Utrecht University

Faculty of Science

Department of Information and Computing Sciences

Master in Business Informatics

Fostering Creativity in the Process of Designing Mobile Applications

Inspiring App Designers by Making Use of Analogical Reasoning and App Review Analysis

Master's thesis

Author:
Ilse L.N. van 't Hul, BSc
5675022

Supervisors:
Dr. Fabiano Dalpiaz
Dr. Sergio España Cubillo
Prof. Neil A.M. Maiden

Utrecht University

August 10, 2020

Abstract

Apps have gained more and more importance over the last decade. Unsurprisingly, the app market has become increasingly crowded. App designers may need to find ways to differentiate their apps from those of others. Creativity may be a resource needed to achieve this differentiation. Therefore, it may be worth fostering it. In this research, we set off to find out how the creativity of app designers can be fostered. We proposed a conceptual design for a tool that aims at achieving this objective. The design incorporates two main concepts that are central different fields, namely analogical reasoning and app review analysis. Taking this interdisciplinary approach aimed at tailoring general creativity research to the field in question to effectively support the creativity of the app designer. In essence, the proposed tool presents analogous apps as example solutions to a similar design problem. In line with that, this research proposed a way to describe apps in analogical terms and a method and associated theory to select appropriate analogous example apps. The tool also provides visualisations of features that are automatically extracted from the app reviews for each presented app. The approach proposed in this research combines both human and machine processing in the quest to foster the creativity of app designers. Both the automatic app review analysis and example app selection theory were validated. The preliminary tentative results of the validations give initial indications of the feasibility of the proposed design.

Acknowledgements

This thesis represents the end of a creative journey and thereby the end of my Master's. It was an interesting period, which I really enjoyed and in which I learned a lot. I want to thank everyone who was somehow involved in this project.

I want to especially thank Dr. Fabiano Dalpiaz for being such a great supervisor. Thank you for your availability and for your motivation. I learned a lot from our discussions and particularly from your helpful feedback. I also want to thank Prof. Neil Maiden for his involvement in this research project. His feedback and expertise have been very valuable to this research. I further want to thank Dr. Sergio España for accepting to be my second supervisor and for evaluating my work.

I would like to thank all persons who participated in my research. Your participation was highly important to my research. Besides that, I want to thank all Utrecht University researchers for participating in the workshop, for doing the pretest of my experiment, and for providing feedback. This has been really helpful to me. I would like to thank my friends for the nice time off. Thanks to Maarten in particular for our thesis discussions and for your helpful feedback.

I would like to thank my family, especially my parents, for always supporting me. Last, but certainly not least, I would like to thank Robin for always supporting me, not only on this journey. Our precious moments helped me a lot during this period.

Contents

Chapter 1

Introduction

During the last decade, mobile applications (henceforth "apps") have increasingly gained importance. App Annie (2019) reported that in 2018, there were almost two hundred billion app downloads and that on average people spent around three hours per day on a mobile device. Evans Data Corporation (2016) reported that in 2016 there were around twelve million mobile developers. It may therefore not be surprising that apps have received an increasing amount of attention from the research community (Wang, Wang, & Guo, 2019).

As there are many apps on the market to choose from (Wang et al., 2019), app designers and developers must find a way to differentiate their apps from those of their competitors (Dalpiaz & Parente, 2019). They may need to come up with novel yet appropriate ideas that help to differentiate them. Creativity may be a resource that is necessary to provide such differentiation (Horn & Salvendy, 2009). Creativity is a concept that has been studied for decades. It entails the generation of ideas that are both novel and useful (Mumford, 2003). The importance of creativity is manifest, for creativity forms the core of innovation (El-Sharkawy & Schmid, 2011). Also, creativity is seen as essential for maintaining an organisation's competitive position (Amabile, 1998; Ferreira, 2013; Gabriel, Monticolo, Camargo, & Bourgault, 2016). The need to keep innovating is also necessary in the software domain for attracting new consumers and for standing out between competitors (Lemos, Alves, Duboc, & Rodrigues, 2012). Lemos et al. (2012) argue that creativity should therefore be considered as essential in software development. One strand of software development that is concerned with the generation and elicitation of novel ideas is RE. RE is argued to be a creative problem-solving process (Maiden et al., 2010). This strand has also shown interest in the app domain (Nagappan & Shihab, 2016).

Creativity, however, does not come out of the blue; it needs to be fostered or supported (Boden, 2009). Many techniques, frameworks, and even tools have been proposed over the years to achieve this goal. Also in the field of RE, several techniques and tools for fostering creativity have been proposed and were found to be effective (Lemos et al., 2012). However, even though creativity has been studied for decades and it has been studied for several years in the field of RE, still little work has been done on creativity in the context of apps. Some researchers for instance have created an app to foster creativity (e.g., Zachos et al., 2013) and others have studied creativity in participatory app design (e.g., Davidson and Jensen, 2013). However, to the best of our knowledge, no work was specifically focused on fostering the creativity of app designers. Therefore, this research takes this opportunity to investigate how creativity can be fostered in the process of app design. Hence, the main research question is formulated as follows:

RQ: How can creativity be fostered in the process of designing mobile applications?

1.1 Sub-Research Questions

In order to find an answer to the main research question, several sub-questions must be answered first. Since this research is centred around the app design process, it is important to have a clear understanding of what this process entails. Moreover, a solid understanding of this process is needed in order to come up with a solution that fits the given context. Therefore, the first sub-question is as follows:

SQ1: What is the current state of the practice in app design?

It is not only important to get insight into the way in which apps are designed, but also to understand how creativity plays a role in this process. Besides that, it is important to understand how ideas emerge in this process. This results in the following sub-question:

SQ2: What role does creativity currently play in the app design process?

App markets can be seen as a rich source of information (Nagappan & Shihab, 2016). Since this research is conducted in the app domain, we are interested in exploring whether and how this source of information could be used in the strive for fostering creativity of app designers. Therefore, the third sub-question is as follows:

SQ3: How can app markets be used for fostering creativity of app designers?

As mentioned before, already various tools exist that aim at fostering or supporting creativity. However, it is unclear whether tools that foster creativity can effectively or sufficiently support the creativity of app designers. Therefore, the fourth and final sub-section is as follows:

SQ4: How can creativity of app designers be fostered through the use of a tool [1] *?*

[1] Even though one can interpret the term *tool* in a broad sense, the term refers in this context to an information system.

Chapter 2

Research Design

To find answers to the aforementioned research questions, several steps had to be taken. First of all, the research problem needed to be investigated and insight needed to be obtained into relevant concepts. Furthermore, this research had the objective of designing a tool that aims at supporting the creativity of app designers. Hence, the design of this tool had to be validated as well.

2.1 Research Approach

The research was structured along the design cycle of Wieringa (2014). This specific framework was selected, since its phases are consistent with the steps that had to be taken in this research. The design cycle consists of three main phases, namely 1) problem investigation, 2) treatment design, and 3) treatment validation. The research therefore consisted of three main phases, which were called the *Problem investigation phase*, the *Solution design phase*, and the *Solution validation phase*. All research questions were answered by completing all phases and sub-phases. Thereafter, the main research question could be answered as well. An overview of the research can be found in Figure 2.1.

The design cycle was supplemented by principles of Design Thinking (DT). Design Thinking is defined as an approach or as a method for innovative design that is human-centred and that makes intensive use of prototypes (Vetterli, Brenner, Uebernickel, & Petrie, 2013). Important aspects in DT entail early prototyping, human-centredness, iterativeness, and collaboration (Brown, 2008; Tschimmel, 2012; Vetterli et al., 2013). For instance, early prototyping allows for "early failure", which may help "to succeed sooner" (Vetterli et al., 2013, p. 93). The choice for supplementing the design cycle with principles of Design Thinking is best explained by looking at the nature of this research: *creativity*. It is claimed that DT helps designers in being creative by providing various models and techniques (Tschimmel, 2012). This research itself may be seen as a creative endeavour. Moreover, the main objective of this research entailed the design of a creativity support tool. Various researchers have argued that DT may be advantageous for software engineering (Valentim, Silva, & Conte, 2017) and requirements engineering (Hehn, Mendez, Uebernickel, Brenner, & Broy, 2019; Vetterli et al., 2013).

There are different perspectives on Design Thinking, namely DT as a *process*, as a *mindset*, or as a *toolbox* (Tschimmel, 2012). In this research, merely the latter two were considered. Various process models have been proposed to shape the DT process (Tschimmel, 2012). In general, they all take into account the same principles, but at a different level of granularity. According to Tschimmel (2012), there is not one best DT model and therefore, the selection of a model depends on the preferences and background of the designer. Hence, this first perspective, DT as a process, was not considered. Also this perspective was not considered for the design

cycle was already in place. The design cycle and the general DT steps largely overlap, meaning that the same general tasks are performed in both processes. In the perspective of DT as a mindset, concepts such as human-centredness and empathy (Brown, 2008) were kept in mind when designing the tool. The last perspective, DT as a toolkit, served as a guide for selecting methods and techniques for conducting this research. These are discussed in Section 2.2.3.

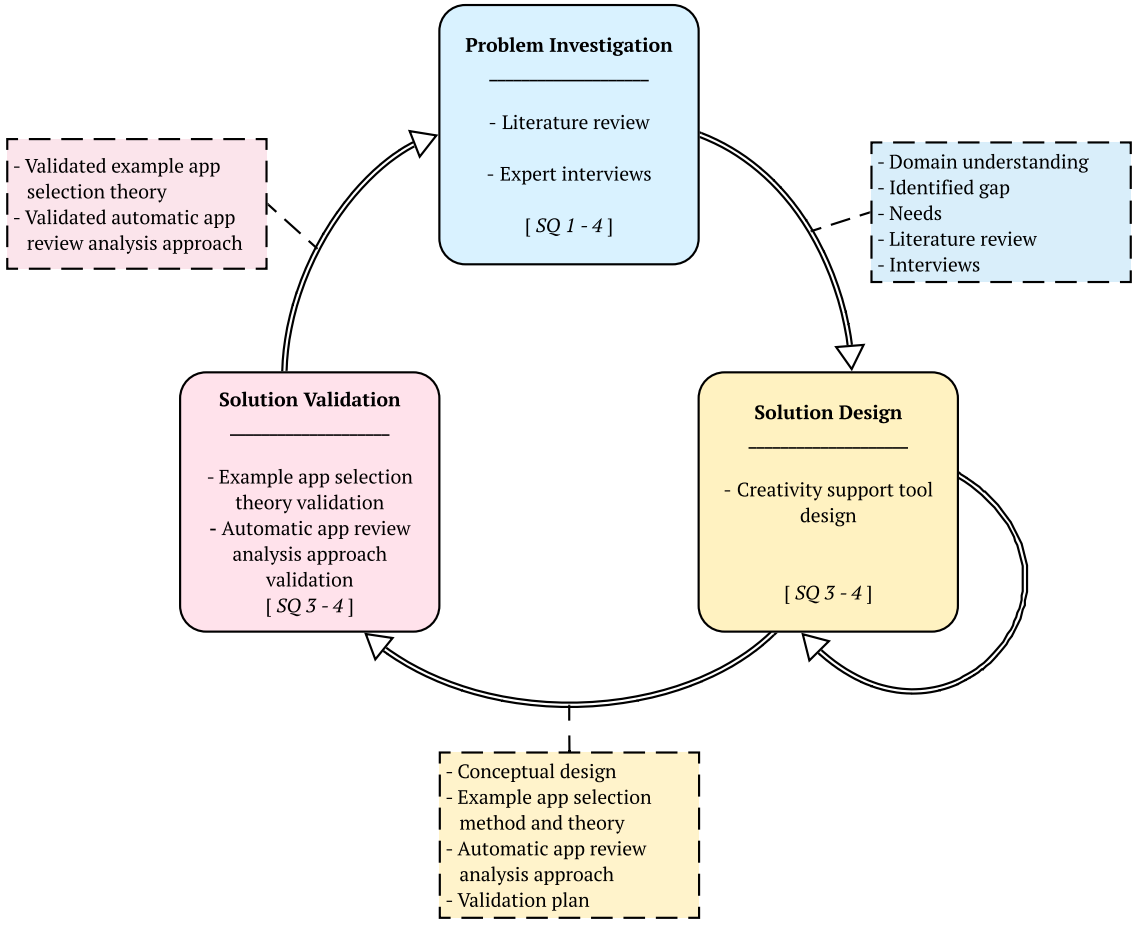

Figure 2.1: Overview of the research with its phases and artifacts. Based on Wieringa (2014).

During the early stages of the research, not every element of the research (e.g., research methods) was defined in detail yet. The reason for this was that we did not want to converge too early to specific ways of working or to specific solutions. It is stated that convergence is important for generating appropriate solutions. However, too little divergence may lead to the risk of getting stuck in generating novel ideas (A. Cropley, 2006). Moreover, Vetterli et al. (2013) mention that in DT, ambiguity should be maintained during the first phases. Thus, in the early stages, some elements of the research approach were not set in stone to prevent from constraining the creative process.

One design cycle was completed during this research. Initially, it was planned to deliver a working prototype after one cycle, which is in accordance with the early prototyping principle

of DT. However, as the research unfold, we gained more knowledge about creativity and about validating creativity. On top of that, we found out that there was no existing theory that could be readily applied to the design of the tool. Therefore, a theory fitting to the specific context of the research needed to be formulated that would form the basis of the tool. We thus decided to put more focus on the conceptual design and validation of our theory, instead of putting focus on the design of a prototype of the tool[1]. It is important to first make sure that conceptual design of the tool and the theory behind it are sound, before validating the tool as a whole. When validating the tool as a whole, various confounding factors (e.g., usability and user experience) need to be considered. By putting more focus on the conceptual design and the theory first, we strove to study the concepts relevant to this research in a more confined space. Instead of early prototyping, focus was put on early testing, in order to discover flaws earlier (cf. DT early prototype principle). In the following sections, the phases of the research will be elaborated on in more detail.

2.1.1 Problem Investigation

In the design cycle, the problem investigation is aimed at identifying an improvement problem (Wieringa, 2014). At this stage, the intended artifact has not yet been designed and the requirements have not yet been elicited. The main objective of this phase is to "*identify, describe, explain, and evaluate the problem to be treated*" (p. 41).

In this research, we frame the aspect to be studied not as an improvement problem, but as an improvement opportunity. We do not frame it as a problem, since we do not initiate our research from an observed struggle of app designers. However, the gap explained in Chapter 1 shows an opportunity for a potential improvement of the app design process.

The objectives of this phase were met in two ways, namely via a literature review and via expert interviews. The literature review had the purpose of obtaining knowledge about relevant concepts and getting insight into the current state of the art. The expert interviews had the purpose of supplementing the literature review with information that is not covered in academic literature. Also, Wieringa (2014) noted that the problem investigation is "real-world research" (p. 41) and that therefore the researchers have to observe the real world. This was also one of the reasons for conducting expert interviews. Moreover, DT suggests empathising with end users (Brown, 2008). A method that can be used for this is conducting interviews (Tschimmel, 2012). The literature review approach and the interview approach are discussed in more detail in Section 2.2.

2.1.2 Solution Design

The treatment design phase focuses on the design and/or creation of an artifact that could treat the problem. During this phase, requirements are elicited, existing treatments are investigated, and a new treatment is designed (Wieringa, 2014).

In the second phase of this research, the envisioned solution was designed. As already mentioned earlier in this chapter, the solution comprised in this case a conceptual design for a creativity support tool. Furthermore, as also already mentioned, a theory that forms the foundation of this eventual tool was formulated. The design of the tool and the formulation

[1]We argue that adjusting the research approach during this research does not have negative implications, since this research was exploratory in nature.

of the theory were based on knowledge obtained in the previous phase and on insight gathered through empirical analyses. This phase was iterative, which is in line with one of the DT principles. Small adjustments were made to the ideas, causing an organic evolution of ideas. Intermediate feedback was used to further fine-tune the design. On top of that, various people were involved in the design process to further evolve ideas for the tool.

Mainly during this phase, we tried to refrain from making early commitments regarding the design of the tool. Again, the importance of convergence is recognised. However, enough divergence is needed to stay open minded towards other ideas and solutions (A. Cropley, 2006). Thus, in order to come up with a creative solution, no early commitments were made regarding the design of the tool. Various options were explored (Section 5.3) and these led to a gradually evolving design of the tool.

2.1.3 Solution Validation

The treatment validation phase of the design cycle focuses on trying to validate whether the designed artifact would successfully treat the identified problem (Wieringa, 2014). Also, it aims at investigating whether requirements that were listed in the second phase are met.

Thus, elements of the conceptual design were validated during this last phase. The initial intention was to validate a working prototype of the tool with practitioners to determine their acceptance and the effectiveness of the tool regarding creativity. The reason for involving practitioners can also be traced back to DT. However, the validation plans altered, because of the shifted focus of the research. Moreover, as described in Section 3.7, creativity support tools can best be validated in longitudinal studies. Given the restricted time frame set out for this study, it would not be feasible to perform such an extensive validation. Furthermore, the COVID-19 pandemic put further restrictions on the validation. Inviting participants in a real-life setting was not possible, meaning that the validations had to be done online. Validating components of the conceptual design of the tool in a more confined context was assessed to be most suitable. The research comprised two main strands which were in line with the two major components of the conceptual design (Chapter 5). Therefore, two smaller validations were conducted to gather evidence for the proof of concept. Both an assessment of the performance of the automatic review analysis approach and a small experiment were conducted. More details on the entire validation can be read in Section 2.2.5 and Chapter 8.

2.2 Research Methods

The choice for the selected research methods can largely be substantiated with principles of Design Thinking. The several phases of DT, and of this research, come with unique objectives that may be guided by specific methods and techniques. Tschimmel (2012) discussed various methods and techniques that are relevant for DT projects. According to Tschimmel, the DT process mainly sets off with a literature review and with interviews. Besides that, various other techniques may be used during the process. This section describes the methods and techniques used in this research. An overview of the selected methods and techniques can be found in Table 2.1.

Table 2.1: The research methods and techniques used to find answer to the sub-questions.

Research Method	SQ1	SQ2	SQ3	SQ4
Literature review	x	x	x	x
Expert interviews	x	x		
Design Thinking techniques			x	x
Workshop			x	x
Experiment			x	x

2.2.1 Literature Review Approach

The literature review had various objectives in this research. First of all, it had the purpose of getting insight into relevant concepts and understanding the field, and thereby providing relevant background information. Secondly, literature was assessed to find further inspiration for this research. Lastly, the literature review served to identify gaps in the state of the art. Overall, it aimed at identifying relevant concepts, theories, and related solutions to substantiate the decisions made in this research.

The method used for the literature review is the traditional (narrative) review, as opposed to a systematic review. The reason for selecting this type of literature review method is that it suitable for obtaining an understanding of the field. Moreover, traditional reviews generally have a broad scope (Jesson, Matheson, & Lacey, 2011). This makes this method useful here, since this research is highly interdisciplinary and thereby broad in scope. Most importantly, according to Jesson et al. (2011) this type of literature review leaves room for creativity, which is very important given the nature of this research. Even though this method is criticised for not being transparent or complete, unlike a systematic review (Jesson et al., 2011), this method was still selected. First of all, this review did not have the objective of summarising or synthesising all works that have been done in the field. Furthermore, a systematic review may be very time consuming and requires already some understanding of the field from the researcher (Jesson et al., 2011). Given the restricted time frame and the limited understanding beforehand, we did not find it feasible and needed to conduct a systematic review.

Since the literature review is not systematic, no formal search protocol was maintained. However, some general steps can be identified in the search process. Literature was found by first identifying initial relevant topics and researches that relate to the aforementioned research questions and objectives. The initial topics were: *creativity*, *fostering creativity*, *analogies*, *app design*, *evaluating creativity*, and *tools*. After an initial exploration, this list was supplemented with the following related topics: *design-by-analogy*, *design fixation*, and *app review analysis*. These topics were used to form search terms. Examples of keywords that were part of search terms include: *creativity*, *analogical reasoning*, *app design*, *fostering creativity*, *creativity support tools*, *evaluating creativity*. Google Scholar was used as the main search engine to find literature. Also, international conferences, workshops, and journals were consulted through the dblp computer science bibliography. Access was obtained through the Utrecht University Library Access service. Articles were selected based on their relevance regarding this research and based on the author, number of citations, and/or the source of publication. After initial articles were selected, the snowballing technique was used to find additional relevant works. Snowballing was both done in a forward (i.e., 'cited') and backward (i.e., 'cited by') manner (Wohlin et al.,

2012). The literature review was ended when the following stop criteria were met. The first criterion entailed that the review was ended when it was assessed that for each identified topic a relatively complete overview was given of important and relevant works. Another criterion entailed that literature on these topics formed a coherent overview. The last criterion entailed that the review was ended when enough information was gathered to have a solid insight into relevant topics and potential gaps to be studied.

Table 2.2: Conferences and journal consulted to find app design research.

