Design Thinking versus Design Sprint: A Comparative Study

Carlos Magno Mendonça de Sá Araújo, Ivon Miranda Santos, Edna Dias Canedo[0000−0002−2159−339X], and Aleteia Patricia Favacho de Araújo

Computer Science Department - Professional Masters in Applied Computing - University of Brasília (UnB) – 70910-900 – Brasília, DF, Brazil
carlos.magno;ivon.miranda@gmail.com,ednacanedo@unb.br,aleteia@unb.br
http://ppca.unb.br

Abstract. Design Thinking methodology is being increasingly used by the software industry in an attempt to reduce the problems with requirements elicitation. In the literature it is possible to find several papers addressing the use of this methodology in the analysis and specification of software requirements. In order to help reduce existing problems that have not yet been fully resolved through the use of Design Thinking, Google has launched Design Sprint with the goal of eliciting requirements quickly and efficiently. There are very few academic papers reporting on the use of this technique and what its advantages and disadvantages are. This paper describes a comparative study of the two methodologies used to minimize the problems faced by companies in eliciting the requirements of their software, presenting their phases and how they can be used.

Keywords: Design Sprint, Design Thinking, Methodologies, Requirements Elicitation, Comparative Study.

1 Introduction

Requirements engineering is the science that studies and defines which tasks a software should perform and how [14]. In this process there are several methodologies, the oldest of which is the traditional methodology which become, very widespread due to RUP - Rational Unified Process, a methodology created by Rational [13]. However, this methodology employs a method in which all requirements are raised at the beginning of the project, and problems in the requirements specification will only be identified late in the software development phase, raising the cost of corrections. Encouraged by these problems the studies point out that not only should software development be agile, but so should the gathering of requirements. For this reason, several agile methodologies and processes are being developed to determine software requirements.

The agile methodology emerged from the techniques used by innovative Japanese companies in the 1970s and 1980s (become popular such as Toyota, Fuji and Honda) [12]. These agile methods have in project management because of their high success rates.

The most widespread methodology currently is SCRUM, which works with Sprints to develop the project as a whole [15]. There are however methodologies that complement and even replace SCRUM. We need to think about product development to make the project a success. This paper presents a comparative study between two of the most popular agile requirements analysis techniques:

- **Design Thinking**: a methodology that contrasts with the traditional scientific methodology where it relies on the collaboration of a multidisciplinary team to solve complex problems through the application of design knowledge [16].
- **Design Sprint**: a unique five-day Google Venture process used to solve critical issues through prototyping and brainstorming with customers [4], [3]. All the best ideas are condensed and organized in a short space of time for the creation of an excellent idea. In this process, a step-by-step description of what is to be done in each of these five days is given in detail. That, at the end of these five days, we have a validated product that we believe or we are determine that the project, in the way it was designed, is not ideal [11].

This comparative study allowed us to verify that the two methodologies are being widely used, both in the software industry and in case studies developed by the academy to validate its use. Therefore, with this work it was possible to identify that they are both appropriate approaches when using an agile development methodology. Being appropriate and providing benefits to the requirements elicitation adheres to the needs of stakeholders.

This paper is organized as follows. Section 2 contains the concepts needed for around the Design Thinking and Design Sprint methodologies. Section 3 presents a comparative analysis of the two methodologies and Section 4 presents the conclusions and the future works.

2 Background

Software methodologies have been developed to facilitate, as far as possible, the software development process. Few methodologies were known until recently. The traditional methodology is one of them, iterative development another among a few others. Nowadays, many software methodologies have emerged that are more flexible than existing ones and this is one reason they are called agile [2]. These methodologies, unlike traditional ones, bring the end-user closer to the software development team, whether it is an internal or external customer [9]. With this type of methodology, approvals are often made in short periods of time, usually 2 to 4 weeks, resulting in several shorter deliveries until the final product is reached [10], [8].

The software is gradually being delivered and it is possible to make occasional changes during its construction. Among these emerging methodologies are Sprint Design [4] and Design Thinking [16].

2.1 Design Thinking

Tim Brown [16] believes that the idea that drives Design Thinking is a collaborative process that uses sensitively and creative techniques to meet the needs of people not only with what is technically visible, but with a viable business strategy. Design Thinking aims to convert a need into demand. It is a human-centric approach to problem solving and helping people and organizations to be more innovative and creative.

