Experience Design and Agile Development: From Theory to Practice. *Journal of Software Engineering and Applications*. Vol. 5 No. 10. Pp 743-751. <http://dx.doi.org/10.4236/jsea.2012.510087>
- da Silva S.T, Silveria, M.S, Maurer F, Hellmann T (2011) User-Centered Design and Agile Methods: A Systematic Review. In: *Proceeding of the 2011 Agile Conference*. Salt Lake City, UT. pp. 77–86. <https://doi.org/10.1109/AGILE.2011.24>
- Erickson J, Lyytinen K, Siau K (2005) Agile modeling, agile software development, and extreme programming: the state of research. *Journal of database management*, vol. 16, no. 4, p. 88-100. <https://10.4018/jdm.2005100105>
- Gabrysiak G, Giese H, Seibel A (2011) Towards Next Generation Design Thinking: Scenario-Based Prototyping for Designing Complex Software Systems with Multiple Users. In: Meinel C., Leifer L., Plattner H. (eds) *Design Thinking. Understanding Innovation*. Springer, Berlin, Heidelberg. pp 219-236. [https://doi.org/10.1007/978-3-642-13757-0\\_13](https://doi.org/10.1007/978-3-642-13757-0_13)
- Gurusamy K, Srinivasaraghavan N, Adikari S (2016) An Integrated Framework for Design Thinking and Agile Methods for Digital Transformation. In *Proceedings of the 5th International Conference on Design, User Experience, and Usability: Design Thinking and Methods*., DUXU 2016, Held as Part of HCI International 2016, Toronto, Canada, July 17–22, 2016, Proceedings, Part I, A. Marcus, Ed., Cham. 34-42. Springer, [https://doi.org/10.1007/978-3-319-40409-7\\_4](https://doi.org/10.1007/978-3-319-40409-7_4)
- Hassi L, Laakso M (2011) Making sense of design thinking. In T. M. Karjalainen, Koria, M. & Salimäki, M. (Ed.), *Helsinki: International Design Business Management Program, Aalto University. IDBM papers, vol 1*, pp. 50-62
- Hassi L, Laakso M (2011) Conceptions of Design Thinking in the Design and Management Discourse. In: *Proceedings of I European Academy of Design Conference, Porto, Portugal*, pp 341-351
- Hirschfeld R, Steinert B, Lincke J (2011) Agile software development in virtual collaboration environments. In: Meinel C., Leifer L., Plattner H. (eds) *Design Thinking. Understanding Innovation*. Springer, Berlin, Heidelberg. pp 197–218. Springer. [https://doi.org/10.1007/978-3-642-13757-0\\_12](https://doi.org/10.1007/978-3-642-13757-0_12)
- Hoda R, Noble J, Marshall S (2009) Negotiating contracts for agile projects: A practical perspective. In: Abrahamsson P., Marchesi M., Maurer F. (eds) *Agile Processes in Software Engineering and Extreme Programming. XP 2009. Lecture Notes in Business Information Processing*, vol 31. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-01853-4\\_25](https://doi.org/10.1007/978-3-642-01853-4_25)
- Holzinger A, Errath M, Searle G, Thurnher B., Slany W (2005) From extreme programming and usability engineering to extreme usability in software engineering education (XP+ UE→ XU). *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, vol. 2, pp. 169-72. IEEE <https://doi.org/10.1109/COMPSAC.2005.80>
- Hussain Z, Milchrahm H, Shahzad S, Slany W, Tscheligi M, Wolkerstorfer P (2009) Integration of extreme programming and user-centered design: Lessons learned. In: *Proceeding of the International Conference on Agile Processes and Extreme Programming in Software Engineering*. pp 174-179. Springer, [https://doi.org/10.1007/978-3-642-01853-4\\_23](https://doi.org/10.1007/978-3-642-01853-4_23)
- Hussain Z, Slany W, Holzinger A (2009) Current state of agile user-centered design: A survey. In: Holzinger A., Miesenberger K. (eds) *HCI and Usability for e-Inclusion. USAB 2009. Lecture Notes in Computer Science*, vol 5889. Springer, Heidelberg. pp. 416–427. [https://doi.org/10.1007/978-3-642-10308-7\\_30](https://doi.org/10.1007/978-3-642-10308-7_30)
- Inayat I, Moraes L, Daneva M, Salim S.S (2015) A reflection on agile requirements engineering: solutions brought and challenges posed. In: *Proceeding of the XP2015, Helsinki, Finland*. <https://doi.org/10.1145/2764979.2764985>
- Jokela T, Abrahamsson, P (2004) Usability assessment of an extreme programming project: Close co-operation with the customer does not equal to good usability. In: Bomarius F., Iida H. (eds) *Product*