Journal or Conference	**Scope**
International Conference on Mobile Computing and Networking (MobiCom)	2010 - 2019
International Conference on Mobile Software Engineering and Systems (MOBILESoft)	2014 - 2019
International Conference on Software Engineering (ICSE)	2010 - 2019
International Conference on Mobile and Ubiquitous Multimedia (MUM)	2010 - 2019
International Conference On Mobile And Secure Services (MobiSec)	2016 - 2017
International Working Conference on Requirements Engineering (REFSQ)	2010 - 2019
International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)	2010 - 2019
Symposium on Internetware (Internetware)	2010 - 2018
European Conference on Software Maintenance and Reengineering (CSMR)	2010 - 2019
IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)	2010 - 2019
IEEE international requirements engineering conference (RE)	2010 - 2019
International Journal of Interactive Mobile Technologies (iJIM)	2010 - 2019
Empirical Software Engineering	2010 - 2019
Conference on Creativity and Cognition	2010 - 2019

To make sure a thorough overview of works on app design and creativity in app design could be given, a semi-structured way of searching literature was maintained for this topic. This way of searching literature supplemented the search method as described above. Already various articles were identified by searching on Google Scholar. From these articles, the journals and conferences were extracted. Moreover, a Google search identified additional relevant journals and conferences. These were all consulted through the dblp computer science bibliography. Some workshops were consulted as well. This decision was based on whether these were assessed to be relevant given the scope of this research. As can be seen in Table 2.2, a certain scope was maintained for each journal and conference. The default scope was set at works after the year 2009, since we reasoned that before that year the work on apps is outdated. The repository of Sarro (n.d.) was also consulted for research on app design. Literature was generally selected here based on whether the topic was about app design or creativity in app design. The relevance was assessed per selected work.

2.2.2 Interviews

As already mentioned, the second part of the problem investigation entailed conducting interviews with practitioners. Again, this method was selected based on the principles of the design

cycle and DT. These interviews were exploratory in nature, since they aimed at understanding the field and at identifying design opportunities. The interviews also needed to provide insight into the process of app design and into the role of creativity in that process.

The method chosen was semi-structured interviews (Wohlin et al., 2012). In this type of interviews, questions are prepared beforehand but the order of the questions is not set in stone. The questions may serve as a checklist of topics to be covered during the interview. Semi-structured interviews give the freedom to explore and ask further questions during the interview. Given the nature and goals of the interviews, this type of interview was found to be most suitable for this stage of the research. The interview protocol can be found in Appendix A. The interviews were conducted with app design practitioners. It was planned to conduct around five interviews, since we assumed that this would be a saturation point for new information. Saturation is, according to Wohlin et al. (2012), one stop criterion for conducting interviews. Participants were sampled based on convenience. In the context of this stage of the research, the term app designer was loosely defined. In the context of the interviews, an app designer was defined as:

"someone who is working at or for a company in which apps - mobile apps, hybrid apps and/or web apps - are designed or created and who is actively involved in that process".

One of the reasons for loosely defining app designer was to obtain enough interviews. Moreover, we reasoned that the app design process for web apps and mobile apps is highly similar. Furthermore, the most important aspect of the interviews was getting insight into the view on creativity and experiences with creativity of app designers. The most important selection criteria for the interviews was that the intended interviewee should be able to provide answers to the prepared questions. This judgement was made by either the author, her supervisors, or by the intended interviewee.

The interviews were divided into two main parts in order to achieve the main objectives of the interviews. The first part focused on getting insight into the role of creativity in app design. Questions were asked about the generation of novel ideas, the importance of creativity in the app design process, and possible difficulties with regard to creativity. The second part aimed at the app design process itself. Here, questions were asked about the steps taken during the process, the stakeholders, and the culture and strategy of the company. These questions were all asked in the context of creativity.

Since the interview were exploratory in nature, the results were not analysed quantitatively or in any systematic way. The results were processed by summarising them and reporting them in a descriptive way. It must be noted that this could introduce some subjectivity or misinterpretation. This was prevented as much as possible by trying to stay as close as possible to the words (or their meaning) of the interviewees (e.g., by quoting the interviewees). The threat of misinterpretations was already partly handled during the interview, by rephrasing and summarising the answers of interviewees and asking if that was correct. Moreover, when describing the responses of the interviewees it is tried to give as much context as possible in order to guide the reader.

2.2.3 Design Thinking Techniques

Tschimmel (2012) has given an overview of techniques that are helpful during the design (thinking) process. Tschimmel mentions that these techniques do not specifically come from DT,

but are essential for coming up with design solutions. These techniques include observation, mind-mapping, personas, brainstorming, sketching, storyboarding, and prototyping. Tschimmel (2012) notes that each technique may be more useful in a certain stage of the process. A selection of these techniques was used during this research to guide the design process. Techniques were selected in an ad hoc manner during the design and idea generation of the tool. In the end, mind-mapping, brainstorming, and sketching were used. These three were used to structure the thought processes and to foster the idea generation. It must be noted that these techniques were solely used as a means and were not used extensively.

Design Thinking was applied only in a lightweight manner. As already mentioned, no specific process model was adhered to during this research. Moreover, one important aspect of DT is the intensive involvement of end users during the design process. For this, co-creation in the design process is advised (Brown, 2008; Tschimmel, 2012; Valentim et al., 2017). Furthermore, it is also suggested to conduct observational studies. However, these aspects could only be done to a limited extent in this research, given the constraints of this project. Possible end users were involved in the first and last phase of the design cycle. Observational studies and co-creation were not conducted or applied. To sum up, DT was used to shape the research, but did not receive explicit attention during the research.

2.2.4 Workshop

A workshop was held to test the feasibility of one of the components of the conceptual design, namely asking human annotators to describe apps. This workshop was held to assess whether humans are able to consistently describe apps given a set of guidelines. The workshop took place in the form of an online discussion session. Beforehand, the participants needed to complete an exercise with a set of guidelines. Both the discussion and results of the exercise were of interest for the evaluation of the feasibility of the approach. The workshop is discussed in more detail in Section 7.3.1.

2.2.5 Experiment

To validate the underlying theory of the conceptual design, an experiment was set up. This experiment took the form of an online questionnaire which was at first only distributed among undergraduate and graduate students from Utrecht University. Due to a restricted response rate, the questionnaire was distributed via Reddit[2] as well.

An experiment was chosen over a survey for the validation, since the effect different treatments needed to be compared to confirm our theory (Wohlin et al., 2012). However, a questionnaire, which is a main method for gathering data in surveys, was used to gather the data. Conducting a short experiment that would measure the perception of participants was chosen for, since this method was found to be most feasible given the constraints. The experiment was human-oriented (as opposed to technology-oriented) (Wohlin et al., 2012), since the tested treatment needs to be used by humans in the future. The design, execution, analysis, and results of the experiment are further discussed in Chapter 8.

[2]https://www.reddit.com/

Chapter 3

Literature Review

This chapter discusses a variety of topics that were assessed to be relevant in the context of this research. Each of the aforementioned topics is discussed in a separate section.

3.1 Creativity

Even though creativity is a well-researched topic, there is little consensus about it. Many definitions have been assigned to the concept. Already forty definitions of creativity were found around 1960 by Rhodes (1961). Since then, various definitions have been added to the list. For instance, Boden (2004) defined creativity as "*the ability to come up with ideas or artefacts that are new, surprising and valuable*" (p. 1). Amabile (1983) defined that something needs to be "*both novel and appropriate, useful, correct, or valuable*" (p. 360) to be creative. Overall, many definitions of creativity delineate the generation of ideas that are both novel and useful (Mumford, 2003). Based on the definitions he found, Rhodes (1961) identified the following four facets of creativity: person, process, press (i.e., environment), and product. These '4Ps' of creativity comprise different analysable aspects of creativity (D. H. Cropley, 2016).

One model that provides a description of the creative process is that of Wallas (1926). Wallas' four-stage model of creativity is seen as a fundamental work (Sadler-Smith, 2015). The four stages, which are largely based on the work of Poincaré, are 1) *preparation*, 2) *incubation*, 3) *illumination*, and 4) *verification* (as cited in Amabile, 1983 and Sadler-Smith, 2015). 1) In the preparation stage, information and knowledge are gathered to get insight into the domain. 2) In the incubation stage, the information obtained in the previous stage is processed consciously and unconsciously. 3) The creative idea becomes apparent in the illumination stage. 4) Finally, in the verification stage, the creative idea is validated and used (Wallas, 1926) (as cited in A. Cropley, 2006 and Rhodes, 1961). This work is found to be useful for understanding the creative process (A. Cropley, 2006), even though it cannot be seen as a formal theory (Sadler-Smith, 2015). A. Cropley (2006) argues that each stage of Wallas' model either requires divergent thinking, convergent thinking, or a combination of both. Oftentimes, creativity is only associated with divergent thinking. However, not only divergent thinking, but also convergent thinking is essential for creativity (A. Cropley, 2006). Whereas divergent thinking results in novel ideas, convergent thinking may also be needed to come to appropriate (i.e., "effective") ideas (Figure 3.1). When both are applied simultaneously in a successful way, the outcomes are reasoned to be creative. However, when overused, convergent thinking could be detrimental to the creative process (A. Cropley, 2006).

Rhodes (1961) posed the question about how novel an idea has to be and to whom it should be novel. According to Boden (2009), something can be both historically creative (i.e., H-creative) and personally creative (i.e., P-creative). In the former kind of creativity, the resulting ideas is

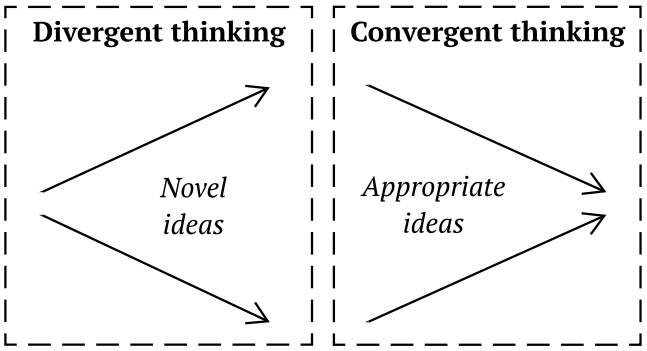

Figure 3.1: Divergent and convergent thinking. Based on the work of A. Cropley (2006).

historically novel, meaning that the idea did not exist before. In the latter kind, the resulting idea were already generated by someone else in the past. However, the idea was still novel to the person who just thought of that idea. Boden (2004) made another distinction in creativity. This distinction comprises three different types of creativity, namely combinational creativity, exploratory creativity, and transformational creativity (Boden, 2004):

- In *combinational creativity*, existing ideas are combined to come up with new ones. An example of this type of creativity, given by Boden (2009), is a collage.
- *Exploratory creativity* concern peoples' conceptual spaces. Within this type of creativity, existing ways of thinking are applied (differently) to come up with new ideas. An example of this type of creativity is creating a novel painting within a certain art movement (Boden, 2009).
- Within *transformational creativity*, the existing ways of thinking are changed to generate novel ideas. Boundaries are stretched and it thereby results in outcomes that seemed impossible before. This type of creativity also concerns peoples' conceptual spaces, just as it is the case with exploratory creativity. An example of this type of creativity is creating a painting in a different art style than those that already existed and that were already widely accepted (Boden, 2009).

3.2 Fostering Creativity

To be creative, one does not only need creative skills, but also motivation and domain expertise (Amabile, 1983). Boden (2004) also noted that both expertise and motivation are needed to be creative. Boden (2009) added to this that self-confidence is essential in the creative process. Boden argues that each of the three types of creativity needs to be fostered differently. An overview of how each creativity type can be fostered according to Boden can be found in Table 3.1. Amabile, Conti, Coon, Lazenby, and Herron (1996) proposed a conceptual framework comprising factors that influence creativity in organisations. This framework, which is based on a literature review and a critical-incidents study, describes factors that may influence creativity in work environments. It was reasoned that organisational encouragement, supervisory encouragement, group support, freedom, sufficient resources, and challenging work have a pos-

itive influence, whereas work pressure and organisational impediments were reasoned to have a negative impact on creativity in organisations. A validation of the framework indicated that only organisational encouragement, supervisory encouragement, group support, challenge, and organisational impediments have an impact on creativity in organisations (Amabile et al., 1996).

Table 3.1: How to foster each type of creativity according to Boden (2009).

Creativity Type	**Way of fostering**
Combinational	Practice to make combinations of existing ideas. Obtain more knowledge. Become used to evaluating creative combinations.
Exploratory	Become acquainted with and master a specific thinking style. Learn to apply a specific way of thinking differently.
Transformational	Become acquainted with a certain way of thinking. Become familiar with transformational creativity. Learn to comprehend how transformational creativity works. Practice to share ideas appropriately, so that others can recognise their value. Have self-confidence.

3.2.1 Fostering Creativity in Requirements Engineering

Many techniques have been proposed that aim at supporting creativity. Vieira, Alves, and Duboc (2012) found over 95 distinct creativity techniques, of which 41 were determined to be relevant for RE. Some examples of creativity techniques are brainstorming, synectics, mindmapping, idea triggers, and Six Thinking Hats (Sakhnini, Mich, & Berry, 2012; Scott, Leritz, & Mumford, 2004; G. F. Smith, 1998). Various research efforts have been devoted to fostering creativity in RE and software engineering. For example, creativity workshops were held in the process of RE (Maiden, Gizikis, & Robertson, 2004). These workshops were held, as part of the RESCUE process, to elicit requirements and produce ideas for a new system. The workshops were based on the concepts of convergence and divergence and on the four aforementioned stages of creativity. During various workshops, the three different types of creativity of Boden (2004) were fostered. The workshops resulted in 201 novel ideas for the envisioned system and various lessons learned (Maiden et al., 2004). This study illustrates how different creativity theories and models can be applied in RE research.

Besides that, creativity techniques were researched in the context of RE. According to Lemos et al. (2012), a variety of researches in the field of RE indicated that creativity techniques are useful for fostering creativity in RE, but also that many require a considerable amount of resources. In order to help software developers and requirements engineers in finding a suitable creativity technique, a catalogue has been created. This catalogue, called the Creativity Patterns Guide, gives direction on which technique to choose during a particular development phase (Vieira et al., 2012). An initial empirical study, in the form of creativity workshops, showed some first promising results regarding practitioners' satisfaction with the guide. Creativity techniques have also been proposed for the RE field. A technique specially created for fostering creativity in RE is the EPM Creative Requirements Engineering TEchnique (EPMcreate) (Mich, Anesi,

& Berry, 2005). This technique comprises sixteen steps that help the requirements engineer in assessing a problem from various perspectives of different stakeholders. Experiments indicated that this technique is both effective in generating creative requirements and promising regarding the acceptance of users (Mich et al., 2005; Sakhnini et al., 2012). Since its creation, two variations of this technique have been proposed, namely Power-Only EPMcreate (POEPMcreate) (Sakhnini et al., 2012) and Redundant, Odd Step EPMcreate (ROSEPMcreate) (Herrmann, Mich, & Berry, 2018). Both techniques are a smaller version of EPMcreate in which only four of the sixteen steps are used. Each technique contains a different subset of steps, meaning that the steps in both techniques do not overlap. Results from experiments with these techniques indicated to be in line with those of the validation of the original technique (Herrmann et al., 2018; Sakhnini et al., 2012). Another, "more lightweight" (p. 36), technique that has been proposed for the RE field is the creativity trigger (Burnay, Horkoff, & Maiden, 2016). Creativity triggers represent qualities that are found in creative products and that help in generating requirements that reflect those qualities. Examples of triggers are *service*, *trust*, *convenience*, *complete*, and *green*. Initial validations gave indications that creativity triggers are useful for fostering creativity in RE (Burnay et al., 2016).

Also several frameworks that describe the role of creativity in RE have been proposed. The C/RE framework aims at describing the role of and getting insight into creativity in information systems development and RE (Cybulski, Nguyen, Thanasankit, & Lichtenstein, 2003). The framework was later extended by Dallman, Nguyen, Lamp, and Cybulski (2005). The original framework comprised three aspects, namely context, outcome, and process. The extended framework expands the initial context aspect by including various factors that may affect the creative process in RE. Examples of those factors include motivation, perception and knowledge of creativity, experience and design bias, conformance versus risk taking, group leadership, group dynamics, stakeholder conflict, and organisational constraints and consequences (Cybulski et al., 2003; Dallman et al., 2005). Another creativity framework that has been proposed in the field of RE is called the Collaborative Creativity in Requirements Engineering (CCRE). This framework also describes factors that influence creativity in RE, but from a collaborative viewpoint (Mahaux et al., 2013; Mahaux, Nguyen, Mich, & Mavin, 2014). The framework makes a distinction between factors that influence individual creativity and factors that influence group creativity. The framework builds on the work of Amabile et al. (1996) that was described at the beginning of this section. Factors part of this framework include risk profile, culture, subject matter expertise, motivation, and facilitation and leadership. The C/RE framework has been constructed based on literature reviews and by conducting a focus group with RE practitioners and case studies with students (Cybulski et al., 2003; Dallman et al., 2005). The CCRE framework has been constructed based on a semi-structured literature review and a non-anonymous Delphi study with RE experts (Mahaux et al., 2013). The latter framework was validated in an empirical study. Results suggested that the framework is useful, but also that it should be used carefully, since other factors may also have an impact on creativity (Mahaux et al., 2014). As can be noticed, the discussed frameworks show a large overlap in their factors. For a more extensive list of the factors of the frameworks please consult the works of Dallman et al. (2005), Mahaux et al. (2013), and Mahaux et al. (2014)

Finally, various tools have been developed that aim at supporting creativity. These will be discussed in Section 3.5.

3.3 Analogies

Creativity and analogies are frequently associated with each other (Gentner et al., 1997). For instance, Boden (2004) described that analogies are useful, if not important, in the process of creativity. Gentner and Markman (1997) mention that analogies are applied in "creative discovery" (p. 45). It is claimed that analogies can alter perception (Boden, 1996) and that perception is an important ingredient in creativity (Boden, 2004). Gentner and Markman (1997) state that "*in creative thinking, analogies serve to highlight important commonalities, to project inferences, and to suggest new ways to represent the domains*" (p. 53). Some (e.g., Scott et al., 2004) see analogies as creativity techniques and others (e.g., G. F. Smith, 1998) see analogies only as aspects of creativity techniques.

3.3.1 Analogical Reasoning

An analogy is sort of comparison and is defined as a mapping from a source to a target (Gentner, 1983; Holyoak & Thagard, 1997). The source is the domain from which knowledge is taken, whereas the target is the domain that one tries to analyse or explain. The structure-mapping theory describes that both domains consist of objects, attributes, and relations. The attributes and relations are denoted as predicates in this theory (Gentner, 1983). Mappings can be both one-to-one and one-to-many (Holyoak & Thagard, 1997). Holyoak and Thagard (1997) reason that people are normally better at handling one-to-one mappings.

Gentner (1983) makes a distinction between several different sorts of comparisons. These sorts of comparisons are the following (Gentner, 1983):

- **Literal similarity**. In literal similarities, both many attributes and relations are mapped from the source to the target.
- **Analogy**. In analogies, many relations are mapped from the source to the target, but only few or even no attributes are mapped. This also explains the name of the theory: the structure matters, not the semantics (p. 165). In the context of this research, this type of comparison is of main interest.
- **Abstraction**. In abstractions, both domains consist mainly of relations that can be mapped. Thus, only few attributes are available in those domains.
- **Appearance match**. The appearance match is defined as a mapping in which many attributes are involved, but few relations.
- **Anomaly**. A comparison is denoted as an anomaly when there are no relations and no attributes that can be mapped, making it not useful.
- **Metaphor**. Metaphors can be seen as a special kind of comparison. In this kind, the mappings may mainly contain relations, mainly contain attributes, or may contain a mixture of both (Gentner & Markman, 1997). A large part of the metaphors can be seen as analogies, but some are just attribute matches (Gentner, 1983).

An important principle in the structure-mapping theory is the systematicity principle. This principle denotes then when source relations are interrelated, they have a higher chance of being applied to the target. In other words, the analogies may be stronger when the mapped relations are interrelated. A well-known, commonly used example of an analogy is Rutherford's analogy of the solar system and atoms (Boden, 2004; Gentner, 1983). In this analogy, the structure of the solar system is mapped to atoms to better understand the functioning of atoms. This

example clearly shows that the structure (e.g., 'revolving around'), rather than the semantics (e.g., size), matters in analogies.

The multiconstraint theory describes that a set of constraints is active when people apply analogies (Holyoak & Thagard, 1997). In general, there are three constraints, namely 1) similarity, 2) structure, and 3) purpose. The first constraint suggests that each analogy is based on a similarity between the source and target. As discussed at the beginning of this section, the similarity should mainly be found in the relations and not in the attributes of the two domains (Gentner, 1983). This relates to the second constraint, which states that the structure is essential in the comparison between the source and domain. The third constraint describes that each mapping is driven by the objectives of the person who is making the analogies (Holyoak & Thagard, 1997).

Various steps that describe the process of analogical reasoning have been identified. For instance, Holyoak and Thagard (1997) and Thagard, Holyoak, Nelson, and Gochfeld (1990) described four different steps in analogy, namely 1) access or retrieval, 2) mapping, 3) inference, and 4) learning. In the first step, a relevant source domain is identified. In the mapping step, comparisons are made between the source and target domain. In the inference step, the identified mappings are used to apply knowledge from the source domain in the target domain. Finally, the last step involves learning as a result of the analogical process (Holyoak & Thagard, 1997). Hall (1989) proposed a framework that is claimed to be more relevant for computational approaches to analogies. This framework consists of the following four steps that can be used to describe analogical reasoning: 1) recognition, 2) elaboration, 3) evaluation, and 4) consolidation. The first step also aims at identifying a suitable source domain. The second step entails making mappings and transferring knowledge between the source and target domain. In the third step, the mapping is assessed with regard to its context. In the last step, the results of the analogical mapping are consolidated, so that these can later be applied in different situations (Hall, 1989).