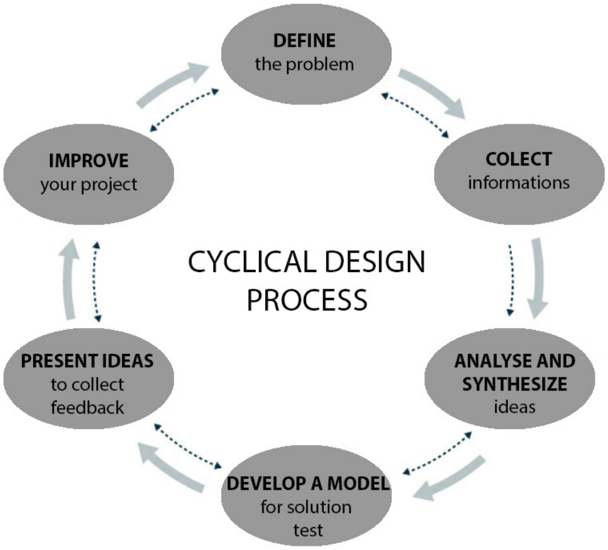

Fig. 1. Design Thinking Process, adapted [17]

Although the term design usually refers to something visual, the concept can mean much more than that. Design means planning or making decisions about something that is being created, as well as planning, making, or executing a plan to produce something for a specific purpose [7]. The concept of design is much closer to a project than to a visual design. Design brings a more human-like perception to problem-solving, caring about individuals before tools to solve a problem [6]. New ways of thinking about a problem are established from Design Thinking, seeking to understand a number of contextual factors in thinking about a solution:

1. Who are the individuals we are serving?
2. What are their needs and desires?
3. How do they live? 4. What are their personal experiences?

The design makes it clear that it is an approach focused on **people**. By following a design process, we go through procedures that depart from the definition of the scope of a problem, collecting information that can help solve this problem, experimenting with and testing different solutions and refining them through feedback until achieving a effective solution that addresses the people affected by the problem, as shown in Figure 1. Design Thinking is based on three pillars focused on people and their needs, enabling human factors to be in focus as protagonists throughout the process [5] namely:

- **Empathy** - the ability to put yourself in another's place, understanding their context and their point of view, including their needs, fears and longings.
- **Collaboration** - integration of multidisciplinary and multifunctional teams. It promotes a richer development process through diversity of experience and specialities.
- **Experimentation** - offering the solutions to be evaluated in the real world, and learning from feedback, refining the product or service.

The Design Thinking approach aims to answer the following questions[6]:

1. **Who** are we solving this problem for?
2. **What** are the needs of these people?
3. **How** will this problem be resolved?
4. **Why** does this job or solution matter to people?

According to [6], these questions are structured into a model. Within this model design thinking starts at the discovery phase that generates options which, after choices are made, converge on what was observed to best the resolve the problem, and then diverge again into a range of development options to build the solution. There are two distinct moments of analysis of the process: during the definition of the problem and during the development of solutions.

Design Thinking is an iterative methodology that brings a new approach to solving complex problems by applying Design knowledge. It has five distinct and well ordered steps so that the predecessor is an input to the successor. The evolution of these steps up until the conclusion of each iteration cycle in this process are [1], [16], [18]:

1. **Discovery**: immersion (empathy or exploration). At this stage the team should immerse themselve in the problem to get to know it better, using several techniques, to do at complete mapping of the problem. The immersion can be performed in a Preliminary stage and another one in Depth [6]. Reframing, Exploratory Research and Desk Research are part of the preliminary immersion, having the objective of reframing and developing an initial understanding of the problem. The immersion in Depth is intended to identify the needs and opportunities that will guide the generation of solutions for the next phase of the project, to define the scope of the project and its boundaries, as well as to identify the user profiles and other key actors that should be addressed [6].
2. **Interpretation**: analysis and synthesis (definition of the problem). After the immersion stage it is necessary to structure and group the collected data to begin the analysis and synthesis phase. Some techniques are used,

for example: affinity diagram, insight cards, mental maps, conceptual maps, personas and proto-personas and the user's journey. The objective of the analysis and synthesis stage is to organize the collected information and to obtain a synthesis of the desires, needs and motivations of the interested parties, broadening the understanding of the problem that the team intends to solve [6]. At this stage the project team should focus on analyzing all the data collected in the dive immersion stage, and produce a number of ideas. In addition to the wealth of information, team experience and creativity are essential elements for having good **insights**. Another key factor is to form a multidisciplinary team, the more diversified, the greater the wealth of ideas generated [16].