- Focused Software Process Improvement. PROFES 2004. Lecture Notes in Computer Science, vol 3009. Springer, Berlin, Heidelberg. pp 393-407. [https://doi.org/10.1007/978-3-540-24659-6\\_28](https://doi.org/10.1007/978-3-540-24659-6_28)
- Kamimori S, Ogata S, Kaijiri K (2015) Automatic Method of Generating a Web Prototype Employing Live Interactive Widget to Validate Functional Usability Requirements. In: Proceeding of the 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI). pp. 8-13. IEEE. <https://doi.org/10.1109/ACIT-CSI.2015.11>
- Kautz K (2010) Participatory Design Activities and Agile Software Development. In: Pries-Heje J., Venable J., Bunker D., Russo N.L., DeGross J.I. (eds) Human Benefit through the Diffusion of Information Systems Design Science Research. TDIT 2010. IFIP Advances in Information and Communication Technology, vol 318. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-12113-5\\_18](https://doi.org/10.1007/978-3-642-12113-5_18)
- Lindberg T, Meinel C, Wagner R (2011) Design thinking: A fruitful concept for its development? In: Meinel C., Leifer L., Plattner H. (eds) Design Thinking. Understanding Innovation. Springer, Berlin, Heidelberg, 3-18. Springer, [https://doi.org/10.1007/978-3-642-13757-0\\_1](https://doi.org/10.1007/978-3-642-13757-0_1)
- Mandel T (1997) The elements of user interface design. Wiley New York
- Newton K, Riggs M.J (2016) Everybody's talking but who's listening? Hearing the user's voice above the noise, with content strategy and design thinking. VALA2016: libraries, technology and the future, VALA, Australia, pp. 1-16.
- Plattner, H (2010) An Introduction to Design Thinking Process Guide. The Institute of Design at Stanford. <https://dschool-old.stanford.edu/sandbox/groups/designresources/wiki/36873/attachments/74b3d/ModeGuideBOOTCAMP2010L.pdf> Accessed 22 Jan 2018
- Rittenbruch M, McEwan G, Ward N, Mansfield T, Bartenstein D (2002) Extreme participation - Moving extreme programming towards participatory design. In: Proceeding of the participatory design conference, (PDC2002). pp 29-41
- Rylander A (2009) Design Thinking as Knowledge Work: Epistemological Foundations and Practical Implications. Design Management Journal. 4: 7-19. <https://doi.org/10.1111/j.1942-5074.2009.00003.x>
- Siricharoen W.V (2012) User persona roles in the end-user web developing approach. In: Proceeding of the 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), pp. 415-420. IEEE. <https://doi.org/10.1109/DICTAP.2012.6215394>
- Silva T. S. d, Silveira M. S, Maurer F (2015) Usability Evaluation Practices within Agile Development. In: Proceeding of 48th Hawaii International Conference on System Sciences, Kauai. pp. 5133-5142. <https://doi.org/10.1109/HICSS.2015.607>
- Sivaji A, Abdullah M.R, Downe A.G, Ahmad W.F.W (2013) Hybrid usability methodology: integrating heuristic evaluation with laboratory testing across the software development lifecycle. In Proceeding of the 10th International Conference on Information Technology: New Generations (ITNG), pp. 375-83. IEEE. <https://doi.org/10.1109/ITNG.2013.60>
- Sohaib O, Khan K (2011) Incorporating discount usability in extreme programming. International Journal on Software Engineering and its Applications, vol. 5, no. 1, pp. 51-61.
- Sohaib O, Khan K (2010) Integrating usability engineering and agile software development: A literature review. In: Proceeding of the 2010 International Conference On Computer Design and Applications. pp. 32-38. IEEE. <https://doi.org/10.1109/ICCD.2010.5540916>
- Soutome T, Ling D.K, Niibori M, Kamada M (2013) A Web-Based Platform for Clients and Designers to Prototype Web Sites. In: Proceeding of the 16th International Conference on Network-Based Information Systems. pp. 459-63, IEEE. <http://doi.ieeecomputersociety.org/10.1109/NBiS.2013.76>
- Strauss A.L, Corbin L (1990) Basics of Qualitative Research: Grounded Theory Procedures and Techniques. Thousand Oaks, CA, US: Sage Publications, Inc.

Van de Ven A.H (2007) *Engaged Scholarship: A Guide for Organizational and Social Research*. Oxford University Press, New York

Wells D (2009) *Extreme programming: a gentle introduction*. <http://www.extremeprogramming.org>. Accessed 05 Nov 2017

Wölbling A, Krämer K, Buss C.N, Dribbisch K, LoBue P, Taherivand A (2012) *Design Thinking: An Innovative Concept for Developing User-Centered Software*. In: Maedche A., Botzenhardt A., Neer L. (eds) *Software for People. Management for Professionals*. Springer, Berlin, Heidelberg. pp. 121-136. [https://doi.org/10.1007/978-3-642-31371-4\\_7](https://doi.org/10.1007/978-3-642-31371-4_7)