Several studies have argued that it may be complicated for people to apply analogies (Gentner, 1983; Gick & Holyoak, 1980; Maiden et al., 2004). However, those researchers also argued that the application of analogies can be valuable in, for instance, problem-solving and design.

3.3.2 Design-by-analogy

Analogies are frequently applied in design tasks (Goel, 1997; Linsey, Wood, & Markman, 2008). The term design-by-analogy, or analogical design, has been coined to describe design processes in which analogies are applied. As Figure 3.2 illustrates, this process entails the mapping of aspects or concepts that are part of a solution of one domain (i.e., the source) to solve a design problem in another domain (i.e., the target) (Goel, 1997). To aid designers in solving problems, solution examples are frequently used in design-by-analogy (Chan et al., 2011; K. Fu et al., 2013; Linsey et al., 2008).

The concept of *analogical distance* is frequently researched in the context of analogical design (Chan et al., 2011). Some have even assigned special terms to this, namely within-design-by-analogy and between-design-by-analogy (Verhaegen, D'hondt, Vandevenne, Dewulf, & Duflou, 2011). Analogical distance denotes that the domains involved may be closely related or seemingly very dissimilar (Chan et al., 2011; Christensen & Schunn, 2007; K. Fu et al., 2013). In near-field analogies (also within-domain analogies), the source and target may reside in the same or similar (problem) domain(s). In far-field analogies (also cross-domain analogies or between-domain analogies), the source and target come from reasonably different (problem) domains

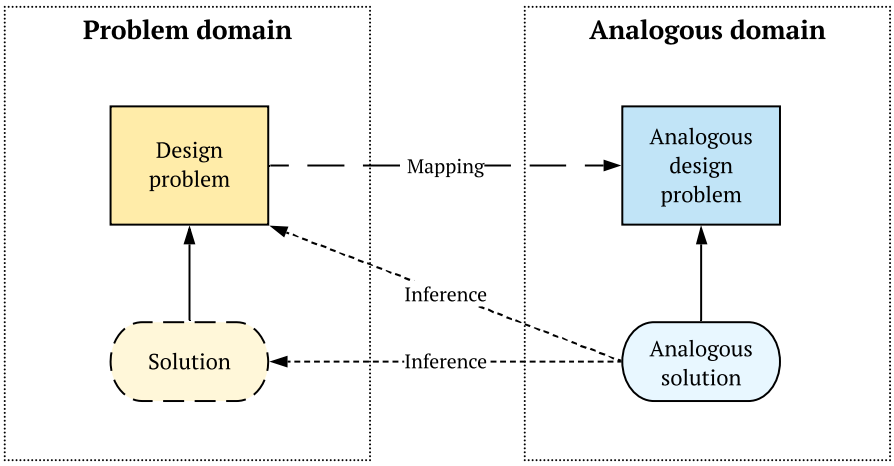

Figure 3.2: Design-by-analogy. Based on the definition of Goel (1997).

(Chan et al., 2011; Christensen & Schunn, 2007; K. Fu et al., 2013). An example of a far-field analogy is using ideas that are taken from nature when designing a piece of furniture. An example of a near-field analogy is looking at other types of furniture when designing that a piece of furniture (Verhaegen et al., 2011). The source and target domains are more similar on the surface in near-field analogies than in far-field analogies (Chan et al., 2011; Christensen & Schunn, 2007). Thus, in far-field analogies only few attributes are mapped, whereas in near-field analogies higher numbers of attributes are mapped (Verhaegen et al., 2011). Goel (1997) makes a distinction between analogical design (i.e., design-by-analogy) and case-based design. Goel argues that in case-based design the exact solution to the problem in the source domain is slightly altered and applied to problem in the target domain. This author claims that case-based design is a restricted type of analogical design, since all objects are mapped, but in a modified way. In case-based reasoning, the two problem domains are very much alike, whereas in analogical reasoning the two problem domains may be more different (Goel, 1997).

The distance between the problem domains is determined differently between various studies. K. Fu et al. (2013) state that analogical distance is commonly seen as a dichotomous variable. However, they argue that it is rather a continuous variable and that the meaning assigned to it is context dependent. Goel (1997) states that in AI literature, the terminology used to describe domain concepts determines whether design problems reside in the same domain or in a different one. In the study of Chan et al. (2011), the difference between near-field and far-field was specified based on whether the source domains presented (i.e., example solutions in the form of patent documents) were determined to have the same intended functionality as the target domain (i.e., the design problem to be solved). In a later study, K. Fu et al. (2013) made the distinction between near fields and far fields by determining the semantic similarity between the source domains (i.e., example solutions in the form of patent documents) and the target domain (i.e., the design problem to be solved). For this, a hierarchy was created in which the root node was the target domain and the other nodes were clusters of semantically similar source domains. The similarity between each cluster and the target domain determined the positions of the clusters. Similarity was found by applying latent semantic analysis to the

patent texts, to the description of the design problem, and to the clusters. The distinction between near-field and far-field was determined by the number of nodes between the source and target domains. The distinction between near-field and far-field analogies was least complex in the study of Christensen and Schunn (2007). In their study, it was determined whether an analogy was near or far based on whether the mappings were within the target problem domain (i.e., medical plastics) or not.

Gentner and Markman (1997) argue that the creativity of ideas is more obvious when they are a result of the application of far-field analogies. The outcomes may seem more novel or surprising, since the compared domains may look very dissimilar on the surface. Results from the research of Dahl and Moreau (2002) indicate that the use of far-field analogies has a positive influence on the originality of the created designs. However, the far-field analogies themselves may be difficult to apply (Gick & Holyoak, 1980). Results from the study of K. Fu et al. (2013) indicate that the distance of analogical examples influences the quality and novelty of design solutions. Their research indicates that there is an optimal distance, meaning that "*there may be such a thing as "too near" and "too far" when searching for analogies to employ ...*" (p. 021007-9). Overall, Dahl and Moreau (2002) suggested that by supporting the use of analogies, design outcomes will become more original.

3.4 Role of Examples

As already discussed, examples are frequently used in processes such as design-by-analogy. However, using examples has different implications on, for instance, creativity and may be related to concepts such as *design fixation*.

One can argue that providing examples can be seen as providing a form of knowledge. A. Cropley (2006) argues that the generation of novel ideas may be hindered by earlier obtained knowledge. The provision of examples and prior knowledge is related to the concept of design fixation (Jansson & Smith, 1991). According to Crilly and Cardoso (2017), "*design fixation is a state in which someone engaged in a design task undertakes a restricted exploration of the design space due to an unconscious bias resulting from prior experiences, knowledge or assumptions*" (p. 6). Jansson and Smith (1991) argue that providing examples leads to design fixation, since people tend to reuse elements of the examples that are shown to them in their own solution. Several studies have shown that presenting examples increases the amount of reuse of elements of the examples in the designer's own design outcomes (Dahl & Moreau, 2002; Jansson & Smith, 1991; Linsey et al., 2010; Marsh, Landau, & Hicks, 1996; Marsh, Ward, & Landau, 1999; S. M. Smith, Ward, & Schumacher, 1993). People even seem to reuse example elements when they are explicitly asked not to do so (Jansson & Smith, 1991; Marsh et al., 1999; S. M. Smith et al., 1993). Moreover, people even seem to reuse defective example elements (Jansson & Smith, 1991). When trying to replicate the study of Jansson and Smith (1991), Purcell and Gero (1992) could, however, not find consistent evidence for design fixation. Participants in their study did not consistently reuse elements of the shown examples. They noted that part of the difference in research outcomes could potentially be contributed to the difference in types of participants used. The subjects of both studies showed differences in the level of design expertise and came from different fields. In a later study, Purcell and Gero (1996) could also not find unambiguous answers to the question of whether showing examples leads to design fixation. In their study, results differed between participants from different design fields and differed between different types of measures of fixation. They found that in some cases showing examples leads to an

increased reuse of elements. However, showing elements did not significantly result in a reduced number of design ideas generated. The difference in their conclusions is partly attributable to the way in which design fixation is operationalised. Crilly and Cardoso (2017) note that design fixation research may be improved by, among other things, clearly defining how fixation should be measured.

There are mixed results on the impact of providing examples on concepts such as creativity and originality. The results from the study of Jansson and Smith (1991) suggested that creativity is hindered by providing examples. This study was concerned with the generation of product designs. Participants who were shown examples scored lower on creativity measures than participants who were not shown any example. The study of Dahl and Moreau (2002) indicates that providing examples does not positively contribute to the originality of the products of a design task. This study was also concerned with product design. Their research also indicated that when (near-field) examples are given, people tend to use less far-field analogies as compared to near-field analogies. On the contrary, the results from the study of Marsh et al. (1996) indicate that creativity is not limited by the provision of examples, even though participants reused elements of the examples. The experiments in this study involved designing novel creatures. They argued that since the total number of generated ideas and the average number of generated features did not differ between the example and control group, creativity was not hindered. Furthermore, subjective creativity ratings were higher for the group that was shown examples. Finally, they found that example elements were not specifically reused detriment of novel or uncommon elements. However, Marsh et al. (1996) did argue that creativity may be hindered when the reused elements are defective. Crilly and Cardoso (2017) argued that design fixation and the reuse of elements may not always have negative implications, since they reasoned that it may save resources and increase acceptance of stakeholders.

One of the reasons for the mixed results regarding the effect of providing examples on creativity is the difference in the definition assigned to and measurement of creativity. This is also something that Marsh et al. (1996) observed. One may also ask whether it is desired to accomplish H-creativity or whether P-creativity suffices. For instance, Mohanani, Ralph, and Shreeve (2014) researched creativity by measuring originality, which they equated to H-creativity. However, Marsh et al. (1996) argued that creative ideas are hardly really novel, since they frequently contain elements of existing ideas. One view on creativity that is in line with this, is combinational creativity of Boden (2004), which entails the combination of existing ideas. Furthermore, S. M. Smith et al. (1993) mentioned that element reuse is not always unfavourable, but in creativity it may be unfavourable as novel ideas can be needed as existing ideas may not be sufficiently helpful anymore. It is not surprising that Crilly and Cardoso (2017) argue that more research must be done to better understand the relation between creativity and design fixation. Also the form of the examples may have influence on their impact. This is discussed in the next section.

3.4.1 Form of Examples

The form of the examples may also have a different impact. Many studies investigating design fixation and the effect of providing examples used near-field examples, meaning that these examples were closely related to the design task and its envisioned solution (see e.g., Jansson and Smith, 1991; Linsey et al., 2010; Marsh et al., 1996). Dahl and Moreau (2002) also used near-field examples (i.e., sketches of potential solutions to the design problem) in their study

and reasoned that using far-field examples may foster originality instead of reducing it. In the experiments of Gick and Holyoak (1980), subjects obtained different textual example solutions that all could be seen as far-field examples. Subjects received examples in the form of stories that described a solution to an analogous problem in order to solve a medical problem. The experiments indicated that the subjects who were shown the analogous examples were able to produce solutions to the medical problem that were similar to the solutions to the analogous problem. In their second experiment, they even showed that the percentage of subjects who were able to come up with the envisioned solution was higher for subjects who received an example than subjects who did not receive any example. However, subjects who did not obtain any example were able to generate a higher number of alternative solutions on average. They concluded that "*the more effective the story analogy is in prompting the analogous solution, the more it inhibits production of alternative, disanalogous proposals*" (p. 337). They also found that in many cases, hints were needed to come up with the envisioned solution (Gick & Holyoak, 1980). As already discussed before, Chan et al. (2011) and K. Fu et al. (2013) researched the difference between far-field and near-field examples (i.e., in the form of patents). Results from the former study indicated that far-field examples lead to more transfer of example elements compared to near-field examples. The ideas resulting from assessing far-field analogies were found to be more novel compared to near-field examples. In the latter study, the authors reevaluated the results by assessing and comparing the distances of the examples used in both studies. They found that in the former study, the far-field examples were still close to the target domain and even closer than the near-field examples of the later study. So, K. Fu et al. (2013) concluded that there is an optimal distance of analogical examples for the impact on the novelty and quality of design ideas. They added that the definition of near and far may differ between researches. Their study indicates that far-field examples do not necessarily lead to an increased transfer of example elements.

The way in which the examples are represented also seems to have an influence on its impact. Many of these studies presented examples in the form of pictures (see e.g., Jansson and Smith, 1991; Linsey et al., 2010; Marsh et al., 1996). The design task in the study of Marsh et al. (1999) entailed the creation of nonwords. The examples they provided were thus in a textual form. Chan et al. (2011) found that participants who got an example in a textual form reused more example elements than participants who got an example in a graphical form. They also found that those who got textual examples generated less ideas compared to participants who assessed graphical examples and participants who did not assess any example. However, there was no difference in number of ideas generated between participants who assessed graphical examples and participants who did not assess any example. They could not find any differences in novelty of the ideas between the three groups. Purcell and Gero (1992) tried to determine the difference in impact between verbal solution examples and graphical solution examples. They were not able to determine this, since they could not consistently significantly determine whether participants reused example elements. Christensen and Schunn (2007) found that even preliminary solution examples, in the form of prototypes, made by the designers themselves may inhibit the use of far-field analogies. They reasoned that thereby creativity is hindered. Sketches, however, were not found to reduce the number of far-field analogies made.

Chan et al. (2011) also investigated the influence of the commonness of shown examples on the design solution. The authors defined common examples as examples that participants may frequently observe in daily life, whereas less common examples may not be frequently observed in daily life. Results indicated that participants who received more common examples came

up with less solutions than participants who received less common or who did not receive any examples at all. However, there was no difference between participants who received less common examples and participants who received no examples. Furthermore, participants who received less common examples scored higher on novelty of the ideas than participants who received more common examples (Chan et al., 2011).

3.5 Creativity Support Tools

As mentioned in Section 3.2, various tools have been developed that aim at fostering or enhancing creativity. Put simply, a creativity support tool (CST) can be seen as a tool that aims at fostering creativity (Shneiderman et al., 2006; Voigt, Niehaves, & Becker, 2012). In general, researchers refer to some form of information system when discussing CSTs. However, some define CSTs more broadly, stating that CSTs can be any tool that can be used in the generation of new products (e.g., Cherry and Latulipe, 2014). Frich, MacDonald Vermeulen, Remy, Biskjaer, and Dalsgaard (2019) also observed that there is little overall agreement on the definition of this concept. Their definition of a CST is as follows: "*A Creativity Support Tool runs on one or more digital systems, encompasses one or more creativity-focused features, and is employed to positively influence users of varying expertise in one or more distinct phases of the creative process.*" (p. 10). The term creativity support system (CSS) is also used to describe information systems that aim at supporting creativity. It seems that both terms are used literature to denote the same concept.

Many different types of CSTs exist that are created for a wide variety of intended users. Gabriel et al. (2016) reviewed 49 publications that discuss or propose a CST and Frich et al. (2019) even reviewed 143 distinct publications in which new CSTs were proposed. Both studies took a different approach and used a different search strategy. Probably as a result of that, the reviewed publications of these two studies do largely not overlap, if even at all. This indicates that already many CSTs and types of CSTs exist. These CSTs are created for a wide variety of users, including designers, writers, researchers, students, engineers, artists, and architects (Frich et al., 2019; Shneiderman et al., 2006). Various researchers have discussed different types of CSTs. For instance, Shneiderman (2007) gives an overview of various types of creativity support systems, which includes information visualisation tools, software development tools, media sharing tools, and concept mapping tools. Shneiderman et al. (2006) described CSTs at a higher level of granularity. They talk about tools for searching, visualisation, collaboration, and composition. Nakakoji (2005) uses a sports metaphor to make a distinction between the roles a CST can have. In this metaphor, a CST can be seen as *dumbbells*, *running shoes*, and *skis*. Systems that are compared to dumbbells aid people in acquiring or developing creative skills. Systems that are seen as running shoes are systems that aim at fostering and encouraging the creative process. The last type of systems, skis, allows for activities that would not be possible in absence of the tool. The difference between the running shoes and the skis is that without the latter certain activities could not exist, whereas without the former they could still exist, but would be different. Shneiderman (2007) identified three different views on creativity that may provide different perspectives on and reasons for developing CSTs. The first one, the structuralist view, argues that methods and systematic ways of working are essential for creativity. This view would opt for tools that help to structure the creative process. The inspirationalist view argues that incubation and inspiration are important concepts in creativity. This view would suggest the creation of tools for searching, visualising, and mapping information. The situationalist view,

which sees creativity as a social endeavour, would benefit from collaboration and communication tools. Shneiderman (2007) argues that search tools may be quite helpful, since creative people often need to explore possibilities, need to learn from the work of others, and may need to use those works as a starting point for their own work. However, Shneiderman also notes that this may create challenges regarding intellectual property rights.

CSTs can be created for a variety of devices, such as computers, tabletops, mobile phones, and tablets (Frich et al., 2019). Many CSTs are developed in the form of a web application (Frich et al., 2019; Gabriel et al., 2016). Most CSTs proposed in literature take the form of a prototype (i.e., both high and low fidelity) (Frich et al., 2019). CSTs can be created to support multiple phases of the creative process. However, most CSTs do not focus on more than one phase of the creative process. The ideation phase seems the most popular phase for the creation of these tools (Frich et al., 2019; Gabriel et al., 2016). Finally, Gabriel et al. (2016) found that most CSTs support some form of collaboration.

An example of a creativity support tool that is based on the design-by-analogy concept is Idea-Inspire (Chakrabarti, Sarkar, Leelavathamma, & Nataraju, 2005). This tool provides product designers with inspiration from of natural and artificial systems for the generation of ideas. This tool makes use of analogical reasoning to search in databases with natural and artificial systems. An updated version of the tool has been proposed after various year (Chakrabarti et al., 2017). Another tool that aims at providing designers with inspiration from nature is DANE (Vattam, Wiltgen, Helms, Goel, & Yen, 2011). This tool also makes use of analogical reasoning and gives designers access to models of biological and engineering systems. These tools mainly differ in the way the systems are represented in the databases. Also several tools for supporting creativity have been proposed in the context of RE. Examples of this are the works of Karlsen, Maiden, and Kerne (2009), Solis and Ali (2010), and Zachos and Maiden (2008). Platzer and Petrovic (2011) presented a preliminary design for a tool for app designers. The design of the tool incorporated app review analysis to identify general "motives for any human activity" (p. 45) for apps. The tool presents rankings and trends of usage motives in apps and shows examples of apps for which the motives are identified most often in the user reviews. To the best of our knowledge, this work has not been implemented afterwards. They frame their tool as various types of tools. They frame it mainly as a learning environment, but also a innovation support tool and a creativity tool. However, since their work had a unspecific focus on creativity and was not based on creativity theory, we do not consider their tool as a CST. For a completer and more diverse overview of tools, please consult the works of, for instance, Frich et al. (2019) and Gabriel et al. (2016).

Already quite some research effort has been put into the design of CSTs. Various design principles or theories have been proposed in literature. For instance, during a workshop on CSTs, Shneiderman et al. (2006) came up with twelve design principles for CSTs. These principles include (p. 70): *assist extensive explorations*, *allow for collaboration and for different ways of working*, *keep the design simple*, *perform various iterations*, and finally *test the designed CST*. Voigt et al. (2012) also observed that already many different design theories exist for CSTs. As a response to this, they proposed a unified design theory for CSTs. For this, they conducted a systematic review of eight CST design theories in which they identified three latent variables that form the basis of design theory for CSTs. These variables, called playfulness, comprehension, and specialisation, cover eleven design principles that are extracted from the eight design theories. It was hypothesised that all three latent variables have a positive influence on creativity.

Shneiderman et al. (2006) argued that people may have a critical attitude towards the cre-

ation of CSTs. They describe for instance, that people may think that computers negatively influence imagination. Shneiderman (2002) stated that people might argue that computers should not interfere with creativity, since it is a human endeavour. However, according to Shneiderman (2007), "*creativity support tools extend users' capability to make discoveries or inventions from early stages of gathering information, hypothesis generation, and initial production, through the later stages of refinement, validation, and dissemination*" (p. 22). Frich, Mose Biskjaer, and Dalsgaard (2018) found that a lot of research in the field of HCI on creativity focuses on the creation of new CSTs and that relatively little research is done on analysing actual systems in practice. Frich et al. (2019) stated that this could lead to the "*risk of reinventing the wheel*" (p. 2). However, Shneiderman (2002) noted that tools specifically created for a certain field will be more effective. Moreover, multiple researches have argued that more CSTs are needed in the field of RE (Lemos et al., 2012). Thus, one may argue that even though proposing a new tool comes with a risk, it may still be fruitful.

3.6 App Design

An app is an application specially created for mobile devices (Nagappan & Shihab, 2016). One can make a distinction between three types of mobile apps, namely native apps, web apps, and hybrid apps (Jobe, 2013):

- A **native app** is an app that is created for a certain mobile operating system. An advantage of these types of apps is that they can make use of the hardware components of the device (e.g., microphone).
- A **web app** is an application that runs within a browser, but that has the look and feel of a native app.
- A **hybrid app** can be seen as a combination of the aforementioned two. Web programming languages are used to create the apps, but the apps are run like native apps and can also make use of the hardware components.