3. **Ideation**: In the ideation stage the team proposes several solutions, using techniques such as *brainstorming* and co-creation workshops. Once all the contributions have been made the group should analyze them together to refine them and decide which will be used in the process [6]. At the time of the synthesis the solutions will be analyzed deeply and many can be eliminated for lack of viability; others are disregarded because better exist; these better will be refined and perhaps even merged with other equally good ones, improving them further [16]. This phase aims to generate innovative ideas for the project theme, stimulating creativity to generate solutions that are in accordance with the context of the subject worked. It is good practice to include in the team those people who will make use of the solution after the final implementation [16]. Thus, in addition to the multidisciplinary project team, other members are selected as users and professionals of areas that are convenient to the topic under study, usually through Workshops co-creation. The objective of bringing together different expertise is to contribute different perspectives, thereby making the final result richer and more assertive [6].
4. **Experimentation**: the prototypes begging with all the materials that were created in the previous steps: personas, user's journey, ideas generated in brainstorming, etc. Ideas are converted into a more concrete deliverable for evaluation. Prototyping has the function of assisting the validation of generated ideas and although presented as one of the last phases of the Design Thinking process, can occur throughout the project in parallel with Immersion and Ideation [6]. Prototypes are classified into high and low fidelity, also called functional prototypes. In this step, a prototype will be constructed that can provide a tangible or intangible experience of the tool through the implementation of a presentation or a prototype [16]. You can use tools such as Storyboard, Video, Innovation Fair, Storytelling, Role play, Magic of Oz (Wizard of Oz Prototyping), among others [16]. Similary, design thinking teams should regard the critiques of their solutions positively and constructively. They should not take personally people's negative comments about their proposed solutions. Instead, they should remember the adage, "*The customer is not always right but always has a point*" [6].
5. **Evolution**: the prototypes should be evaluated in a real-use scenario for field comparison and validation [6]. In the evaluation phase, Design Think-

ing teams test their prototype solution with the users representing target **personas**. They then update the solution in an iterative manner until the solution meets the needs of the user and exceeds the challenge set in the initial design phase [6]. Design Thinking team members should always thank users for their criticisms of solutions. Criticism is a natural part of many team efforts, including design and artistic aspects. In history, criticism has even contributed to the emergence of some artistic movements. These artists did not give up because of criticism; instead, they used them to improve their artistic styles and perspectives. Likewise, Design Thinking teams should receive the criticism of their solutions positively and constructively. They should not take people's negative comments about their proposed solutions personally. Instead, they should remember the saying: " **The client is not always right, but always has a point**" [6].

After the evaluation it is possible to use the results obtained to start a new cycle from the immersion stage, making the process repeatable until the desired level of refinement is reached.

2.2 Design Sprint

A Design Sprint is a unique, five-day process created by Google Venture [4] to solve critical issues through prototypes and brainstorming with customers. The methodology and its application include the following activities:

1. **It's time to prepare the environment for everything that will happen**: Before we start a Sprint, we need the team the defined challenge set, and space and time for a Sprint to take place. As previously mentioned, you need a well-defined challenge before you start the 5- day Sprint's 5-day in order to generate a prototype that will lead to a successful product. Besides the goal, it is necessary to choose a team, perhaps a maximum of 7 people and a comfortable space for everyone to make the game happen. Figure 2 presents the phases of Sprint Design.

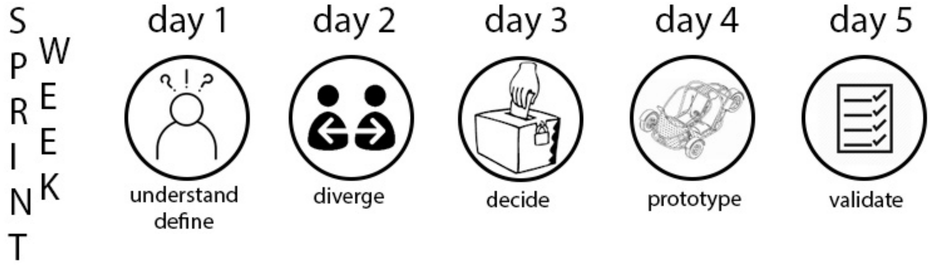

Fig. 2. Phases do Design Sprint [4]