The creation of mobile apps can be traced back to the 1990s. The version of mobile apps that many are familiar with nowadays arose only in 2007, the year in which also the first app market came to its existence (Nagappan & Shihab, 2016). Over the years, developing apps has become very interesting for organisations, because of the good market prospects, the ease with which apps can be distributed via app markets, and the wide potential customer reach (Nagappan & Shihab, 2016; Pagano & Maalej, 2013). According to Vetterli et al. (2013), apps are different from traditional software systems that are linked to large back-end systems. They argue that apps are small in size, may be altered frequently, are fast, provide only a restricted number of functionalities, and are hardly connected to back-end systems. In an analysis of a set of open-source Android apps, Minelli and Lanza (2013) also found that apps are different from traditional software. They found that apps are smaller, provide fewer functionalities, and make great use of external libraries.

To date, the two most widespread mobile operating systems are Google's Android and Apple's iOS. Both have their own app market, which are respectively the Play Store and the App Store. These app markets are one of the reasons why apps are popular in research, since they provide an extensive source of information that was not available beforehand (Nagappan & Shihab, 2016). Much research has been also done on app reviews and app descriptions. This will be discussed in the next section. Wang et al. (2019) conducted a large empirical research

on the characteristics of app developers (i.e., covering individuals to large organisations) across seventeen Android app markets. They analysed more than six million apps of over one million developers. They found that few developers are responsible for the largest part of app downloads, as 1% of the developers is responsible for 83% to 92% of the downloads across various app markets. They found that the Play Store is the largest player in the field, since more than half of the developers release their app to that market. It appears that most developers only release one app, but some distributed thousands of apps. However, most of them did not seem to be one individual developer but a variety of developers that distributed their work under the same signature. Finally, the findings of Wang et al. (2019) suggest that many developers do not actively release new versions. More popular developers (i.e., in terms of downloads) seemed to be more proactive than less popular developers.

A quite extensive amount of research has already been devoted to apps. However, to the best of our knowledge only limited research has been done on the app development process. Nagappan and Shihab (2016) describe that the "traditional" software development phases also apply to app development. The phases they identify are requirements, development, testing, maintenance, and monetisation. Wang et al. (2019) describe four phases that are associated to apps, namely development, release, maintenance, and promotion. They argue that app developers are a vital part of the app creation process, since they are involved all phases. Inukollu, Keshamoni, Kang, and Inukollu (2014) indicate that the phases of traditional software development are also applicable to app development. The phases they followed in their work are: requirements, design, development, testing, and maintenance. To the best of our knowledge, also a limited amount of work has been done on the process of app design or on creativity in that process. Davidson and Jensen (2013) researched creativity of elderly in the design of a healthcare app. Zachos et al. (2013) designed an app to foster creative reasoning of people in dementia care.

3.6.1 Analysing App Reviews and Descriptions

Substantial research has already been done on app reviews (see e.g., Chen, Lin, Hoi, Xiao, and Zhang, 2014; Jha and Mahmoud, 2019). Besides that, app descriptions have also been a topic of research (see e.g., Jiang, Ma, Ren, Zhang, and Li, 2014). This section gives an overview of research on app review and description analysis. It is not attempted to provide a complete overview here, because this is outside of the scope of this research.

Mainly the app reviews have become a topic of interest in RE research (Nagappan & Shihab, 2016). Examples of this are the works of Guzman and Maalej (2014), Dalpiaz and Parente (2019), and Pagano and Maalej (2013). Pagano and Maalej (2013) analysed more than one million reviews of 1,100 apps. They found that most users give a review for free apps and that most users (i.e., 90%) only write one review. Also, they found that most people leave a review after a new release, reviews are rather concise, and reviews mainly discuss multiple topics. Overall, seventeen topics can be identified in reviews, of which *praise*, *helpfulness*, and *feature information* are the most prevalent. Their results indicated that, in general, the length of the reviews seems to be related to the rating given, with the higher ratings being given with shorter reviews. The different topics receive on average different ratings. Reviews of the topics *recommendation*, *helpfulness*, *feature information*, *how-to* and *praise* represent the highest ratings (i.e., on average above 4.7), whereas *dissuasion*, *dispraise*, *bug report* and *shortcoming* reflect the lowest ratings (i.e., on average below 2.2). The researchers noted that automatic review analysis can be difficult, since the quality of app reviews may be low and expressions may

be subtle. Nagappan and Shihab (2016) also noted that the quality of app reviews introduces obstacles for researchers. Pagano and Maalej (2013) also noted that large amount of feedback are not interesting for app developers, and suggested thereby to filter and classify reviews.

In the past years, several research efforts have been put into classifying app reviews. Maalej and Nabil (2015) proposed an approach for classifying reviews into four categories. These categories were partly based on the aforementioned topics of Pagano and Maalej (2013). The reviews were classified as *bug report*, *feature request*, *user experience*, and *ratings* using supervised learning and basic string matching. Chen et al. (2014) proposed a computational framework for analysing app reviews called AR-miner. AR-miner first filters out reviews that are not informative through the use of semi-supervised classification and then groups ranks and visualises the informative reviews. Instead of classifying entire reviews, Gu and Kim (2015) classified individual review sentences, thereby taking into account that each review may comprise multiple topics. The review sentences were classified into the categories *aspect evaluation*, *praise*, *feature request*, *bug report*, and *others*, which were derived from the work of Pagano and Maalej (2013). The review sentences were classified with a supervised machine learning approach using lexical and structural features. In a response to this study, Shah, Sirts, and Pfahl (2018) compared this approach to a simpler bag-of-words (BoW) approach for representing features. For the comparison, they replicated the results of Gu and Kim. They found that the BoW approach performs nearly as well as the approach of Gu and Kim (2015) (i.e., average F-scores of 0.73 and 0.74 respectively), while being less complex.

Also efforts have been put into extracting features from app reviews. For instance, Johann, Stanik, Alizadeh B., and Maalej (2017) proposed SAFE, an approach to extract features from app descriptions and app reviews based on predefined patterns. These patterns included part-of-speech (POS) patterns and sentences patterns. They managed to obtain an average precision of 0.24 and an average recall of 0.71 on the app reviews. In a replication study, however, Shah, Sirts, and Pfahl (2019) only managed to obtain an average precision of 0.12 and an average recall of 0.54. Al-Subaihin et al. (2016) extracted features to cluster apps given their functionality. Features were extracted from app descriptions by applying the framework of Harman, Jia, and Zhang (2012) in which features were extracted through the use of patterns, collocation, and hierarchical clustering.

Finally, various studies focused on extracting user opinions regarding apps from the app reviews. For example, Guzman and Maalej (2014) perform feature extraction and sentiment analysis to determine whether users have a positive or negative stance towards features. Features were extracted as collocations consisting of nouns, verbs, and adjectives. After extraction, features were first grouped using Wordnet and were later merged into high-level features using Latent Dirichlet Allocation (LDA). Gu and Kim (2015) performed aspect-opinion extraction and sentiment analysis to extract the users' opinion about certain aspects of the app. The aspect-opinion pairs were extracted using predefined semantic dependency patterns. Similar to the work of Guzman and Maalej, aspects were grouped. However, in this study, aspects were grouped by applying frequent item (i.e., aspect words) mining. The main difference with the approach of Guzman and Maalej is that with this approach opinion words are extracted in addition to the sentiment scores. Moreover, Gu and Kim first filter out sentences that are not an aspect evaluation and that thus introduce potential noise. The feature extraction of Guzman and Maalej achieved an average F-score of 0.55, whereas the aspect extraction of Gu and Kim achieved an average F-score of 0.81. These differences could possibly be attributed to the fact that Gu and Kim filtered out irrelevant review sentences.

B. Fu et al. (2013) analysed app reviews at three different levels, mainly focusing on why users dislike an app. These levels comprise review level, app level and market level. With their tool (i.e., WisCom), rating inconsistencies per review, the development of user opinions per app over time, and market trends can be analysed by applying LDA and regression modelling. Vu, Nguyen, Pham, and Nguyen (2015) proposed a semi-automated framework called MARK with which keywords about which users have an opinion in app reviews can be extracted and suggested. Besides extracting keywords from app reviews, MARK ranks them based on frequency and rating, groups them using Word2Vec and k-means clustering, and returns the most relevant reviews for group of keywords. Finally, MARK visualises the presence of keywords in app reviews over time. In a later study, Vu, Pham, Nguyen, and Nguyen (2016) proposed PUMA, an automated approach for extracting phrases about which users have an opinion through the use of POS patterns. The pattern templates were automatically extracted from reviews. Similar phrases were again grouped, but now by using a custom similarity measure and soft clustering. PUMA also visualises trends of the phrases. In both studies, the focus was on negative opinions. The polarity of the opinion was based on whether a review obtained a positive rating (i.e., four or five stars) or a negative rating (i.e., one or two stars). Both studies return keywords or phrases about which users have an opinion, but those do not return specific words describing the opinion like in the work of Gu and Kim (2015). The performance of PUMA was only assessed through two case studies to show that the approach works. MARK was empirically validated by letting experts determine whether keywords were relevant or not (i.e., an acceptance rate). The two methods for recommending of keywords (i.e., keywords clustering and expanding) obtained an average accuracy of 83.1% and 89.7%.

3.7 Evaluating Creativity

Already some evaluations of creativity were discussed in the previous sections. This section will dive further into this process. According to Horn and Salvendy (2006), each of the four Ps of creativity of Rhodes (1961) can be evaluated. One reason for evaluating the outcomes of creativity is to assess whether creativity has been successfully fostered (Horn & Salvendy, 2006). This section only discusses the evaluation of the products or outcomes of creativity, since this current study is focused on fostering and enhancing creativity. Amabile (1982) also argued that the evaluation of the product is most suitable for empirical research of creativity. Rhodes (1961) defined a product as an idea that has been expressed or made tangible. However, many define the term product more broadly, including intangible artifacts such as ideas, processes, and services (e.g., Besemer and O'Quin, 2011; D. H. Cropley, Kaufman, and Cropley, 2011; Horn and Salvendy, 2006).

As discussed earlier, many definitions of creativity are maintained. It is therefore not surprising that this concept is measured and evaluated in many different ways. Some have evaluated creativity by counting the number of ideas generated and assessing their usefulness in a certain context (e.g., Maiden et al., 2004). Mich et al. (2005) also not only counted the number of ideas generated, but assessed the feasibility and novelty of these ideas as well to evaluate EPMcreate. Jansson and Smith (1991) used measures for flexibility and originality to assess the creativity of design outcomes. In many studies, creativity is only defined as something novel and useful (Mumford, 2003). Besemer and O'Quin (2011) mentioned that merely taken into account these two aspects of creativity is not enough. They argue that a third aspect, called style or elegance, is also needed to cover the reception and emotion of the beholder or evaluator.

3.7.1 Product Creativity Evaluation Instruments and Techniques

Different instruments or techniques for evaluating creativity of products have been proposed over the years (Table 3.2). Horn and Salvendy (2006) have reviewed various evaluation instruments and techniques for product creativity. Their work provides a comparison between different ways of evaluating product creativity. They discuss two ways of evaluating creative outcomes, namely rating scales and subjective assessment. This distinction is in line with the three ways of evaluating creativity of products as identified by Besemer and O'Quin (2011). The three ways of evaluating creative products are, according to them, 1) indirect measures of product creativity, 2) global judgements, and 3) evaluations based on criteria. An example of the first type they give is self-reported creative activities and achievements. The second and third correspond to the subjective assessment and rating scales of Horn and Salvendy (2006) respectively. The second way involves experts or other judges who evaluate creativity of products. A prominent example of this is the Consensual Assessment Technique (CAT) of Amabile (1982). This technique is grounded in the following consensual operational definition of creativity: "*A product or response is creative to the extent that appropriate observers independently agree it is creative*" (Amabile, 1982, p. 1001). Appropriate observers are defined as people who have some form of expertise in the domain of interest. Within the context of this technique, people are asked to come up with creative ideas under certain preconditions. Then, judges are asked to independently rate those products on certain dimensions, of which one is creativity. Amabile (1982) has also defined some preconditions for this. For instance, no specific definitions or criteria should be made available or agreed on before the evaluation takes place. Also, the products should be evaluated in comparison with each other. After the evaluations, the results are analysed and interrater reliability is obtained. Experiments of Amabile (1982) have indicated that the technique is reliable and that judges do not need to have high levels of domain expertise. Disadvantages of this technique include that it is time consuming and that it may be less suitable for highly innovative products (Horn & Salvendy, 2006). Another disadvantage of this technique is that comparisons of products can only be made within one sample (Besemer & O'Quin, 2011).

An example of the third type of creative product evaluation is the Creative Product Semantic Scale (CPSS) proposed by Besemer and O'Quin (1986) (Besemer, 1998; Besemer & O'Quin, 1999; O'Quin & Besemer, 2006). O'Quin and Besemer (2006) explicitly mentioned that the CPSS can be used for all kinds of products, including ideas, prototypes, and tangible products. The CPSS is based on the Creative Product Analysis Model (CPAM) of Besemer and Treffinger (1981) (Besemer & O'Quin, 1999). This model comprises three dimensions that describe essential aspects of creativity. The three dimensions of the CPAM are novelty, resolution, and style. These dimensions are composed of in total nine sub-dimensions. The CPSS consists of in total 55 bipolar adjective pairs, that are structured along those nine sub-dimensions. Each item pair is answered on a 7-point scale (Besemer, 1998; Besemer & O'Quin, 1999; O'Quin & Besemer, 2006). The CPSS has been refined and adapted various times, resulting in different numbers of items used (Besemer, 1998). The CPSS was found to be reliable in various studies and has been found to be applicable in various domains (Besemer & O'Quin, 2011). Also, the instrument may be used to compare products between samples, as opposed to CAT (Besemer, 1998). Horn and Salvendy (2006) argue that the limitations of this instrument encompass its definition of creativity, the absence of objective creativity evaluation criteria, and its subjective character.

Horn and Salvendy (2006) argue that both CAT and CPSS have a restricted usability and usefulness in the context of evaluating creativity. They concluded their paper by arguing that

Table 3.2: Overview of the discussed evaluation techniques and instruments.

Type of evaluation technique	**Name**	**Description**
Subjective assessment	CAT	Evaluation technique in which experts independently and subjectively rate creative products (Amabile, 1982).
Rating scales	CPSS	Creative product evaluation instrument that can be used to assess creative products in the broadest meaning of the term. Dimensions of the instrument are: novelty, resolution, and style (Besemer & O'Quin, 1999, 2011)
	PCMI	Creative product evaluation instrument that is claimed to be more valid than CPSS. Dimensions of the instrument are: novelty, importance, and affect (Horn & Salvendy, 2009).
	CSDS	Creative product evaluation instrument that aims at being fast to apply by various kinds of evaluators. Dimensions of the instrument are: relevance and effectiveness, problematisation, propulsion, elegance, and genesis (D. H. Cropley et al., 2011).

"*more valid, applicable, and predictive measures of product creativity*" (p. 172) are needed. As a response to that, the authors proposed in a later study another product creativity evaluation instrument, called the Product Creativity Measurement Instrument (henceforth PCMI). This instrument was later refined and validated by Horn and Salvendy (2009). The refined instrument comprises three dimensions of product creativity (i.e., affect, importance, and novelty) and fourteen items. Horn and Salvendy (2009) claimed that this instrument has better construct validity and predictive validity than the CPSS. D. H. Cropley et al. (2011) mentioned that only limited research has been done on the evaluation of creative products that are tangible, scientific, or technological. As a response to that, they developed and proposed a creativity evaluation instrument called the Creative Solution Diagnosis Scale (CSDS). This instrument is based on a theoretical framework of product creativity proposed by D. H. Cropley and Cropley (2005). The revised version of CSDS consists of 27 items that relate to five different aspects of (product) creativity. These aspects comprise relevance and effectiveness, problematisation, propulsion, elegance, and genesis. The instrument is created with the aim of being fast and easy to apply by various kinds of evaluators with a high agreement (D. H. Cropley et al., 2011). The study of D. H. Cropley et al. (2011) indicated that the CSDS is valid, reliable, and easy to use.

As already lightly touched upon, one of the difficulties of evaluating creativity and creative products seems to be the lack of consensus on the concept and its definition. Various researchers have argued that this lack of consensus leads to subjectivity during the evaluation. For instance, Horn and Salvendy (2006) argued that until there are any objective evaluation criteria for creativity, evaluations will be inherently subjective. Besemer and O'Quin (2011) also noted that the evaluation of creativity is inherently subjective. Moreover, various researchers have argued that seemingly objective tests or ratings scales for creativity are more subjective than is initially perceived (e.g., Amabile, 1982; Horn and Salvendy, 2006).

3.7.2 Evaluating Creativity in Computer Science

Creativity has also been evaluated in the context of computer science. For instance, Davidson and Jensen (2013) used an adapted and shortened version of the CPSS to evaluate the creativity of app designs. Maiden et al. (2004), Lombriser, Dalpiaz, Lucassen, and Brinkkemper (2016) and Mich and colleagues have evaluated creativity in the context of RE. Also, AI models of creativity have been created. For example, Maher and Fisher (2012) proposed an AI approach for evaluating creativity of design outcomes which incorporates three aspects of creativity. Finally, Zeng, Salvendy, and Zhang (2009) proposed an instrument for evaluating creativity of web sites. The instrument, consisting of 28 items, was based on a literature study and on the instrument of Horn and Salvendy (2009), and was refined by applying factor analysis. Zeng et al. (2009) also developed a conceptual model for user perception of and reaction to website creativity. The construct validity of the instrument was based on literature and an experiment showed internal consistency.

3.7.3 Evaluating Creativity Support Tools

Evaluating creativity support systems is not straightforward, just like evaluating creativity. For instance, Shneiderman (2007) argues that evaluating these kinds of systems is harder than evaluating systems such as productivity systems. The reason for this is that the operationalisation of the concept creativity is a difficult task. Shneiderman et al. (2006) state that the evaluation of CSTs cannot be properly conducted in controlled experiment, since these do not account for learning curves and developments that happen over a longer period of time. They argue that CSTs should be evaluated in longitudinal observational studies and with interviews. Shneiderman et al. (2006) also noted that individual methods, measures, or studies on their own are not sufficient. So, they state that to properly evaluate a CSTs, studies should be conducted over a longer time spans and with combinations of methods.

Cherry and Latulipe (2014) proposed the final version of a measurement instrument for evaluating CSTs, called the Creativity Support Index (CSI). This instrument, in the form of a survey, comprises six dimensions. These dimensions are the following: enjoyment, exploration, expressiveness, immersion, results worth effort, and collaboration. The survey consists of two parts, namely a rating scale and a paired comparison of the dimensions. Each dimension is represented by two items in the rating scale part. Each item takes the form of a question that must be answered on a 10-point Likert Scale ranging from "highly disagree" to "highly agree". The items for the collaboration dimension may be skipped in case the system is not intended for collaboration. In the second part, each dimension is compared to another dimension, resulting in fifteen comparisons. The outcome of the survey indicates how good a particular CST is at supporting creativity in a certain task given that the subject has a certain expertise with the tool. Cherry and Latulipe (2014) argue that this instrument is suitable for longitudinal studies and that it can be used in combination with other measures or methods as suggested by Shneiderman et al. (2006). The instrument has been tested in various studies. Also, in one of the studies, the reliability of the tool was tested. Results indicated that the overall test-retest reliability of the instrument were good.

It seems that even though methods and measures exist for evaluating CSTs, most systems are still evaluated using standard creativity or usability measures (Frich et al., 2019). Frich et al. (2019) re-emphasised the need for a wide variety of evaluation measures and methods, as a single approach is not sufficient for the wide range of different systems.

Chapter 4

Expert Interviews

As already discussed in Section 2.2, interviews were conducted to get insight into app design in practice. In total, five interviews were conducted, since at that moment a saturation point was reached. The interview protocol with the questions asked during the interviews can be found in Appendix A.

The interviews were conducted with app designers with a variety of backgrounds (Table 4.1). The experience they had with app design differed per person. In general, each interviewee was involved in at least one project in which a mobile app was designed, may it be a native app, a web app, or a hybrid app. Only one interviewee did not have any professional experience with the design of a native app. However, this interviewee had experience with multiple web app projects. The two UX designers showed the most experience with app design as they had worked on multiple app design projects for various clients.

The type of projects the interviewees talked about also varied. Three interviewees mentioned that they had designed apps for clients, while two interviewees (i.e., the company owners) mentioned to only have designed an app for their company. Besides that, the companies where the interviewees worked at also varied in, for instance, size and type. Some of the interviewees had worked at multiple companies where they were involved in the app design process. Three interviewees had experience working at a start-up. Another interviewee (I2), mentioned to also have done individual projects for clients, besides designing an app as part of employment duties. The size of the teams in which each interviewee had worked varied. However, overall, the teams were relatively small (i.e., generally, between two and six members). Interviewee I5 mentioned that also an external design team was involved in the design of the app they were working on, resulting in a higher number of people involved in the project.

Table 4.1: Interviewee background information.