2. **The Challenge**: Sprints can be useful in challenging situations, such as high-risk projects. A Sprint is a good time to check out initial ideas and change direction in a short time frame. It also fits in well when you are short of time to test the outcome of a project. In five days you will need to find good solutions quickly. One tip the experience with Sprints has given us: Do not get hung up on details, worry about the bigger picture. It is what will be visible to and attract customers. A poorly-made prototype or product has a good chance of failure. After worrying about the and completing its prototype we will worry about the internals. Now that we know how to choose a good challenge, it's time to choose one. Once we have chosen the challenge, we will have to choose the team. We know that it is not easy to select a team [4].
3. **The Team**: For a Sprint to have a chance of success, it is advisable to set up a team of 7 people or less, preferably including people with the following roles in the company: Definer; Finance Specialist; Marketing Specialist; Consumer Specialist; Technology Specialist; Design Specialist. When choosing the team it is important that people with good relationships are included, because they will work together full time for 5 days. Despite the need for a relation, it is important that you have a person on the team who is by nature a "problem maker". There is a reason to include a person with this profile on the team. Usually these people are intelligent and see problems differently than anyone else. If you feel that 7 people is too few to run the project, you could include other experts to be consulted at the start of the sprint. A good time is Monday afternoon. During this visit, they will be able to tell the team what they know and share their opinions. Now that the team is assembled, we need a facilitator. A facilitator should be a person who manages time, debates and the process in general very well. They must have experience in conducting meetings and know how to arbitrate the discussions, enforcing the time to stop and move on to another subject. The facilitator must always be impartial about decisions. The specialists included in the team should be from different areas and positions as each of them will make an essential contribution, be it with basic information, a new idea, or even useful information about your customers' vision [4].
4. **Time and Space**: To do a week of Sprint Design, you need a team willing to devote their entire time to being fully dedicated. By doing so, you will be able to generate one of the best aspects of Sprint: freedom to work the way you want, with an uncompromising schedule and a challenging goal to be met. On a typical day of a sprint, the team devotes 6 hours to be in the same room from 10am to 5pm, Monday through Thursday. The Friday will be dedicated to testing with users. Electronic devices are banned in the sprint room. There are studies that prove that when a person has their thoughts interrupted, it takes some time to return to the same point of thought. So the ban on electronics is useful in the sprint room. It is important to drive the sprint in the same room throughout the week so we can have white-boards that help us keep the best ideas and check if the focus of the work is not distancing itself from the main idea. White boards are best for viewing these

goals. We can not just rely on our brains. Now that we have the team, time and space defined, it's time to start the Sprint [4].

5. **Start at the End**: To help chart the purpose and goal of the sprint, the team should ask themselves: Why are we doing this project? Where do we want to be in 6 months, 1 year or even 5 years? The discussion to get these answers and the purpose of the Sprint should not be extensive and, to help, you can ask some questions:
   (a) What questions do we want to answer in the sprint?
   (b) To reach our long-term goal what do we need to do?
   (c) What can cause the project to fail?

Activities by day of the week:

- **Monday**: on Monday, you must choose the viable goal to be achieved and draw a map to achieve this goal. You should also ask for help or confer with experts who are not part of the team. After all this discussion one must draw a target to be reached. **Map** During the sprint week, it's very easy to get lost if we have not mapped our idea. On Monday we should already draw an initial map of what we think is the solution to the challenge. Alongside the map will help us choose a more realistic target within a broader challenge. In the same chart that is the map we should:
  1. List the actors: Who are the important actors of our story?
  2. Writing the end: It is easier to write the end than the path we will take to reach this end.
  3. Words and arrows in the middle: write an initial flowchart of how we will achieve this goal.
  4. Keep it simple: The map must have a few steps that everyone agrees on.
  5. Ask for help: we should ask the team if the map seems correct and, based on the answers, reach a map that adheres to the goal.

  The first map of the day may seem simple, but it will have improved over the course of the day's discussions. Once the map is created, we will interview the specialists who are not part of the team to gather more information about the problem. We have to keep in mind that it is necessary to get the maximum information from expert interviews. To do this, each member of the team should have a white-board to jot down questions that came up in the interviews. At the end of the day, we'll have more questions about how to solve the problems we'll face in order to achieve the goal. The team will also have lots of extra information on the problem and several questions to answer when looking for solutions to reach the goal [4]. **The target**. The last task on Monday is to choose the target for the sprint. What is the most important audience and what is their crucial moment in the experience of this audience [4].
- **Tuesday**: The day begins with reviewing existing ideas to fit and perfect. In the afternoon each member will sketch what they think is the best solution to the problem. Sometimes the best way to broaden your search is to look within your own organization. Great solutions come at the wrong time, and the

sprint can be the exact time to rescue them. Also, look for ideas in progress, but that are unfinished. Make a list: Ask everyone on staff to create a list of products or services to be analyzed for inspiring solutions. Do three minute demonstrations: The people who suggested the products make a presentation to show the team the interesting points. Make a note of good ideas as the presentations are made. The day begins with reviewing existing ideas to adjust and improve. In the afternoon each member will sketch what they think is the best solution to the problem. Sometimes the best way to broaden your search and look inside your own organization. Great solutions come at the wrong time, and the sprint may be the exact time to rescue them. Also, look for ideas in progress, but that are unfinished [4]. Make a list: Ask everyone on the team to create a list of products or services to be analyzed for inspiring solutions. Make three minute demonstrations: The people who suggested the products make a rush to show the team the interesting points. Post good ideas as presentations are made [4]. **Draw sketches** Explaining with words only is sometimes quite complicated and it is not so easy to make yourself understood. Ideally, one should use sketche. It is advisable to use the sketch technique in 4 steps. The steps are: Make notes: In this first step, you and your team will walk through the room, check the white boards and make notes. These notes are basically the best ideas of the last 24 hours, put together. Have ideas: in this step you should write the ideas that came up about solving the problem. Write down everything that comes to mind. Crazy 8's: This is an accelerated exercise. You should get your most interesting ideas and create 8 variations of these ideas in 8 minutes. Making the variations of an idea helps you to improve it by thinking of several solutions. Outline of solutions: Now it's time to create something to show the team. You should create a sketch of your best idea and put it on paper. Each sketch is a hypothesis for solving the problem. These sketches will be examined and judged by the rest of the team. The sketch created should: be self-explanation; be anonymous; be detailed, well thought through and complete; use words; have a striking title.