Interviewee ID	Current job title	Current company type
I1	Consultant	Software company
I2	UX designer	Software company
I3	Company owner	Start-up
I4	UX designer	Design company
I5	Company owner	Start-up

4.1 Creativity in App Design

The first main part of the interviews comprised questions about creativity in the app design process. Interviewees were asked questions about their experience with creativity in the projects they worked on. As various interviewees worked on multiple app design projects, they could use this experience and knowledge to answer the questions from different perspectives.

4.1.1 Emergence of Ideas

The way in which ideas seem to emerge during the app design process, differed per interviewee. Interviewee I1 mentioned that ideas start with a customer wish to create or realise something. The app design team first gathered information about the client and conducted various stakeholder interviews. These interviews then resulted in initial ideas. The interviewee also mentioned that the design team found it very important to constantly take the end user into account. Interviewee I5 also mentioned that it is important to listen to the end user. This interviewee told that "new ideas emerge when you take a step back and listen to someone". During their app design process, they interview target users and non-target users and carry out field observations. This interviewee told that the creative process only begins when you do not take the literal words of people, but when you start analysing and translating them to find a solution. Interviewee I4 described a similar way in which ideas emerge. This interviewee told that first, "boundary conditions" (e.g., time, budget, and corporate identity of client) need to be made clear. For this, information is gathered, conversations are held with stakeholders, and theory is consulted. What is important, according to this interviewee, is that first, structure is needed beforehand in order to know where to start working on. This interviewee told that ideas arise when information is gathered and when sketches are made. The other UX designer also mentioned that drawing helps in generating ideas. This interviewee described that in the beginning, it may be difficult to decide where to start and that sketching helps with that. This interviewee tries to generate ideas by empathising with the end user as the process is very user centred to this interviewee. On top of that, sometimes, when talking to other people, ideas suddenly come to mind with this interviewee. These conversations do not necessarily need to be about the same topic as the emerged idea. Interviewee I3 mentioned that in the beginning, the ideas for the app emerged from a personal frustration. Now that the app is more mature, ideas are either generated by the team or elicited from the end users. Multiple interviewees mentioned that they ask themselves questions during the generation of ideas. Examples are: "What are we going to do now and what is important?" and "If I was an end user and I am going to do something in which the app is going to play a role, which actions do I take and how is the app going to support me with that?".

In general, ideas seem to emerge both in isolation as in collaboration. However, it seems that most of the times, the interviewees do not come up with ideas in mere isolation. One aspect that all interviewees mentioned, is that end users are involved in some way in the process of generating ideas. Also, discussions within the team is central in their idea generation process. Overall, most of the interviewees mentioned that information needs to be gathered somehow or some form of knowledge needs to be obtained for ideas to emerge. This is in line with the fact that creativity can be fostered by obtaining more knowledge (Boden, 2009). Moreover, also domain expertise is needed to be creative (Amabile, 1983; Boden, 2004). Thus, it seems that practice is consistent with theory here.

4.1.2 Inspiration

Becoming inspired is intertwined with the emergence of ideas. The interviewees mentioned various sources of inspiration for their app design. For instance, interviewee I1 described that they hung a board on a wall, on which they put all kinds of information they gathered during the process. For instance, they created personas that formed a source of inspiration by constantly imagining what the personas would do and how they would do that. According to the interviewee, this helped in getting empathy with the end user. This interviewee also looked at artifacts that the target user might use to become inspired. Interviewee I2 also mentioned that empathising with end users helped in getting inspired. This interviewee mentioned walking around trying to imagine what the end user would do in order to come up with ideas. Interviewee I4 noted that the source of inspiration really depends on the thing they are working on. This interviewee told to sometimes take inspiration from nature when working on visual design or to take inspiration from, for instance, books and theories. Interviewee I5 mentioned to get inspiration from talking to people and listening to them. This interviewee also mentioned that to become inspired in daily life by watching people doing or using certain things and asking their opinions about and experiences with it.

One source of inspiration that all interviewees mentioned is the work of others. Four of them even mentioned this before explicitly asking them about this. The interviewees looked at various types of works of others, such as apps and websites (e.g., retail websites and online platforms for designers). Interviewee I2, for instance, searches to find examples of what is needed at that moment. Most interviewees said that they looked at the works of others to get inspiration, to learn something, or to see what already exists on the market. Interviewee I3 mentioned looking at what already exists, picking out some aspects and trying to learn from that. This interviewee mentioned that it mainly depends on how good they are at something if they need to learn from others. Interviewee I4 tries to adapt "patterns that are widely used in the digital landscape" to the current context of the project. Interviewee I5 described actively asking the following questions when looking at the works of others: "What do you think that this company was trying to achieve?" and "If they did that and if they try to make you feel a certain way, what would we try to do?". This interviewee tries to understand the thought processes of the one who has created the work when assessing that work. This interviewee also described a "zoom out process" in which they start with visiting and analysing physical locations and then proceed with analysing digital works to become inspired. They do not do this to copy elements, but to get inspiration. This interviewee added that the creative ideas do not always instantly arise, but that they develop in the back of the mind. Interviewee I1 mentioned that they once literally transferred one element of the work they looked at to their own design, since they thought it looked "clean".

In general, the interviewees mentioned that they also look at the works of others that are not direct competitors. Interviewee I4 searches for general patterns as many things have already been done in different fields. According to this interviewee, those things are widely known and can be seen as best practices. Interviewee I3 also mentioned that many things are already done somewhere. Remarkably, both interviewees gave the same example: the on-boarding process. Two other interviewees (I2 and I5) also mentioned that some things (e.g., the burger menu) are common practice and that doing those things differently could lead to frustrating the end users. Four interviewees mentioned that they sometimes download apps to see and analyse how they work. Only interviewee I2 mentioned to frequently make use of app reviews in an app market.

This interviewee assessed reviews to see what is good and not good in apps and tries to use this knowledge when designing the app. Interviewee I4 mentioned to sometimes, but not often, look at the number of ratings and the number of stars of an app. One of the interviewees (I3), who told not to look at app reviews, gave the reaction that it could be a good idea to look at app reviews as end user input is very important for this interviewee.

Shneiderman (2007) mentioned that people may learn from the works of others and may use those works as a starting point for their own. This was also observed in the interviews. All interviewees mentioned to look at the works of others to become inspired and to learn something, but not to directly copy ideas per se. Moreover, various interviewees mentioned not only looking at the works of direct competitors, but also at works in other domains. This suggests that both near fields and far fields are consulted. Thereby, this indicates that not only cased-based design, but also analogical design (Goel, 1997) (as described in Section 3.3.2) is applied in practice. Moreover, various interviewees noted that there are also common ways of doing things and that changing those may be a risk. Also here, the expression of "preventing to reinvent the wheel" may apply. Another important source of inspiration is the end users. They may both directly and indirectly lead to the generation of new ideas.

4.1.3 Implicit or Explicit Element

When asking whether creativity was an implicit or explicit part of their app design processes, all interviewees at least explained that it happens implicitly. However, interviewees I2 and I5 additionally said that they also gave explicit attention to it in their projects. Interviewee I2 really tries to be creative and to do things differently than others. However, this interviewee said that this can be difficult and that "you cannot force creativity". I2 also told that by trying to do things differently, you may change standard guidelines and risk that end users become frustrated as a result of that. Therefore, this interviewee tries to find a balance in doing things differently and in getting inspiration from others. The other interviewee mentioned that "you do have to be consciously creative, but at the right moment". This interviewee told to become subconsciously inspired and that when they have to generate design ideas, they need to start being creative. Two other interviewees stated that creativity was something that "just happened naturally" and that they "did not focus on it". The last interviewee said that they already have many ideas and that prioritisation is most important. Thus, the creative app design process also involves both conscious and unconscious thinking, just like the general creative process described by Wallas (1926) (as cited in A. Cropley, 2006 and Rhodes, 1961). Moreover, already in these first three sections, it can be observed that the first three phases of Wallas' model also seem to apply to the app design process. In general, interviewees start with gathering information and obtaining knowledge. Then, this information is processed consciously and/or unconsciously. From this, the ideas emerge. Some of the interviewees even mentioned to recognise an "aha moment" (Shneiderman, 2007) in which suddenly an idea emerges.

Overall, the interviewees only discussed few different creativity techniques. One technique that was mentioned by four interviewees is brainstorming. Interviewee I4 did not explicitly mention this technique, but described techniques such as workshops and ideation sessions, in which ideas are generated with the team and the client. Interviewee I5 also mentioned having done a workshop session with end users to become inspired. As already mentioned, Interviewee I1 used a board that was hung on a wall to generate ideas. Some mentioned that they also made use of drawing or sketching to come up with ideas.

4.1.4 Creative Ideas

Each interviewee was asked to give an example of one of their creative ideas. Interviewee I1 came up with an example involving the end user could reach each page in the entire app in just three clicks. This interviewee mentioned that this was a good idea, since it considered the context of the end user and since it was simple. Interviewee I2 described the example of a creative idea as the way in which large surveys were implemented in the app. The team found a way in which those surveys could be easily captured in the app. This interviewee told that the initial survey was complex, but that the solution led to a surprised end user. Interviewee I3 described the creative idea as the flexibility that the app provides. The interviewee described that the format used in their app to represent a specific part of the service, supports the needs of a wide variety of users. Interviewee I4 described the example of a creative idea as they way in which data could be filtered. This interviewee told that the problem that needed to be solved was also quite complex. I4 used inspiration from a different domain to find a way of easily filtering the data. The interviewee told that it was important that end users should be able to "search quickly and extensively" and that "the learning curve should be low". Interviewee I5 described described the example of a creative idea as the way in which multiple types of information could be captured in one graph in such a way that it was easy to understand for the end user.

When assessing these creative ideas, it can be noticed that each idea is related to some quality. Qualities that were mentioned or that could be deducted from the examples given comprise simplicity, satisfaction, flexibility, quickness, and easiness to understand.

4.1.5 Importance of Creativity

The importance of creativity for or within a company seems to differ per company. Interviewee I1 told that in the previous company, it was more important to be creative, because they had to deliver prototypes within one week. This forced them "to think outside of the box" and to solve problems quickly. However, ideas did not need to be innovative and wishes of the end user and their processes were the most important motives. Interviewee I2 also stated that the importance of creativity depends on the project, the sector, the persons, and the stakes of the project. This interviewee mentioned that, for instance, sometimes ideas may only be creative to some extent and may not always be too radically changing. I2 told that depending on how much there is at stake in a project, ideas may be more or less creative. Interviewee I3 mentioned that creativity and innovation are important within the company for staying ahead of competitors. The company tries to be an innovator and that drives the generation of new ideas. Interviewee I4 also mentioned that being creative is important in the company for, for example, delivering innovative products. Moreover, in that company, knowledge from various sources needs to be applied and existing products need to be taken to a next level. This interviewee also told that many decisions need to be made and that this requires creativity. Interviewee I5 mentioned that the company is not concerned with being different, but with trying to achieve simplicity in the products they deliver. This interviewee told that in their strive to achieve simplicity, creativity is highly necessary and innovation will come with that strive.

It seems that, overall, the interviewees see creativity as an important aspect in the app design process. Creativity is here not per se related to the creative product, but also to the creative process, which is in line with two of the four 'Ps' of creativity of Rhodes (1961). A strive for innovation is not a prerequisite for creativity here.

4.1.6 Difficulties & Challenges in Creativity

Each interviewee mentioned to have experienced some difficulties or challenges in being creative. However, the nature of the difficulties or challenges varied per interviewee. Interviewee I1 mentioned that the project team consisted of only members who each had a strong opinion. The interviewee noted that on the one hand this could be an advantage, because it leads to critical reflection of ideas. However, oftentimes team members have their own view on things, making it harder to make decisions. I1 noted that this did not hinder the generation of creative ideas. This challenge was handled by both compromising and by showing new ideas to individual team members to prevent lengthy discussions. The interviewee also noted that it could have helped if the team would have been more multidisciplinary.

Interviewee I2 noted that politics can make the creative process more difficult. This interviewee gave an example that some managers may not always accept ideas that are proposed by the design team or that some ideas are only accepted when they hear it from the end users directly. This interviewee noted that it is a challenge that many stakeholders have different stakes. Another challenge I2 mentioned was a "writers block" which entails getting stuck in coming up with new ideas. This interviewee tries to solve this by either working on a different task or by drawing on paper. Interviewee I5 also mentioned to sometimes get stuck in coming up with solutions and that this may last for weeks or even months. This interviewee explained that it helps to "take a step back" and do some testing with end users. I5 noted that it is essential to keep listening and keep doing research, because this helps in gradually moving towards a creative solution. This interviewee added that a good team is needed in the creative process and that it essential that everyone keeps in mind that the solution is not created for themselves.

I3 did experience difficulties in the creative process. However, the main challenge for this interviewee was the translation of the generated ideas into specific functionalities. This interviewee mentioned that it could be difficult to add new functionalities while keeping the app as simple as possible. Another challenge I3 mentioned was the design of the user interface as the team had little experience with it. To cope with this, they make use of existing frameworks. Interviewee I4 experienced difficulties by weighing many aspects. For this interviewee, one way of handling this is structuring the process and creating an overview. Also drawing and sketching also helps to cope with this. Other aspects that have influence on the creative process that this interviewee mentioned are the client's attitude towards user experience (UX) and the client's branding. These aspects may impose boundaries on the creative space.

As already mentioned, the nature of the difficulties in the creative process differs. This is not surprising, since there are many factors that may influence creativity (Section 3.2). Some of the factors that were for instance mentioned by Amabile et al. (1996) and Dallman et al. (2005) also apply to the app design process. These include (but may not be exhaustive): organisational encouragement, supervisory encouragement, group dynamics, and stakeholder conflict.

4.2 App Design Practice

The second part of the interview focused on the app design process. Interviewees were asked to describe what their app design process looks like in general or looked like in a specific project.

4.2.1 App Design Process

In general, it can be observed from the interviews that the app design process is very context dependent. Each interviewee was asked to describe which steps were taken during the app design process. The description each interviewee gave, differed from those of the others. Moreover, some interviewees also mentioned that the app design process also differed per company and project. The description of one of the interviewees (I3), however, stood out. The reason for this was that the app created was for the interviewees company and that first a valuable working product needed to be delivered. The company initially did not have any customers and therefore, the end users could not be consulted in the beginning. As the company started to grow, more and more end users could be consulted.

The process descriptions of the other four interviewees partly overlapped. All of them mentioned to gather information in the beginning of the project. Each interviewee mentioned that the client is involved here in some way. Sometimes this is done to determine project goals and other times it is done to elicit customer wishes and needs. Interviewee I1 mentioned to have created personas after this initial step. Interviewee I5 mentioned to "distil" insights from the initial user interviews and use those to create design tasks for the design team. The other two, the UX designers, started working on sketches after this initial step. Other steps that were further mentioned included creating mock-ups, lo-fi prototypes, and hi-fi prototypes. Three of them (I1, I2, and I4) explicitly stated to make use of prototypes. The other mentioned to draw screens to test those with the end user. Each interviewee tried to consult the end user or the client in between the steps to gather feedback for further refining the design. Three of the interviewees (I2, I4, and I5) mentioned that they did or tried to do some user testing. These final steps seem to describe the fourth step of Wallas' model of creativity (Wallas, 1926), since feedback is asked and testing is done to verify the ideas. Moreover, the ideas are implemented in prototypes and the working app.

Interviewee I2 mentioned that the description of the app design process, is a process that is striven for but that it is not always the case that it happens like that. Two interviewees (I2 and I4) mentioned that most ideas are generated in the beginning of the process. The reason they gave for this is that in the beginning few things are fixed, while in the end, less aspects may vary. However, interviewee I5 noted that the generation of ideas can be seen as a continuous flow with ups and downs. This interviewee mentioned that it seems that there is a linear relation between the number of problems to be solved and the amount of creativity needed.

4.2.2 Methods, Frameworks & Tools

Four interviewees stated that they worked in an Agile way. One interviewee noted to work with Scrum and another mentioned to use a combination of Scrum and Prince2. Yet another interviewee worked with design sprints and that they also adhere to their own design framework. This interviewee mentioned that the use of methods and frameworks depends on the specific client. As already mentioned, one of the interviewees mentioned to make use of a design framework for the interface design. Only one of the interviewees mentioned not to use any method or framework. This interviewee noted that the design team they collaborate with may perhaps use those. None of the interviewees mentioned to specifically apply Design Thinking. However, as can be observed, various DT principles are applied by the interviewees. For instance, it is found to be important to involve the end user and some even try to empathise with them. Moreover, prototyping is used and various iterations are done during the process. Finally, collaboration

also seems to be a vital aspect of the process. These are all characteristics of DT (Brown, 2008; Tschimmel, 2012).

The interviewees listed various tools that are used during the app design process. The tools that were discussed were: a mock-up tool, Bizagi Modeler, Mendix, Adobe XD, Figma, Proto.io, Jira, Trello, Sketch, photoshop, Abstract, InVision, Principle, and Google Drive. Most tools were only listed by only one of the interviewees. Some interviewees also added pen and paper to the list. Interviewee I5 also did not mention any specific tool that was used during the app design process.

4.2.3 Stakeholders

Interviewees already referred to some stakeholders prior to the specific questions about stakeholders. The most frequently recurring stakeholders were clients and end users. It depended on the project and company that was discussed what these roles look like. Sometimes the distinction between these two was even blurred, because the client may be or may represent the end user. Examples of client stakeholders that were discussed during the interviews include: managers, employees working in the field, directors, and a finance department. Interviewees also numerated various internal stakeholders, such as a design team, managers, a development team, and consultants. It differed per interviewee and project that was discussed how many stakeholders were involved and who those stakeholders were. It should be noted that not all stakeholders were mentioned by each of the interviewees. For instance, the development team was not always mentioned as a stakeholder or was either mentioned as an internal or external stakeholder.

Most of the times, the stakeholders were consulted during the projects. Oftentimes, the final decisions regarding the functionalities and qualities of the app were made by the team in which the interviewee was working. However, in some cases, managers or the client had the final say. All interviewees mentioned that they try to involve the end user during the app design process in some way. They all stated that it is valuable to involve the end user. Interviewee I4 mentioned that they try to involve the end user as much as possible and added the following to this: “We are the lawyer of the end user”. This interviewee also mentioned trying to consider the perspective of different stakeholders, while also trying to use their own expertise. Interviewee I5 tries to involve a diverse set of end users from different ages and backgrounds, including non-target users. Interviewee I3 mentioned that it is important to listen to end user “who is willing to think about your product”. The second interviewee noted that the degree of user involvement differed per project. For one it was not really needed and for another project is was more difficult to get it done. Interviewee I1 also mentioned that it differed per company how much end users could be involved. Examples of ways in which the interviewees involved end users in the projects included: interviews, a workshop, observations in the field, and testing.

4.2.4 Relation Company and App Design Process

One aspect that could be observed over all the interviews is the fact that the app design process and creativity in that process differed per project and/or per company. Also here, various factors that may influence creativity that were described in Section 3.2 seem to apply. One factor that two interviewees mentioned was the size of the company there worked at. One of them (I3) said that working at a start-up had a positive influence on the app design process. Having contact with the end user is easy for them and developments can be introduced quickly. The

other interviewee (I4) mentioned that being a smaller company may be less of an advantage. This interviewee mentioned that smaller companies may need projects to survive, whereas the bigger companies may be seen more as a real entity by clients. Expressing an opinion towards the client was easier in the latter case for this interviewee. This interviewee also mentioned that the types of clients and stakeholders differ per company. Interviewee I4 mentioned to experience that the open culture of a company has a positive influence on the generation of ideas, as others working there are approachable. The first interviewee mentioned that the one of the companies this interviewee worked at was user interface and user experience oriented, which made that it was natural to consult the end user. This company also provided freedom towards its employees, resulting in the fact that the interviewee was not constrained in coming up with ideas. Interviewee I5 described that the culture in the company is that "nothing is impossible until proven otherwise". This helps this company in keeping exploring and being creative.

Finally, as already mentioned, the project also influenced the app design process. Interviewee I1 mentioned that they had to create an app from scratch and that main objective was to deliver a functioning product. Therefore, some of the creative ideas were not implemented. The fact that they had to decide on a scope restricted this interviewee in a sense. Interviewee I4 also mentioned that the processes differed between starting from scratch or starting from an existing product. Moreover, this interviewee noted that the type of clients and their branding also seems to influence how the process happens and which methods are used.

4.3 Main Takeaways

- In general, the app design process and the role creativity has in this process differs per project, company, and per person.
- Ideas seem to emerge in various ways. However, these interviews indicate that ideas oftentimes emerge from interacting or empathising with the end user.
- During the app design process, the designers have various sources of inspiration. All interviewees had in common that they looked at the works of others to become inspired.
- Creativity mainly takes place implicitly. Creativity techniques are mentioned, but are not extensively used by all interviewees.
- It was observed that each interviewee seems to value creativity as important. This may be for coming up with novel solutions or for creatively handling the design process.
- Interviewees experience difficulties in the creative process, but the nature of these difficulties varies.
- The steps taken during the app design process are also context dependent. However, common steps that could be identified are gathering project information, sketching, prototyping, testing, and gathering user feedback.
- A variety of methods, frameworks and tools are used by the interviewees. Most of them mentioned to work in an Agile manner.
- The stakeholders in the app design process differ per project and company. However, the two most important stakeholders mentioned are the end user and the client.
- The four-stage creative process as described by Wallas (1926) also seems to be applicable to the app design process.

Chapter 5

Conceptual Design

In Section 3.5 was discussed that already many CSTs exist. Nevertheless, this research focused on the design of a new CST. CSTs targeted at a specific domain are suggested to be more effective (Shneiderman, 2002). To the best of our knowledge, currently no CSTs exist that are specifically created for the app design field. Furthermore, CSTs enable users to make discoveries and inventions (Shneiderman, 2007). This is a valuable resource for app designer in their attempt to differentiate their apps from those of others. Hence, we argue that this field may benefit from a CST.

The main goal of the eventual tool is thus to foster the creativity of app designers, so that they may find a way to differentiate themselves. We pose that an app is creative when it does something different from or new compared to either all existing apps or existing apps of the same type, whilst at the same time being useful or valuable for the end user. As described in Section 3.1, Boden (2004) makes a distinction between P-creativity and H-creativity and notes that H-creativity is more difficult to achieve. In the app design field, there are already generally accepted ways of designing apps (Chapter 4). This means that also in this field it may be more difficult to come up with historically new app design ideas. Therefore, we strive for fostering P-creativity.

R1. *In essence, P-creativity is a must-have effect of the envisioned tool and H-creativity is only a nice-to-have effect.*

In the context of the app design domain, potentially another term may even be coined, namely "*app-creativity*". This type of creativity may entail that an app can be creative compared to apps of the same type, while this may not be creative compared to other types of apps. Ideas that are commonly applied in one type of app could possibly be highly novel when applied, may it be in an adjusted way, to another type of app.

It must be kept in mind that when trying to foster creativity, merely striving for generating new ideas is not enough. The second aspect of creativity (i.e., usefulness or value) should not be overlooked, especially in a field where the product needs to comply to requirements and user needs. Since the creation of this tool is rooted in the RE field, it is essential that the tool should help in generating new requirements. The tool should not necessarily be a source of inspiration for the implementation or appearance of functionalities, as already many of those resources can be found on the Web. However, the tool could still help app designers with that. Therefore, the second main tool requirements is as follows:

R2. *The tool must form a source of inspiration for new requirements for both already existing apps and new apps.*

The interviews reported in Chapter 4 indicated that app designers look at the works of others to find inspiration for their own work, to identify common practices, and to learn from others. This

insight was used as one of the starting points for the design of the CST. In short, the CST will be a tool that shows app designers examples of well-functioning apps that are relevant for the design task at hand. The tool may become an application filled with references to real-world examples (i.e., apps) of good design practices. As described in Section 3.3, analogies are important in creativity and design and may help to find solutions to design problems. However, it was also described that making analogies may be difficult. Therefore, the following requirement for the tool was identified:

R3. *The tool should guide app designers in making analogies to find creative solutions to their own design problem.*

The knowledge of app designers will be expanded by making use of the tool. As could be read in Section 3.2, creativity can be fostered by obtaining more knowledge. By assessing the examples that are presented, app designers will ultimately be inspired with new ideas that could solve their own design problem. The eventual CST can be seen as a combination of two types of tools for innovation, namely 1) tools that will broaden designers' knowledge base and 2) tools that help people in identifying relations between domains (Markman & Wood, 2009). The tool can be seen as a *running shoes* type of CST (Nakakoji, 2005) that was discussed in Section 3.5. The CST is intended to improve the app design process, but the app design process may still function without the tool.

The app field is big and diverse. Different app designers may have different design problems to solve, leading to the last main requirement for the tool:

R4. *The tool must cover a wide variety of apps to help solving a wide variety of design problems.*

The challenge in this research is to *find meaningful example apps that will foster creativity and at the same time will prevent or limit design fixation.* This chapter and the following two chapters will elaborate in more detail on this challenge and our proposed solution to it.

5.1 Important Concepts

In Chapter 3, a variety of concepts was discussed that play a role in creativity and/or in RE. A selection of those concepts serves as the foundation of the envisioned creativity support tool. The main concepts that serve as the foundation for the tool are **analogical reasoning** and **app review analysis**. The application of the concepts discussed in this section will contribute to handling the aforementioned challenge. Furthermore, the theory behind the tool contributes to the existing body of knowledge of multiple disciplines by combining these concepts into one design. An overview of the relation of the concepts and their expected influence can be found in Figure 5.1.

The main concept around which the design revolves is analogical reasoning, or to be even more specific: **design-by-analogy**. Analogies help in solving problems (Christensen & Schunn, 2007) and in altering perceptions, which are important in creativity (Boden, 1996, 2004). Also, Dahl and Moreau (2002) argue that the use of analogies has a positive impact on the originality of design solutions. As described in Section 3.3.2, design-by-analogy entails solving design problems by assessing solutions to analogous design problems. By showing examples of analogous design problems, app designers may be guided in finding a solution to their own design problem. As

mentioned at the beginning of this chapter, the tool will thus present examples of well-functioning apps to app designers. These example apps embody solutions to certain design problems, which may help app designers in finding a solution to their own design problem. Using examples is inherent to design-by-analogy (e.g., Chan et al., 2011). However, presenting examples needs to be done with care as it could lead to **design fixation**, which could have a negative influence on the creativity of the ideas (Section 3.4). Again, app designers mentioned to already look at the works of others (i.e., examples) to solve their own design problem. Applying design-by-analogy aims at guiding this process to foster creativity while preventing a possible design fixation.

The tool should cope with design fixation in two interpretations of the concept. On the one hand, the tool needs to expand the exploration space of the app designer by proposing new ideas to consider. Thereby, the knowledge and expertise of the app designer is expected to be enhanced, which is positive for creativity (Amabile, 1983; Boden, 2009). On the other hand, the tool needs to be designed in such way that it is prevented that app designers copy elements of the examples that are presented to them. The latter is important to make sure that the designs of the app designers and are truly creative and that they do not infringe the copyright of the creators of the example apps. It was explained in Section 3.4.1 that **analogical distance** may influence design fixation and the originality of designs. Therefore, the tool will work with the notion of different distances between the example apps and the app of the design problem. The choice for and application of these concepts will be discussed in more detail in Chapter 7.

The other main concept on which the design of the tool is based is **app review analysis**. The main reason for making use of app reviews is incorporating the opinion of end users about apps. This reflects the human-centredness principle of Design Thinking. The user reviews are analysed to give app designers more insight into app features about which users have a positive or negative opinion. These features will be shown to the app designer as an additional source of inspiration. Pagano and Maalej (2013) suggested that user reviews can be used as a potential source of inspiration, as users express their ideas regarding the app in them. They also argue that incorporating user feedback is essential and useful in software development and RE. As discussed at the beginning of the chapter, ideas must be useful or appropriate for them to be creative. This is especially important in the app design domain, since the app must comply to user needs, expectations, and wishes. These needs and wishes can be found in user reviews (Pagano & Maalej, 2013). Finally, applying app review analysis will form a bridge between disciplines such as app design, RE, and creativity research. As described in Section 3.6.1, app reviews have become a topic of interest in RE research. We aim at finding out whether this rich source information may be used to foster creativity.

5.2 Design Approach

During the design process and during this research, we decided to follow a bottom-up approach instead of a top-down approach. In this way, a better overview could be maintained and choices could be substantiated more easily with empirical evidence. In taking this approach, a toy design problem was selected. In later iterations, the findings for this specific problem can be generalised to other design problems. The toy design problem that was selected to begin the research with is the design for a *recipe app*. This specific topic was chosen, because it is an everyday topic to which many people can relate. Furthermore, we think that this specific design problem would allow for creative ideas regarding functionalities and requirements.

The objective of this research was to deliver a first proof of concept. As discussed in Sec-

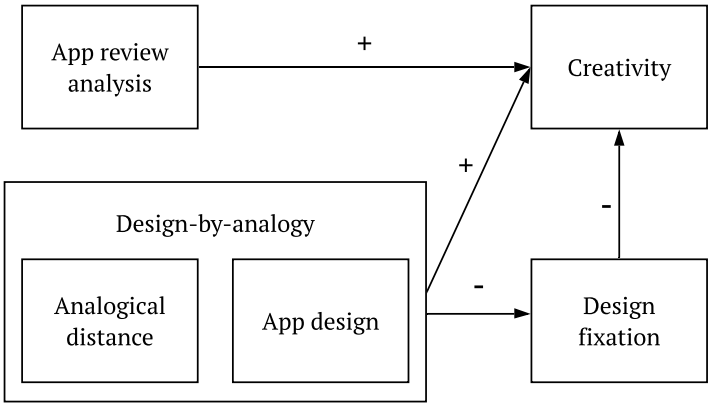

Figure 5.1: Concepts central to the conceptual design and their expected influence.

tion 2.1, the initial idea was to create a first prototype. However, the focus shifted to the development of a solid design theory. This means that first theory behind the design was validated before proceeding to the creation of a prototype. Also, this approach had the objective to ensure that the tool would actually be able to fulfil its main purpose.

In this research, the scope was confined by focusing on apps that were present in the Google Play Store[1], as this is one of the biggest app markets. Furthermore, mobile games were not considered, since it was reasoned in line with Johann et al. (2017) that games are different from other apps. Furthermore, the analyses in this research only focused on apps that were downloadable for free. The intention is to only show free apps in the tool, so that designers are able to download and inspect those apps.

5.3 Evolution of Design Ideas

During the design process, a variety of alternatives was considered for the design of the tool. The ideas evolved continuously in an organic manner as more empirical evidence was gathered and more knowledge was obtained. Describing all small changes is not the main objective here and would also not be feasible. Therefore, this section describes the main alternatives considered during the design process.

Already early in the design process, it was decided to use analogical reasoning and app review analysis for the aforementioned reasons. The main challenge was to find a way to form analogies and to determine analogical distance. Ideas started with using functionalities, qualities, and services. Earlier research showed that these former two concepts could be extracted from user reviews (e.g., Groen, Kopczyńska, Hauer, Krafft, and Doerr, 2017; Johann et al., 2017). We believed that these could be used to form analogical relations and to foster creativity. The idea was for the app designer to create a query comprising functionalities and/or qualities and a domain descriptor. Services could potentially be used to denote the analogous domain. Also, we reasoned that app domains could for instance be formed by clustering apps with similar features (cf. Al-Subaihin et al., 2016). The idea was for the tool to return apps from analogous domains

[1] https://play.google.com/store/apps

with the same qualities and/or similar or the same functionalities. The main problem with using these concepts to find analogous apps, is that the app designer already needs to have identified those before using the tool. This means that the tool would only be usable in a later stage of the design process. Also, when the app designer would search for qualities and functionalities in analogous domains, the tool would mainly serve as an inspiration for the implementation and not for new functionalities and requirements. Thus, the tool would not or only to a limited extent help in broadening the app designers' horizon. Also, after some initial analyses (Section 6.1), it seemed that automatically extracting services would be difficult.

Thus, it was needed to find a way in which analogous domains could be found in the early stages of the design process. Using the categories of the Google Play Store for depicting analogous domains would not be a workable solution, since the quality of the categorisation varies widely per category and even per app (Al-Subaihin et al., 2016). It was determined that a manual approach for finding analogous domains would be more suitable. It was considered to let humans describe apps in terms of a set of variables, which could be used to create clusters of apps. However, this approach resulted in two problems. Firstly, suitable variables that could be used to describe all apps needed to be determined. Secondly, letting app designers fill in many variables when using the tool was not expected to be not user friendly. Furthermore, expecting the app designer to know the values for the variables in advance would be conflicting with the objective to create a tool that is usable in the early stages of the design process. Conducting a workshop in which participants would perform the analogical reasoning and would classify apps as either near-field, middle-field, and far-field apps was also considered. However, it may be difficult for humans to perform analogical reasoning on the spot (e.g., Gick and Holyoak; 1980). Also, this approach would not scale well to many design problems. Eventually, the idea came up to use goals and keywords to find analogous apps. Using VerbNet (Kipper Schuler, 2005) was briefly explored for supplementing the goal identification. It was observed that many verbs used to describe goals are member of large verb classes (see e.g., "inspire"). It further seemed that many verbs within the same class may not share the same meaning as the ones specified in the goals. Furthermore, the meaning of verbs can differ per context (see e.g., "find"). Determining the right main class for a specific context would be challenging. All in all, we expected that the precision of using VerbNet would be too low for our purposes without heavy processing measures. The app reviews were still used to find functionalities and qualities, but to determine what people like and do not like about apps as an additional source of inspiration or guidance.

5.4 Current Design

App designers need to generate ideas (i.e., functionalities) for the app they are working on. It is a precondition that app designers have already identified some high-level requirements (e.g., user needs and wishes). The app designers need to translate these preconditions to high-level **goals** (Figure 5.2). These goals are already prespecified so that app designers only have to select the goals that meet the high-level requirements best. Besides that, app designers need to insert a set of **keywords** that describes the app they are working on. This approach adheres to requirement two (R2) described earlier in this chapter. The designer only has to specify high-level requirements and keywords, making it suitable for both existing and new apps. The tool then returns a set of **analogous apps** together with **word clouds**. These apps all solve a problem that is similar to that of the app designer. We hypothesise that by presenting a set of apps that are structurally similar and on the surface dissimilar, app designers will start

reasoning about why these apps are relevant for their case. We reason that the app designer will thus make mappings and inferences when assessing the presented apps. Ultimately, the app designers will, as a result, come up with creative ideas and requirements for the app they are working on. In this way, the tool guides app designers in analogical reasoning, thereby fulfilling requirement three (R3) listed at the beginning of this chapter. More detail about how analogous apps are found and why the app designer needs to provide keywords and goals shall be given in Chapter 7. Since examples are shown of analogous apps, we believe that the ideas the designer may generate may be analogous to those of the examples. It could also be the case that the app designer combines ideas seen in various examples, leading to combinational creativity (Boden, 2004). Example ideas may also be slightly changed and placed in a new context or domain (i.e., transformational or exploratory creativity). Therefore, we believe that the tool at least fosters P-creativity, which is in line with requirement one (R1).

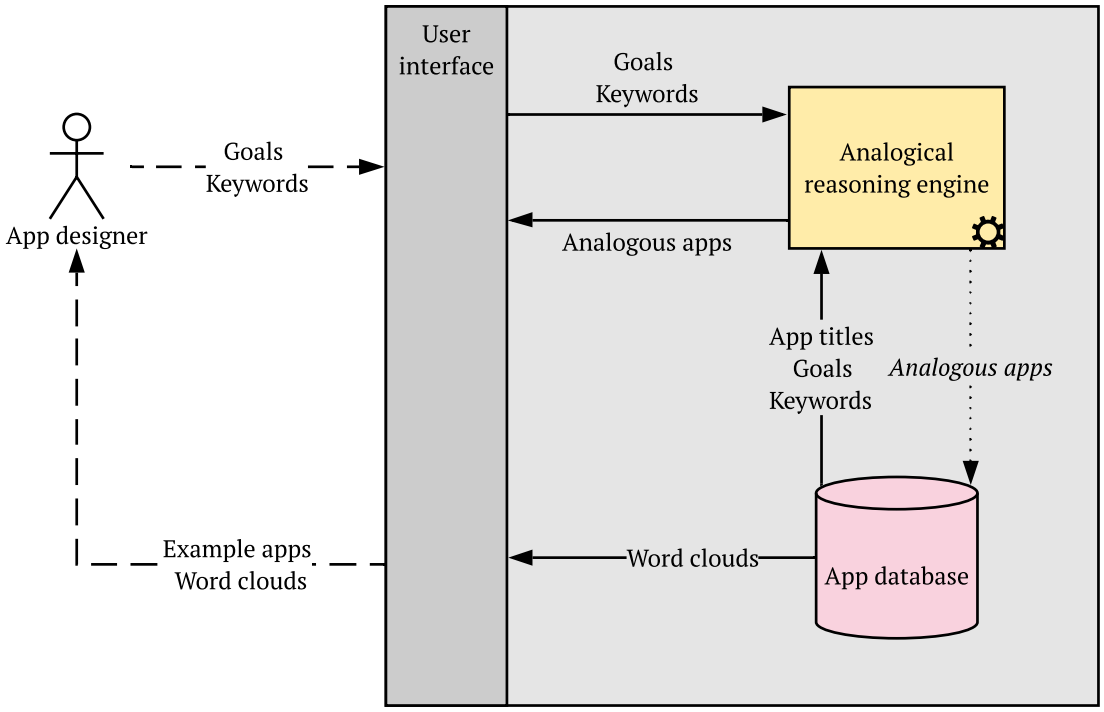

Figure 5.2: Overview of the usage and architecture of the tool.

The word clouds are provided to the app designer as a source of information about the analogous apps. These word clouds aim at helping the app designer better understand the app by showing features about which users have expressed their opinion. The app designer will both see features about which users expressed positive opinions and features about which users expressed negative opinions. In essence, these word clouds show solutions to the design problems of the analogous apps. They may thereby further guide designers in finding a solution to their own design problem. We hypothesise that the word clouds will contribute to the fostering of the app designer's creativity, as those provide new knowledge to the app designer (Amabile, 1983; Boden, 2004). Furthermore, an example app may be seen as a creative combination of ideas

(i.e., features). By evaluating these combinations in the word clouds, one may also expect that the creativity will be fostered (Boden, 2009).

The current approach comprises both human and machine processing. Humans are to date still better at analogical reasoning (Markman & Wood, 2009) and creativity (Boden, 2004). Therefore, incorporating human processing in the design is a natural choice. Still, machine processing is needed to make the design feasible. As could be read in Chapter 1 and Section 3.6, there are many apps and large amounts of app reviews. Performing a manual analysis on apps and app reviews would therefore be unfeasible. Applying machine processing ensures that the tool could incorporate a wide variety of apps and app reviews, thereby fulfilling requirement four (R4).

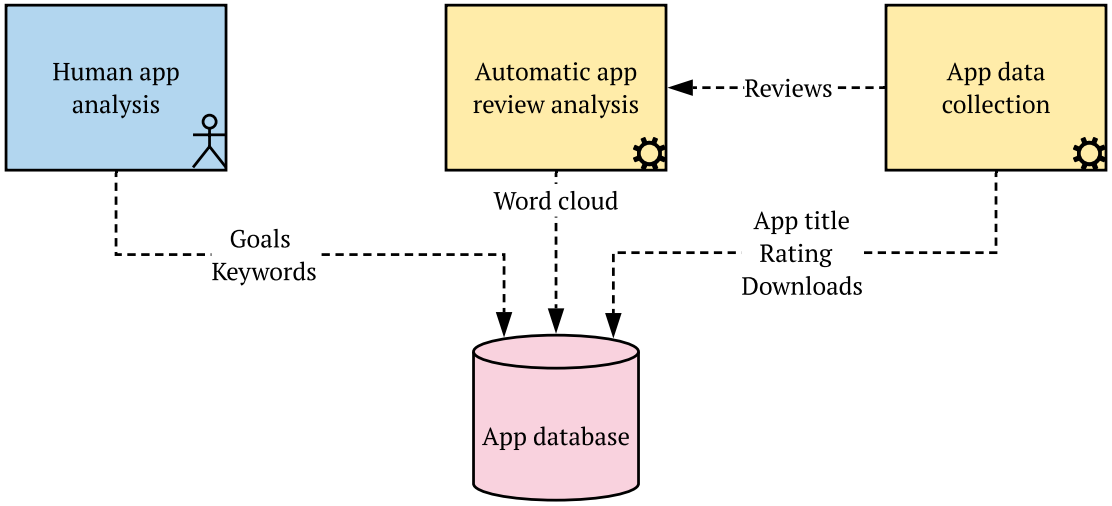

Figure 5.3: Overview of the approach behind the tool.

The division of human and machine processing can be seen in Figure 5.2 and Figure 5.3. Figure 5.3 shows that human annotators will provide goals and keywords for a set of apps. App data collection and app review analysis are performed automatically. Figure 5.2 shows that the human input is used to find analogies. This human input is combined with the gathered data to select appropriate example apps, which in turn can be assessed by the app designers. The gathered data such as ratings and number of downloads are used for selecting high-quality apps that are not too common (see Chan et al., 2011). The exact preconditions for selecting apps shall be discussed in Section 7.4.2. When using the tool, app designers are required to perform some sort of analogical reasoning to come up with analogous solutions to their own problem (Figure 3.2). The analogical reasoning component is mainly human oriented. However, machine processing is still needed to process a large number of apps (see yellow component in Figure 5.2). In the following two chapters, the two main strands of the design are described and explained in more detail.

Chapter 6

App Review Analysis

There are several reasons for incorporating app reviews in the design of a creativity support tool for app designers. First of all, app reviews can be seen as a rich source of information (Maalej & Nabil, 2015; Nagappan & Shihab, 2016) for app designers from which they can get insight into the users' opinions, requested features, bugs, and more (Pagano & Maalej, 2013). Secondly, this research is guided by some principles of Design Thinking. The DT principle of human-centredness (Brown, 2008; Tschimmel, 2012), also inspired the use of app reviews. App reviews express the experiences, thoughts, and opinions of the end user of apps. Given the human-centred approach, these reviews can represent many end users. In this research, we were mainly interested in finding out how this rich source of information can be aggregated and used to foster the creativity of app designers.

There is a vast amount of user reviews in the various app markets (Wang et al., 2019). Also, many apps receive over thousands or even millions of reviews. Analysing these reviews and extracting features manually is therefore not a feasible option. Therefore, as already discussed, an automatic approach is taken to analyse app reviews and extract features and opinions.