- **Wednesday**: on Wednesday morning the team will make a decision to select the most promising projects. The Battle on Wednesday morning, your team will make a joint decision to choose the most promising drafts. What if there is more than one winning sketch? If it is not possible to unite them in just one solution, the team will the prototypes and the client will decide which is the best, on the Friday. Having made the choice of the sketches, it is time to turn all decisions into a plan of action so that the prototype is ready on Friday. In order not to get lost and have a sketch ready by the end of Wednesday, the following tips are provided: Work with what you have: Work on the ideas that have come to you so far. Do not try to get new ideas. Do not write together: Your prototypes should have simple headers and important phrases. Include only simple details: Add enough detail to leave no doubt about the next step or where to go. One should not be too specific, a prototype does not have to be perfect. The Definer defines: On the Wednesday afternoon the team will be tired and it is very important

that the Definer does not let the discussions go on and defines some points that he thinks is the best decision. The Definer can ask for opinions or help from the specialists.
If you are in doubt, take risks: leave aside simple solutions and those have been tested elsewhere, seeing these solutions in prototype will not help much. Prefer to prototype large and bold ideas. Limit the story to fifteen minutes or less: You'll need to set aside time for iteration with customers. If we spend 15 minutes with the story, surely the presentation will take much more [4]. After all the sketches are included, the storyboard is finished. We will have overcome the hardest part of Sprint. Decisions have been made and the plan is outlined for the construction of the prototype [4]. **Storyboard**: It's time to draw up a plan and the schedule is short. It is possible to see customers testing the prototype and the desire to start is high. If the team starts without a plan, they may trip over small unanticipated obstacles, and the project may collapse. One should put the winning drawings on the wall and begin to think and draw what one imagines the finished prototype to be. This prototype, even on paper, should be rich in detail and depict not just the prototype but all the iterations you imagine for Friday.

- **Thursday**: After the storyboard is finished, it's time to be creative and turn the storyboard into a realistic prototype. **The prototype**: After much time spent testing, failing and succeeding, we have some tips to give for the prototype to come out in just one day: Choose the right tools; Divide and conquer; Sew everything together; Test. **Choose the right tools** Forget the day-to-day tools, they're just too perfect... and slow. Use something simple like Power-point or Keynote. You need to prototype, and doing that in a slide-show template and presentations sounds like a good idea. If it is not possible to do your prototype in a slide show, follow this advice: 1. For software, websites, applications, brochures, etc. Use Keynote, Power point or even Word. 2. If it is a service write a script and use the team as actors. 3. If it is an object, modify a pre-existing object, print on 3-D material, or prototype a marketing material [4]. **Divide and conquer**: The Facilitator should help the team divide into the following roles: Executor: 2 or more; Compiler: 1; Writer: 1; Resource Collector: 1 or more; Interviewer: 1. Executors create the individual components of the prototype. Usually designers or engineers. The compiler is responsible for assembling the components of the executors into a cohesive model. The writer is responsible for the impact words of the prototype. It is very important to have simple, straightforward phrases. Usually a product manager from the prototype end area. The resource collector is the role that will scan the web, image libraries, company products and all the possible places to find photos, icons or content samples that you did not need to create from scratch in order to illustrate the prototype. The interviewer is the one who will write an interview script that will be done with clients on Friday. After everything is ready, it's time to test. Around mid-afternoon re-convince the whole team and revise the prototype. You will still have time to compare it with the storyboard and make minor adjustments [4].

- **Friday**: After an incredibly productive week, you'll be face-to-face with the customer to present the product prototype. This test makes the whole sprint worthwhile: at the end of the day you'll know what to do next. **Interview**: An interview with the client can reveal a lot about the people who will use your product, problems where you did not think they existed and the reasons behind everything. The structured interview in the way that follows helps the client to feel comfortable and ensures that the entire prototype is analyzed: A friendly word of welcome; Context questions about the customer; Presentation of the prototype; Tasks for the client; Record customer's thoughts and impressions. Friday's action occurs in 2 environments. In the sprint room the team watches the interviews while they take place in another room that makes the client feel more comfortable. Once the presentations are over, it is time to look at interview videos, study user reactions, improve what needs to be improved and get to work. It's time to create the real product [4].