6.1 Manual Analysis

In order to get more insight into the nature of app reviews and to select the most optimal method for automatically processing the app reviews, manual analysis of app reviews was conducted first. This manual analysis was done in two phases: 1) an initial analysis to get some insight into the nature of app reviews and 2) a more in-depth analysis to find more empirical evidence to base further decisions on. Both analyses were conducted in spread sheets.

6.1.1 Initial Manual Analysis

The initial manual analysis was conducted on a set of six apps from the Google Play Store. The selected apps came from five different categories. Two apps came from the same category to compare the reviews within one category. The apps differed in number of reviews (i.e., ranging from tens of thousands to hundreds of millions), number of installs (i.e., ranging from over one million to over ten million), and number of stars (i.e., ranging from 2.4 stars to 4.6 stars). These criteria were used to see whether reviews different between app with different characteristics. It must be stressed that this was not a formal analysis and that it was not the objective to extract analysable, quantitative results. This analysis was highly exploratory in nature, with the intention to obtain more insight as a starting point for further research.

Approximately ten reviews with different lengths and ratings were selected for each app. The text of each review was annotated with different colours. Each colour represented a different

type of topic, which was based on our own observations and partly on the topics identified by Pagano and Maalej (2013). The list of topics was supplemented if additional ones were found during the annotation process. This forced us to perform some iterations. It was mainly of interest in this analysis to find out how end users express their opinion about certain features of an app. In general, there were two types of labels, namely 1) elements of the app and 2) broader topics of discussion. The former included *app functionality*, *visual component of an app*, *quality or opinion*, and *app service*. The difference between the app functionality and visual component was that functionalities are something that the app or the end user can perform (e.g., logging in or searching), while the component is the visual aspect on which the functionality is performed (e.g., a user account or a search bar). The latter types of labels included *feature request*, *problem report*, and *referral to another app*. Those topic types were not interesting for the final analysis, but were still assessed at this stage to find out how extensively they are present in reviews and how they differ from the review parts that were of interest. As a final step, functionalities and qualities were roughly extracted to find out how those are expressed by users.

From this initial analysis, some observations were made that were useful for the final analysis. Most findings were in line with literature. First of all, the quality of the reviews differs considerable between reviews. Pagano and Maalej (2013) also observed this issue. Secondly, most reviews cover multiple topics. Most of the topics are not interesting for the final analysis and thereby introduce noise into the data. In Section 3.6.1 was discussed that in some researches (e.g., Gu and Kim, 2015) sentences covering an irrelevant topic were filtered out when analysing app reviews. Perhaps unsurprisingly, functionalities and qualities are expressed in different forms. The same functionality is often expressed in different ways, see for example "a house search" or "searching for a house". Functionalities are not only expressed as nouns, but also as, among other things, verbs. Qualities are not only expressed as adjectives (e.g., "great" or "simple"), but also as verbs (e.g., "was working fine"). Next to that, user reviews contain many subtleties. For example, sometimes users talk about a functionality that an app had, but that is not present anymore (e.g., "[this feature] was working fine, until the app was updated"). Pagano and Maalej (2013) encountered this as well. Furthermore, subtleties are also introduced by negations as those may refer to qualities (e.g., "not accurate") or something that the app does not do. Finally, it also became clear that extracting services from user reviews would be hard, since the distinction between services and functionalities seemed to be blurred for apps. Also, users seem to mainly talk about functionalities and not about services.

Summarising, from this initial analysis it became apparent that it may be challenging but possible to extract user opinions about functionalities from user reviews. A more in-depth analysis was needed before being able to create an automatic analysis script.

6.1.2 In-depth Manual Analysis

The second analysis had the main objective of gathering more information on which choices regarding the automatic analysis could be based. This analysis was done in a stepwise manner on reviews of five different apps. For the selection of the apps, the concept of *analogical distance* was applied in a light-weight manner. The problem domain as discussed in the previous chapter (i.e., the recipe app) should be seen as the target domain. The to-be selected apps needed to come from either a near field, middle field, or far field. Apps were seen as near-field if these were also recipe apps. Apps were seen as middle-field if their main topic was food, but not recipes. Far-field apps were seen as apps that did not involve food and were therefore (seemingly)

unrelated. The following selection criteria were also maintained:

- The app should be downloadable for free, since the app designer may need to be able to download the suggested apps for further inspection when using the eventual tool.
- The ratings of the apps should be at least 4.5 or higher, in order to retain only high-quality, well-functioning apps. Also, the review topics of main interest (i.e., praise, helpfulness, and feature information) are in general found in reviews with an average rating above 4.7 (Pagano & Maalej, 2013).
- Each selected app should come from a different app category of the Play Store, in order to analyse a diverse set of apps.

For each app, two different sets of fifty reviews were manually extracted from the Google Play Store: one set with the fifty newest reviews and one set with the fifty most relevant results. The results of these sets were compared to find out which type of sorting would be most optimal for extracting user reviews. The apps that were used for this analysis can be found in Table 6.1. Thus, in total, ten sets of user reviews were analysed.

Table 6.1: Apps used for the in-depth manual app review analysis. * = *At the time of selection.*

Name	**Play Store category**	**Rating***	**# downloads***	**# reviews***
Tasty	Food & Drinks	4.8	5,000,000+	90,700
Calorie Counter - MyFitnessPal	Health & Fitness	4.5	50,000,000+	2,234,462
PlantNet Plant Identification	Education	4.5	10,000,000+	89,855
Houzz - Home Design & Remodel	House & Home	4.7	10,000,000+	380,474
Polarsteps - Travel Tracker	Travel & Local	4.7	1,000,000+	26,116

For each app, the following procedure was followed two times:

1. Manually extract the text and rating for the top fifty reviews of either the newest or most relevant list.
2. Assign a unique review identifier to each review and extract the length of the review text.
3. Analyse the user reviews by calculating the mean and standard deviation of the review length and rating.
4. Split each review into sentences (i.e., period and exclamation mark as delimiters).
5. Label each sentence as either *App evaluation*, *Feature opinion*, or *Other* (cf. Gu and Kim, 2015). If the sentence covers both an app evaluation and a feature opinion, then the sentence was labelled as the latter.
6. Analyse each sentence type in terms of relative frequency.
7. Filter out sentences of the types *App review* and *Other* as these types are not of interest for the final analysis.
8. Manually extract features and opinions from each of the remaining review sentences. Something was seen as a feature when 1) it was part of the app or the use of the app and 2)

the user expresses some opinion about it (e.g., "user experience" or "notifications"). An opinion was seen as either a positive or negative statement (e.g., "helpful", "great", or "annoying").

9. Manually assign part-of-speech (POS) labels to the extracted features and opinions. For this, an online POS tagger[1] was consulted in case of doubt.

Based on these steps, a general analysis was conducted on all apps to compare the sampling method. For a set of metrics, the (weighted) mean and standard deviation (where possible) were determined. The results can be seen in Table 6.2. Some of the sets of the same app overlapped in reviews, as some of the newest reviews were also the classified as most relevant. These duplicate reviews were not removed here, since it was important to compare different sampling methods.

Table 6.2: Results of the manual in-depth analysis of app reviews. ** = Weighted mean. ** = number of characters including spaces.*

Opinion POS Pattern	**Newest**		**Most relevant**	
	Mean	SD	Mean	SD
Review length**	74.7	28.9	**224.3**	73.6
Review rating	**4.4**	0.4	3.8	0.9
# sentences	82.8	18.3	**178.4**	41.1
% App evaluation sentences*	**40.3**		27.7	
% Feature opinion sentences*	**25.1**		22.9	
% Other sentences*	34.5		**49.4**	
# features	26.6	10.2	**52.8**	24.1
# opinions	30.4	11.1	**55.6**	25.7
# features : # sentences*	0.3		0.3	
# opinion : # sentences*	**0.4**		0.3	
# features : # Feature opinion sentences*	1.3		1.3	
# opinion : # Feature opinion sentences*	**1.5**		1.4	

In general, the most relevant reviews are longer and comprise more sentences than the newest reviews. However, on average, these most relevant reviews hardly contain a higher percentage of sentences relevant in this research, namely Feature opinion sentences. Interpreting it in absolute terms, the most relevant reviews contain substantially more irrelevant sentences than the newest reviews. Furthermore, the ratings given to the newest reviews are on average higher than those given to the most relevant reviews. This is in line with the findings of Pagano and Maalej (2013). They found that, in general, the longer the reviews, the lower the rating is that users give. In absolute terms, the number of features and opinions extracted from the reviews is substantially higher for the most relevant reviews. However, the number of features and opinions per sentence or per Feature opinion sentence hardly differs between the two types of sets. Thus, while analysing the most relevant reviews results in more features and opinions, it also results in more sentences that are irrelevant to the analysis. These irrelevant sentences may introduce noise into the analysis when those are not properly filtered out.
After the comparative analysis, an analysis was conducted on the features and opinions and their

[1] https://parts-of-speech.info/

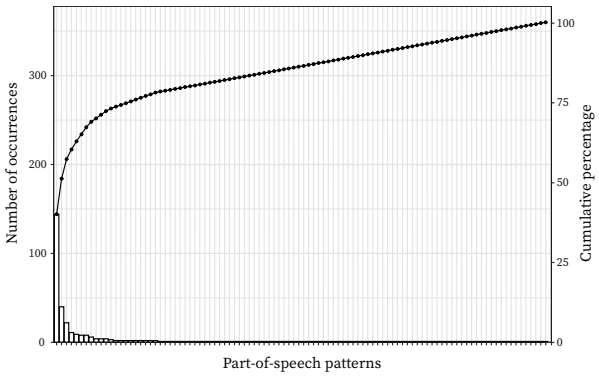

Figure 6.1: Distribution of feature POS patterns.

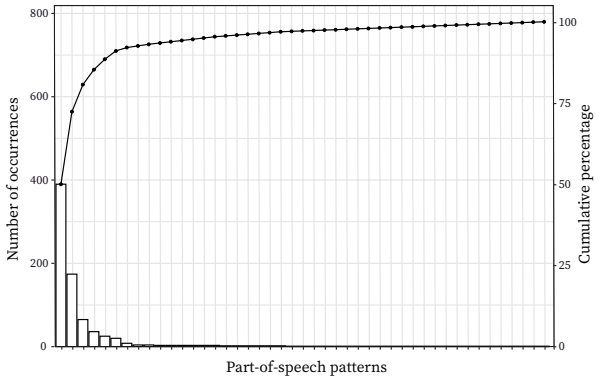

Figure 6.2: Distribution of opinion POS patterns.

assigned POS labels in order to find general patterns in the way users express opinions about app features. This was done by combining the results of all ten sets of reviews. The duplicates were removed here to make sure that those POS patterns would not have more weight in the analysis.

As can be seen in Figure 6.1 and Figure 6.2, most features and opinions are explained by only a small set of POS patterns. In total, 100 different feature patterns and 45 different opinion patterns were identified. For readability purposes, the POS pattern labels are left out in the figures. The most frequently occurring patterns can be found in Table 6.3 and Table 6.4. It must be noted that when one opinion word was used for two features, then this opinion was extracted twice. Thereby, this patterns also was counted twice. Moreover, when one feature was accompanied with two different opinion words or phrases, then this feature was also noted twice. This approach was taken since it was the main aim was to analyse pairs.

The tables show to some extent that many patterns can be seen as a subset of others. Mainly the less frequent patterns were superset of those patterns in the tables. However, when removing some words from those patterns, and thereby shortening them, would lead to a loss of contextual information in many cases. Not all features and opinions can be shortened without losing their meaning. However, the majority of both features and opinions can captured by those short patterns shown in the tables above. The results from this more in-depth analysis is in line with the observations that were made in the initial manual analysis. For instance, features are not only captured with nouns and verbs and opinions are mainly expressed with adjectives. The fact that there are this many patterns, shows that there are subtle ways of expressing opinions and features.

The extracted patterns are largely in line with those found in other researches. Although there is an overlap with the top eighteen most frequent feature patterns of Johann et al. (2017), there are some differences worth mentioning. In general, our findings are in line with theirs, since most feature patterns of Table 6.3, can be found in their list of feature patterns. In their work they only extracted features. Some of their feature patterns could be seen in our research as a feature opinion pair, since some words could also describe an opinion in a certain context (e.g., “enjoy group conversations”). The most remarkable difference is that they did not identify single nouns as feature patterns, while these are the most frequent in our work. This difference can be explained by the fact that they focused on high-level features and not

Table 6.3: Percentage of manually identified feature POS patterns.

Feature POS Pattern	Percentage	Example
Noun	40%	*Pictures*
Noun noun	11%	*User interface*
Verb noun	6%	*Edit photos*
Noun preposition noun	3%	*Suggestions for activities*
Verb pronoun noun	3%	*Analyse my progress*
Other	37%	

Table 6.4: Percentage of manually identified opinion POS patterns.

Opinion POS Pattern	Percentage	Example
Adjective	45%	*Amazing*
Verb	17%	*Love*
Adverb adjective	9%	*Very great*
Adjective adjective	6%	*Neat, clear*
Adjective noun	5%	*Great way*
Other	18%	

low-level features (consisting of single words). Furthermore, in our research conjunctions were not used, but instead split those patterns up into two. In general, the fact that there are some differences with their work is not surprising, since both annotations were done manually and they analysed a larger set of reviews. Vu et al. (2016) found substantially more patterns to denote phrases about which users have an opinion (cf. features), namely over 90,000. The main difference with their approach is that our patterns were extracted manually from less and different types of reviews than theirs (i.e., retail product reviews). Gu and Kim (2015) manually identified 26 pattern templates to extract feature opinion pairs. They decided to only keep those patterns that occurred frequently enough. Several of our patterns also overlap with theirs. They, for instance, also identified adjectives and opinion verbs such as 'like', 'enjoy' and 'love' as opinion patterns and both single and double nouns as aspects (cf. features). The main difference with their approach is that we identified separate patterns for the features and opinions instead of combined patterns. Still, in their patterns the opinions and features can easily be distinguished. Moreover, they extracted the templates from more reviews. In general, it seems that different approaches were taken, leading to different outcomes. Also, most researches had a different scope, causing each research needed a tailored outcome. We do not imply here that one approach is necessarily superior to others.

6.2 Feature Opinion Extraction

As the main goal of the research was to find out how creativity of app designers can be fostered, we decided that a relatively simple approach for the app review analysis would suffice. Moreover, it would not be feasible given the time frame of the research project to perform this part of the

research in great detail. As already mentioned before, it was most important to obtain a proof of concept. Therefore, an approach was taken which aimed at getting a "good-enough accuracy". Striving for a good precision was more important than a good recall, for the quality word clouds is most influenced by the precision.

After reviewing various options in literature (Section 3.6.1) and conducting the manual analyses, it was decided to write a set of templates with which features and opinions could be extracted as pairs. This approach was determined to be feasible, given the results from the manual analysis and the results of others. Also, relatively few steps would be needed to create a working script for this. Hence, the reviews could be analysed within the constraints discussed above. We decided not to exactly follow any of the approaches of related studies for several reasons. First, most approaches of other researches were rather extensive and aimed at achieving a high accuracy. This would not be in feasible here given the constraints listed above. Second, other feasible approaches would lead to a compromise concerning the precision. For instance, the collocation approach of Guzman and Maalej (2014) resulted in an F-score of 0.55. This approach was therefore not followed. Furthermore, this research did also not follow aproaches applied to user reviews in other fields (e.g., Hu and Liu, 2004), since app reviews are different in nature from those (Chen et al., 2014). Nonetheless, our approach is similar to those of previous works. The work of Gu and Kim (2015) can be seen as the most similar to ours. This research focused on the extraction of aspect opinion pairs. This current research focused on features, which are, according to us, a confined subset of aspects. In the context of this research, a feature was denoted as a (visual) part of the app, a behaviour of the app, or something a user can do with the app. Those things may be seen as solutions to a design problem and may be interesting to app designer. We were not able to clearly identify how Gu and Kim denote aspects. It could be that they focused on a larger set of things that could be extracted.

The general overview of our automatic extraction approach can be found in Figure 6.3. The details of this approach and the steps taken to develop the script are discussed in the following subsections. The script was created in R (R Core Team, 2020).

6.2.1 Data collection

The data that was used for creating the script were the reviews that were manually analysed in the in-depth analysis (Section 6.1.2). The reviews from both sampling methods were combined to create a larger dataset. Moreover, a set of 1,000 reviews of the Polarsteps app were used as well. These reviews were collected using an open-source tool[2]. These reviews were the newest reviews for that app at that time.

Only the review text and a unique review identifier are needed for this automatic review analysis. Other data is therefore not needed to be collected. The script is created for the English language only. Therefore, the data that is collected is restricted to that language.

6.2.2 Preprocessing

Before feature opinion pairs can be extracted, the reviews need to be preprocessed. First, each review is split into one or multiple sentences. This is in accordance with the approach of, for instance, Gu and Kim (2015). Then, punctuation (except for commas and apostrophes),

[2]https://github.com/facundoolano/google-play-scraper

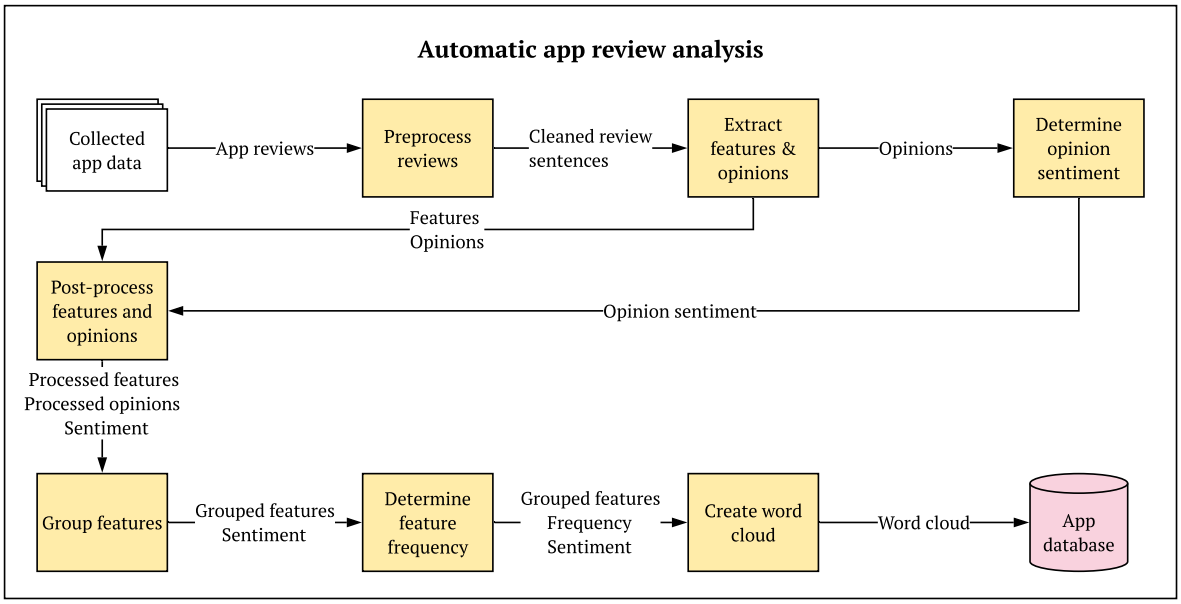

Figure 6.3: Overview of the automatic review analysis approach.

numbers, and emojis are removed, since those do not convey any useful information for the purposes of this research. Commas and apostrophes need to be retained, to preserve the sentence and/or word structures. Then, irrelevant sentences are filtered out using a custom dictionary, which was created by visually inspecting reviews and identifying keywords and phrases. This dictionary can be found in Table 6.5. The initial intention was to filter out review sentences of the type *other* and *app evaluation* like the approach of Gu and Kim (2015), as this would filter out noise. It was tried to apply a bag-of-words approach as suggested by Shah et al. (2018). However, the precision and recall we managed to obtain were too low. Hence, it was decided to filter those irrelevant sentences through the use of a dictionary comprising words and phrases that often seem to occur in those types of reviews. The use of a dictionary is not unique. For instance, Jha and Mahmoud (2019) also used a custom dictionary on app reviews. However, they used it for classifying reviews into non-functional requirement classes. The dictionary currently only contains fourteen phrases. Thus, this dictionary is far for complete. Still, these phrases manage to filter out initial noise to some extent. A more extensive analysis is needed to make the dictionary more complete. This shall benefit the quality of the script even further. Using a dictionary for filtering out sentences may cause that also relevant review sentences are filtered out. This is not a considerable issue, since precision is more important than recall in this research.

It was tried to automatically filter out languages different from English through the use of an existing package. However, this step was omitted, since most words were misclassified as an incorrect language. Even though one can specify the language when collecting reviews, other languages can end up in the dataset. It could unfortunately thus still be the case that languages different from English appear in the final visualisation. Vu et al. (2015) also encountered this issue and created a custom algorithm for removing non-English reviews.

Table 6.5: Dictionaries used to filter out irrelevant sentences and feature opinion pairs.

Name	Step	Words & phrases
Prefilter sentences	Preprocessing	Would like, would love, would be, needs to be, would also love, would also like, thank, wish, complaint, please, improve, bring back, add the feature.
Post-filter features	Post-processing	App, way, aap, experience, thing, work.
Post-filter opinions	Post-processing	Fellow, new, straight, main, permanent, chosen, forthcoming, free.