3 Comparative Analysis of Thinking Design and Sprint Design Methodologies

Technology is advancing all the time. With each new cycle, new transformational products are launched. To keep pace with these changes, it is necessary to evolve the process for building software. When the Rational Unified Process (RUP) [13] was released, it recorded a cycle of evolution that organized the way to develop software, bringing a methodology with remarkable practical efficiency, while maintaining a good theoretical wealth.

During good times the RUP was the standard methodology used by most software development companies. It would be remiss to not observe how much this methodology contributed to the quality of good software produced to the present day. Nowadays, in order to meet the need to increase speed without giving up quality, demand has arisen for more agile methodologies with fewer bureaucratic processes and consequently more lean ones.

It is from this demand that we observe the appearance of studies that propose the implantation of methodologies preaching agility and speed, with the reduction of bureaucratic processes which are considered, in some moments, unnecessary. We discuss two methodologies in this paper that propose to be more agile through the use of Design techniques. They are Design Thinking and Design Sprint. When observing how the two methodologies work, it is noticeable that they share many principles and techniques, as shown in Figure 3. The similarities between the two methodologies are so similar that one could erroneously believe that they are identical.

What are their most striking similarities and divergences between methodologies?

Design Sprint proposes a model of formation of the structure and the profiles of the members for each role, each task having a predetermined proposed time of execution understood as ideal. The format of a Design Sprint gives teams a way to focus the attention of the team on a very specific problem. The exercises embedded in the five phases are designed to reduce politics, increase collaboration across functions and put the focus on answers (outcomes) and not just assets (outputs). The Understand phase of a Design Sprint is used to discover, understand and define the problem that your customer is dealing with. If theres not enough user research or access to customers to make this worthwhile then a Design Sprint might be premature. Design Thinking does not impose this obligation of appear, being in charge of its members defining the time that will be spent in each activity, and specifying which profiles should execute each task.

Design Thinking proposes to solve complex problems through the mobilization of a multidisciplinary team, where the opportunity to gather knowledge and diverse experiences in the same team tends to enrich the solution found for the problem. The idea behind Design Thinking is to change the mental model in which companies are invested, to create products without first knowing the real needs of their target audience and the key players involved in the process. In theory, it can be represented in four phases: immersion, where the team delves into the implications of the challenge; ideation, in which there is a collaborative brainstorm with the use of practices of stimulus to creativity; prototyping, the phase used for the development of several models in search of the ideal product; and testing, in which the product will be validated with the users. The time can vary from weeks to months. In addition, Design Thinking is not a fixed methodology, so it does not have a step by step to be followed or a specific deadline in which it should be done.

Design Thinking really shines when we need to better understand the problem space and identify the early adopters. There are various flavors of Design Thinking, but they all sort of follow the double-diamond flow. Simplistically the first diamond starts by diverging and gathering lots of insights through talking to our target stakeholders, followed by converging through clustering these insights and identifying key pain-points, problems or jobs to be done. The second diamond starts by a diverging exercise to ideate a large number of potential solutions before prototyping and testing the most promising ideas. Design Thinking is mainly focused on qualitative rather than quantitative insights. For each phase of Design Think, there is a possibility of using a technique chosen for the situation. For example, in immersion phase, it could be research search, ethnographic research, check list, among others.

Design Sprint, on the other hand, is a method based on Design Thinking itself and created by Google Ventures. Beyond the theory, Design Sprint has a systematic application. The method is divided into five phases, designed to be performed in a short period of one week. The goal is to flexibly conceptualize and tangibility an idea, its implementations and macro functionalities in a short time. Design Sprint comes as a way to make projects on paper and focus on the ideas that come up but end up being ”shelved”. It is a fast and efficient method which, through a deepening of the problems that need to be solved, results in a solution ready to be implemented. Design Sprint is a process created by Google and that today has become routine within the work processes performed by them. It's a bit of the recipe behind the magic of the internet giant.

For the organization and the people, it serves as an impetus for the company to mobilize human resources and allow time for an idea to be made viable and validated. From a cultural point of view, Design Sprint is a way to bring agility and unity of teams and multidisciplinary teams and an incentive to activate an innovative organizational culture. It appears that the Google Venture-style Design Sprint method could have its roots from a technique described in the Lean UX book [19].

The key strength of a Design Sprint is to share insights, ideate, prototype and test a concept all in a 5-day sprint. Given the short timeframe, Design Sprints only focus on part of the solution, but it's an excellent way to learn really quickly if you are on the right track or not. Thus, it is up to each organization to define the sprint rules, including timing and extension. What is essential is that there is a focus, a specific problem that the organization or the team wants to address. If necessary, unmask macro challenges on micros. This is a way to address complex problems and achieve solutions.

The two methodologies are suitable for most situations, the choice will depend on the team profile, Sprint Design prevails a more formatted method with more defined deadlines. In Design Thinking, greater flexibility prevails in the process itself, which has a baseline that shows the main direction, but allows the team to do the way that suits them best. Choosing the ideal model to develop the project is critical to the success of the final product. Therefore, it is necessary to analyze the stage of maturity of the idea and the availability of time and resources of the client.

If you need to develop solutions Well, if you need to develop a solution or want to build something totally new, it's better to opt for Design Thinking, an approach focused on immersing and understanding a holistic context that complex problem is embedded in. Now, if your goal is to co-create the team to find the workable solution, Design Sprint is more recommended, since it is more objective in the process.

Runtime One of the great villains for innovation is perhaps the time, both for the dedication that your team has to have and for the level of innovation they must present. If the team needs to develop the solution or at least a Minimum Viable Product (MVP) quickly, Design Sprint is recommended. Now, if the idea is to understand the context in depth and then create a solution, Design Thinking fulfills this role well.

Learning Although the two approaches are based on collaboration and experimentation, Design Thinking has a more ”learn by sharing” character, since

Design Sprint is more ”learning by doing” due to the sprints' time and the speed one must have to create one MVP (Minimum Viable Product). Both Design Thinking and Design Sprint are approaches that significantly leverage the ability of people involved in being creative. Both approaches in fact, are characterized by tools and methodologies supporting idea generation like How Might We for Design Thinking or Crazy' s 8 for Design Sprint. The Design Sprint methodology promotes a more critical approach providing for a higher number of sessions dedicated to individual thinking compared to what is suggested by Design Thinking.

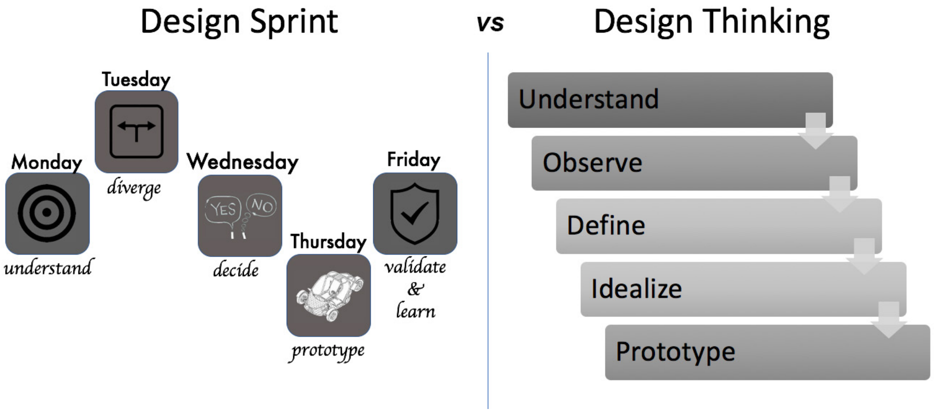

Fig. 3. Comparing Design Sprint versus Design Thinking

Learning by Prototyping The second aspect the two approaches have in common is undoubtedly the role of prototyping. Both methodologies dont just define the steps needed to imagine an idea, a solution, but enable the work team to reach the implementation through the development of a prototype. As regards Design Sprint these solutions appear to be much more concrete, as if aiming to simulate working prototypes, whereas in Design Thinking projects, prototypes sometimes are just Roadmaps describing how the solution will be developed, without providing physical evidence of the solution itself.

User Contribution Another difference regards the role that the end user has in the innovation process. The difference of involvement is connected to the fact that the Sprint comes from an internal company challenge while the Design Thinking process stems from the will to look at the needs users are facing and helping them to solve them. This is the reason why the tools used are different. With Design Thinking broad use is made of ethnographic research while in Design Sprint users are involved in tests such as the AB Test.

Process Dynamics This is one of the aspects that most differentiates the two approaches. If Design Thinking processes can last hours, days, months or even years, with Design Sprint they have a defined duration: 5 business days, from Monday to Friday. This difference in duration has repercussions on process dynamics. In fact: Design Thinking prefers diverging phases that generate innumerable new ideas through brainstorming moments. Design Sprint focuses on convergence to the point of dedicating three days out of five to it and places strong emphasis on decision making moments that are often dealt with through elections. Understanding in just five days (Design Sprint) activities which usually require a greater timeframe (Design Thinking) leads to reducing uncertainties and increase the potential success of a project. This test, which takes place on the fifth day and not after months, enables the team to understand the pros and cons of the solution and learn from it while avoiding that the team becomes enamoured with the solution developed internally and not being open to variations and ascendancy provided by users.

4 Conclusion

Design Thinking is a new way of thinking and approaching problems or, in other words, a people-centered thinking model.

Design Thinking was first cited in 1980, and popularized by the American design and innovation company, IDEO, which began using that approach to solve problems in a more humane way and spread the word around the world so that other companies could discover the power of design in the face of the problems they encounter.

Design Sprint was developed by Google Ventures - Google's Venture Capital arm - based on Design Thinking and agile methodologies, with the main goal of prototyping solutions in a much faster and more dynamic way.

Design Sprint is a five-day process for answering critical business questions through design, prototyping, and testing ideas with customers.

Both Design Thinking and Sprint Design have enormous power of innovation, but aligning expectations and defining why you're doing it will make the whole process clearer for all participants.

References

1. Tillberg-Webb, Heather and Collier, Ned.: Using the Design Thinking Cycle to Tell the Story of Innovative Learning Spaces, Educational Technology and Narrative, pp. 141–153. Springer (2018).
2. Maruping, Likoebe M and Venkatesh, Viswanath and Agarwal, Ritu.: A control theory perspective on agile methodology use and changing user requirements, Information Systems Research Journal, Vol. 20, number 3, pp. 377–399 INFORMS (2009)
3. Keijzer-Broers, Wally JW and de Reuver, Mark.: Applying Agile Design Sprint Methods in Action Design Research: Prototyping a Health and Wellbeing Platform,

International Conference on Design Science Research in Information Systems, pp. 68–80 Springer (2016)

4. Knapp, Jake and Zeratsky, John and Kowitz, Braden.: Sprint: How to solve big problems and test new ideas in just five days, Simon and Schuster (2016)
5. Padmanabhan, Pradyuta and Baez, Alexander and Caiseda, Carmen and McLane, Kathleen and Ellanki, Nithin and Seshaiyer, Padmanabhan and Kwon, Byong and Massawe, Erick.: Design Thinking and computational modeling to stop illegal poaching: Integrated STEM Education Conference (ISEC), 2017 IEEE, pp. 175–181, IEEE (2017)
6. Vianna, Mauricio.: Design thinking: inovação em negócios, Design Thinking (2012)
7. Du, Junpeng and Jing, Shikai and Liu, Jihong.: Creating shared design thinking process for collaborative design, Journal of network and Computer Applications, vol. 35, Number 1, pp. 111-120 Elsevier, 2012
8. Chamberlain, Stephanie and Sharp, Helen and Maiden, Neil.: Towards a framework for integrating agile development and user-centred design, International Conference on Extreme Programming and Agile Processes in Software Engineering, pp. 143–153 Springer (2006)
9. Awad, MA.: A comparison between agile and traditional software development methodologies, University of Western Australia (2005)
10. Sharma, Sheetal and Sarkar, Darothi and Gupta, Divya.: Agile processes and methodologies: A conceptual study, International journal on computer science and Engineering, vol. 4, number 5, pp. 892 Engg Journals Publications (2012)
11. Banfield, Richard and Lombardo, C Todd and Wax, Trace.: Design Sprint: A Practical Guidebook for Building Great Digital Products, O'Reilly Media, Inc. (2015)
12. Morien, Roy.: Agile management and the Toyota way for software project management, Industrial Informatics, 2005. INDIN'05. 2005 3rd IEEE International Conference on, pp. 516–522 IEEE (2005)
13. Kruchten, Philippe.: The rational unified process: an introduction, Addison-Wesley Professional (2004)
14. Pohl, Klaus.: Requirements engineering: fundamentals, principles, and techniques, Springer Publishing Company, Incorporated (2010)
15. Higuchi, Marcelo Makoto and Nakano, Davi Noboru.: Agile Design: A Combined Model Based on Design Thinking and Agile Methodologies for Digital Games Projects, Revista de Gestão e Projetos, vol. 8, n.2, pp. 109. Universidade Nove de Julho (UNINOVE) (2017)
16. Brown, Tim and Wyatt, Jocelyn.: Development Outreach Journal, vol. 12, pp. 29–43. World Bank (2010)
17. Talita Pagani.: Design Thinking, SENAC Journal, pp. 109. (2018)
18. Canedo, Edna Dias and da Costa, Ruyther Parente. The Use of Design Thinking in Agile Software Requirements Survey: A Case Study. International Conference of Design, User Experience, and Usability, pages=642–657, 2018, Springer.
19. Gothelf, Jeff. Lean UX: Applying lean principles to improve user experience. 2013, O'Reilly Media, Inc.