6.2.3 Extracting Feature Opinion Pairs

As already mentioned, features and opinions are extracted by a set of templates. These templates are based on the most frequently occurring POS patterns that were identified in the in-depth manual analysis. Dependency parsing and POS tagging are used to extract the feature opinion pairs. The templates used for extracting feature opinion pairs can be found in Table 6.6. These templates comprise combinations of the three most occurring feature patterns and the two most occurring opinion patterns. Only these patterns are used, since they capture already around 57% and 62% of the features and opinions respectively. The number of rules needed to capture more patterns would grow substantially, while the benefit would increase minimally.

Table 6.6: Templates used to extract feature opinion pairs.

Feature	Opinion	Example
Noun	(Negation) adjective	*Not accurate + distance*
Noun	(Negation) verb	*Like + map*
Noun noun	(Negation) adjective	*Good + location tracking*
Noun noun	(Negation) verb	*Love + photo album*
Verb noun	(Negation) adjective	*Love + take pictures*
Verb noun	(Negation) verb	*Like + write text*

The review sentences need to be annotated first to extract the feature opinion pairs. The *udpipe* package (Wijffels, 2019) is used for this, since it is able to tokenise, lemmatise, and parse dependencies of the raw review sentences and return the results in a single, usable format. The output is a dataframe containing each word of the dataset, its lemmatised version, POS tags, its dependency relation, and the parent word.

The rules in the script were drawn up by visually inspecting the annotated dataframe. The rules comprise both dependency relations and POS tags. In general, words are selected based on their POS tags (i.e., “adj”, “noun”, “verb”, and “part”) and dependency relations from the annotated dataframe. The dependency relations make up the final structure of the patterns. The specific POS tag and the dependency relation (i.e., parent or child) determine whether a specific word is extracted as a feature or opinion. Both the feature and opinion are extracted simultaneously. All features and opinions are returned in individual columns in a data frame. Each row specifies one feature opinion pair and the related review identifier.

Only the dependency relations that were assessed to considerably improve the recall while not substantially decreasing the precision were implemented. As mentioned in Section 6.1.2, many patterns are subsets of others. This was taken into account by merging dataframes in order to filter out subsets of features. Negations are only implemented for the opinion patterns, since more complex measures were needed to implement those in the feature measures. However, we believe that most negations relate to opinions and not features. For instance, many people would say "I do not like adding photos", while we believe that less people would say "I like not having to add photos". This latter phrase would need more advanced templates. The opinion verbs are extracted using a custom opinion verb dictionary and pattern matching. A dictionary is used for this, because only a limited amount of verbs that express opinions (i.e., "love", "like", "enjoy", "hate", "dislike", and "adore") which can easily be extracted through pattern matching.

While many feature opinion pairs were reasonably good (see examples in Table 6.6), also many were not good enough for the visualisation. For instance, many extracted opinions, were not actual opinions (e.g., "standard", "possible", "multiple", "such"). Also many features were not actual features but aspects and/or were not meaningful for our purposes (e.g., "thing", "experience", "app"). This can partly be attributed to the fact that the annotator is not completely accurate and to the fact that still irrelevant sentences are present in the dataset. Moreover, still many spelling mistakes were present as a result of poor-quality reviews. This can mainly be observed in the extracted features. Finally, since only six small patterns were used that may be a subset of larger patterns, many extracted features and opinions lack contextual information or subtleties (see also Section 6.1.2).

6.2.4 Post-processing

Post-processing is needed to improve the quality of the results. For this, a series of steps is taken. First, meaningless features and opinions are filtered out by using patterns matching. Again, custom dictionaries (Table 6.5) were created for this by manually inspecting the results. Second, sentiment analysis is applied to filter out neutral 'opinion' words. The R package *sentimentr* (Rinker, 2019) was used for the sentiment analysis, since this package is able to handle negations, while to the best of our knowledge others are not. Lastly, in some cases, the end users expressed multiple opinions about the same feature in one review. These features would therefore end up multiple times in the final set. Thereby, these reviews would potentially get more weight than other reviews in the word cloud. To prevent this, the sentiments relating to the same feature in the same review were averaged.

The first two steps filter out a substantial amount of feature opinion pairs and improve thereby the quality considerably. A quick inspection was done on the results of the automatic analysis of the labelled dataset that was also used for the in-depth manual analysis. This inspection showed that a substantial amount of feature opinion pairs from the irrelevant sentences were removed by filtering on sentiment.

6.2.5 Feature grouping

The extracted features need to be grouped, since many convey the same feature in different words (Dalpiaz & Parente, 2019; Gu & Kim, 2015; Guzman & Maalej, 2014). This is done by first creating a distance matrix and then by hierarchically clustering the features. Clustering is also used in related work to group features or feature phrases (see e.g., Gu and Kim, 2015;

Vu et al., 2016). The cosine dissimilarity was used as the distance metric and Ward's method was used for the hierarchical clustering. Al-Subaihin et al. (2016) also used both hierarchical clustering for grouping apps with Ward's method and cosine dissimilarity. According to them, these methods are better than Euclidean distance and other hierarchical clustering methods. Others also used cosine similarity for grouping extracted features (see e.g., Dalpiaz and Parente, 2019; Johann et al., 2017). These methods were therefore selected in this study. The threshold set for the hierarchical clusters is 0.20. A visual inspection showed that this threshold would cluster as many features as possible, while compromising the precision as little as possible. After the clustering, the label of the most frequent feature in each group is assigned to the entire group (like e.g., Gu and Kim, 2015; Guzman and Maalej, 2014). Finally, all features in the same group are summarised into one feature by taking the average sentiment of each opinion over all features in the group. This is done to make the features suitable for the visualisation.

A visual inspection showed that not all features that would be good candidates for grouping are grouped. For instance, longer features (e.g., tracking and tracker) are not grouped, while shorter features that are less meaningful may be more easily grouped. This results in somewhat redundant features. However, this is not a major limitation, since the main objective is to inspire and not to return accurate features. Other approaches for grouping features were also considered. For instance, it was considered to use semantic similarity instead of cosine dissimilarity (see e.g., Johann et al., 2017). However, we could not find a method for this in R that would fit the constraints that were mentioned at the beginning of this section. Another approach that was considered was grouping synonyms (see e.g., Guzman and Maalej, 2014), using, for instance, WordNet (Miller, 1995) or the R package called *qdap* (Rinker, 2020). However, finding appropriate synonyms would be a complex task which would not fit the aforementioned project constraints, as synonyms can be very context dependent.

6.2.6 Visualisation

The resulting outcomes of the analysis needed to be shown in a meaningful, yet comprehensible manner to the app designer (Pagano & Maalej, 2013). Features that are frequently discussed should be emphasised, since those are apparently an important topic of discussion. Besides that, the users' opinions with regard to those features should be captured. Word clouds are a compact way of conveying multiple types of information about words and phrases. Both the sentiment and frequency of occurrence can be captured in a single word cloud. Therefore, this visualisation type is selected. Others (e.g., Huang, Etzioni, Zettlemoyer, Clark, and Lee, 2012) have shown that word clouds are a usable type of visualisation for summarising user reviews.

The word clouds are created with the R package *wordcloud* (Fellows, 2018). The size of the words is determined by the frequency of the grouped features. The colour of the words indicates the polarity and strength of the average sentiment. A red to green gradient is selected for this, where on the one end a sentiment of -1 is visualised as dark red and on the other end a sentiment of 1 is visualised as dark green. Review ratings were not incorporated here, since it was observed that users can still express a highly positive opinion about a feature despite giving an overall low rating (and vice versa).

Figure 6.4 and Figure 6.5 show two example word clouds. Only words are visualised that have a frequency of five or higher. The word cloud on the left is created from the 1,000 reviews that were used to create the script. The word cloud on the right is created on a new set of 2,000 reviews from a different app. The review sampling method differs between the two. The

Figure 6.4: Word cloud of 1,000 newest Polarsteps reviews.

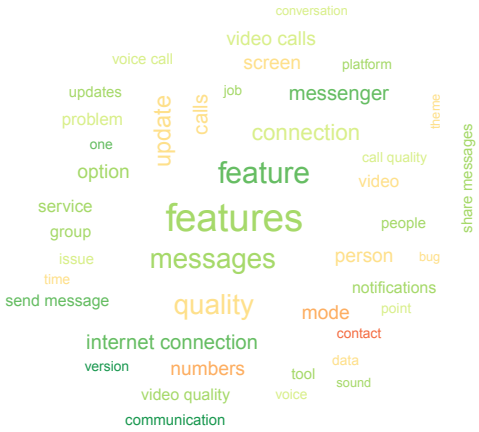

Figure 6.5: Word cloud of 2,000 most relevant WhatsApp reviews.

other sampling method (i.e., newest reviews) was also used for the app of the right word cloud, but quality of the resulting word cloud was substantially lower. That word cloud contained, for instance, substantially less features. Also, the lower frequency features that did not end up in the word cloud (i.e., below the threshold of five) showed various spelling mistakes. As can be noted, the quality of the word clouds of Figure 6.4 and Figure 6.5 differs to some extent, however their quality still seems reasonably good. The left word cloud visualises more meaningful features than the one on the right. It seems that the one on the right also shows various aspects (e.g., "internet connection"), that cannot be seen as features of the app. So, not all word clouds may have the same quality as these two. Overall, this indicates that the quality of the word clouds is affected by the quality of the reviews, despite the cleaning efforts described earlier.

Currently, the opinion words are not (yet) captured in the created word clouds. In the future, it would be meaningful to show the opinion words when the app users would, for instance, hover over the word cloud.

Chapter 7

Analogical Reasoning

Design-by-analogy forms an essential part of the design of the creativity support tool. To foster the creativity of app designers, analogous design problems will be provided as examples. It is intended that app designers first make a mapping from design problem at hand to the analogous design problem and after that make inferences to find a solution to their design problem. Markman, Wood, Linsey, Murphy, and Laux (2009) described that it is less difficult for people to spot an analogy when a comparison between domains is presented than identifying analogous problems by themselves. Therefore, presenting analogous apps aims at helping the app designer in making mappings to and inferences from analogous design problems. Other reasons for incorporating design-by-analogy into the tool were already discussed in Section 5.1.

Section 3.3 described that analogical distance is a central concept in design-by-analogy. This concept is often associated with design fixation. In our strive for preventing a possible design fixation, analogical distance is used to find appropriate example apps. The foremost challenge in this strand of the research was to develop a *method for depicting different analogical domains and determining distances between those domains in a scalable way.* The approach needs to be scalable for the tool to cover a wide variety of design problems (R4). Besides that, the method also needs to be easily implementable and usable in the eventual tool.

Analogies are made based on structural similarities between domains (Gentner, 1983; Holyoak & Thagard, 1997). In order to find analogous apps that can be presented to the app designer, it is thus required that the apps are described in a structural way. These structures need to be decoupled from elements that describe the surface of apps (Gentner, 1983; Gentner, Rattermann, & Forbus, 1993). We reason that apps that are seemingly very different can still solve the same underlying design problem and can have a similar high-level purpose. The notion of **goals** is found to be useful here for several reasons. First of all, the same goal may apply to different types of apps that seem reasonably different. For instance, both a recipe app and a house decoration app could serve the end user's goal of searching for inspiration. Thus, the underlying design problems that these apps are trying to solve are similar. Thereby, goals allow to make structural comparisons between apps from different domains (i.e., with different surfaces). Secondly, we suggest that there is a finite set of goals that could cover the entire app field. This makes goals both feasible and usable for the eventual tool. Thirdly, end-user goals are translatable to requirements (Dalpiaz, Franch, & Horkoff, 2016), making them suitable for the specific context of this research. We pose that high-level goals still allow for creativity as one goal can be fulfilled in multiple ways with different functionalities. Presenting analogous apps that tend to fulfil the same high-level goals may ultimately give the app designer new perspectives on how the underlying design problem can be solved. Finally, appropriateness or usefulness are a precondition for creativity. These aspects are inherently taken into account by making mappings based on goals, as goals represent the end-users' intention of using the app.

Next to describing the structure, a way of describing the surface of an app is needed in order to determine the analogical distance between different apps. The approach of listing **keywords** was chosen for this for several reasons. First, listing a set of keywords that describe an app is a relatively easy task that does not take much time. Secondly, keywords can be easily used to determine the distance between domains. Both arguments will be further elaborated on later. Thirdly, it does not require much effort (compared to other methods, see Section 5.3) for the app designer to list keywords in the eventual tool, making the approach feasible for the tool.

We argue that the combination of a selection of goals and specific keywords defines an app and makes an app unique to a certain extent. The following sections elaborate on the decision for goals and keywords and on how those are applied to find analogous apps.

7.1 Structure - Goals

The choice for using the notion of goals to describe the analogical structure of an apps is a result of the organic development of ideas described in Section 5.3. Also, the goal notion and goal modelling languages such as iStar 2.0 are often used in fields such as RE (Dalpiaz et al., 2016). Concepts of this specific modelling language seem to translate well to the app design field and this research (Table 7.1). However, this does not imply that this research is confined to a specific goal modelling language. Still, these iStar 2.0 elements tie together the objectives in this research: the need to generate new requirements and app design ideas (i.e., tasks and resources), user feedback in app markets (i.e., qualities), and forming analogical relations (i.e., goals).

Table 7.1: iStar 2.0 elements translated to the current research.

iStar element	**Translation to app design and this research**
Goal	The end user's high-level reasons for downloading a specific app.
Task	The use of the functionality of an app or an app in general.
Resource	A specific functionality of an app or an app in general.
Quality	Users' opinions regarding (a feature of) the app.

7.1.1 Goal analysis

To the best of our knowledge, no goal taxonomy or generic set of goals existed yet for the app domain. Hence, a set of generic goals needed to be created first before being able to retrieve analogous apps. To create this set, a diverse set of apps needed to be analysed. Next to that, for the tool to cover as many design problems as possible, many analogous apps should be incorporated in the tool. Thus, set of goals needed to cover as many existing apps as possible, if not all. Also, the goals need eventually to be tagged for a large set of apps. As discussed, this task is done manually. To make this process scalable and workable for human annotators, the final set of goals needs to be relatively small. To fulfil these objectives, the identified goals need to be high-level. Also, as mentioned at the beginning of the chapter, we reason that high-level goals allow for creativity. In general, the strive was to create an as complete as possible set of goals. However, full completeness was not striven for for two reasons. First, we hypothesised that each app could at least be covered by one goal that was present in the identified set, as

those are high-level. Second, it would not harm that some apps would not be usable for the tool, since already many apps are covered by the goals.

To create a rather complete set of goals, a somewhat large and diverse set of apps needed to be analysed first. Because this was a manual analysis, a diverse set of 100 was selected. The precise selection criteria and procedure can be found in Appendix B. In short, 100 free apps with a rating of at least 4.5 were selected from almost all Google Play Store categories. The following steps were taken to create the set of high-level goals:

1. Assess each app page in the Google Play Store. Inspect the description, title, and pictures.
2. List for each app both the explicit and implicit low-level goals. Explicit goals were identified while assessing the app page. Implicit goals were identified by reasoning about the use of the app.
3. Assess the entire list of identified low-level goals to recognise common patterns that form possible high-level goals.
4. Add each identified new high-level goal to the list.
5. Make sure that there is no unnecessary or avoidable overlap between goals and that goals are independent from surface elements descriptors (e.g., keywords). Unnecessary overlap can be avoided by altering the goal description or the terminology used to describe the goal.
6. Refine the list of high-level goals.
7. Iterate over step 4 through step 6, while in between re-tagging the set of selected apps a couple of times to check if the set covers all apps while not being confusing for the tagger.

Initially, a set of about ninety lower-level goals was extracted. It would not be feasible for annotators to tag apps with set of this size. Refinement sessions were held by the author and her first supervisor. After various iterations and refinement sessions, a final set of twenty goals remained (Table 7.2). Seventeen of those goals (i.e., Goal 1 through 17) were assessed to be usable for the analogous app selection. The other three goals (i.e., Goal 18 through 20) were not assessed to be usable, because those covered relatively few apps and/or were too confusing for human annotators (Section 7.3.1). An additional line of explanation is provided for Goal 1 (i.e.,“ E.g., experiences, activities, behaviour, trends, events, progress, etc.”) and Goal 18 (“E.g., open, explore, remove, edit, scan, etc.”) to clarify the goal for the annotator. This was not done for the other goals, because these goals were assumed to be clear enough and overspecifying those is expected to restrict annotators. Before the final refinement, a workshop was held with researchers from Utrecht University to determine whether people are able to tag goals using the identified set and to evaluate whether the identified goals were clear to them. It became clear from the discussion during the workshop that other additional lines of explanation were not required. This pilot workshop and the outcomes are further discussed in Section 7.3.1. As can be seen in the table below, the goals are described in relatively general terms. Again, the goals need to be combined with keywords to describe an app. It is therefore essential that the terminology of goals is as general as possible. Using ambiguous or vague words allows for a freer interpretation and thus potentially for a wider variety of apps to be grouped under a goal. This contributes to giving app designers different perspectives on their own design problem. As can be noted, the granularity is not the same for all goals. For example, *Get guidance or assistance* is less specific than *Stay mentally and physically healthy.* It was attempted to make the granularity the same for all goals. This was not possible, because the goals would become too vague and would lose their specific meaning. However, this is not a limitation for the intended use of the goals in this research.

Table 7.2: High-level end-user goals that apps fulfil.

1.	Log, monitor & track	2.	Be informed & obtain knowledge	3.	Learn & practice skills
4.	Stay mentally and physically healthy	5.	Get guidance or assistance	6.	Organise & plan
7.	Replace physical tools and objects with an app	8.	Control IoT or physical devices	9.	Make & complete transactions
10.	Save money	11.	Search & discover	12.	Be entertained
13.	Communicate & share	14.	Create & alter	15.	Personalise & customise
16.	Get inspiration for new ideas	17.	Preserve security, privacy & safety		
18.	*Perform any action on physical or digital files*	*19.*	*Contribute to project*	*20.*	*Arrange matters*

7.2 Surface Elements - Keywords

As mentioned at the beginning of this chapter, keywords are used to describe the surface of the apps. These keywords are a simplified manner of describing the objects and attributes of the apps as compared to, for instance, using predicates (cf. Gentner, 1983). Following a simple approach ensures that a wide variety of people can annotate apps, which in turn contributes to scalability. Essentially, the keywords need to capture the theme and superficial characteristics of an app. In this way, it can be determined whether apps belong to the same domain. On top of that, the distance between domains can be determined as well (Section 7.4). Keywords that could describe the recipe app of the toy example include: *recipes*, *food*, *home cooking*, *ingredients*, and *breakfast*. A home decoration app could be described using the following keywords: *house decoration*, *interiour*, *design ideas*, *exteriour*, and *styling*. Keywords mainly consist of one word, but can consist of up to three words to give the annotator more freedom.

7.3 Goal & Keyword Tagging

A large set of apps needs to be tagged first, before being able to determine which apps are analogous to the design problem at hand. This tagging is done by human annotators (Figure 5.3 & Figure 7.1). Crowdsourcing would be a feasible option for this, since the tagging process is found to consume not much too time and it was found to be rather easy (Section 7.3.1). The tagging process needs to be done manually for several reasons. While there are already efforts to automatically extract goal models from app reviews (Shimada, Nakagawa, & Tsuchiya, 2019), we believe that humans are to date still better at reasoning about goals. Letting humans annotate apps using the set of Table 7.2 is therefore most appropriate here. We recognise that keywords could also automatically extracted from app pages. However, the quality and completeness of the set of keywords would be heavily influenced by each specific app page. Two app pages of similar apps could use a different terminology to describe their app. Automatic extraction could cause that the extracted distance would be higher than the actual distance between those apps.

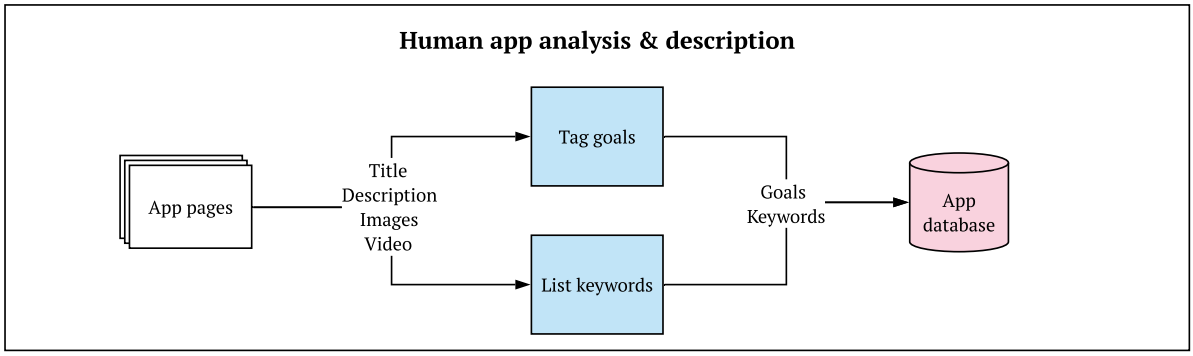

Figure 7.1: Overview of the human goal and keyword tagging process.

App pages in app markets are helpful for describing the apps in terms of goals and keywords. Information that is usable for this can be found in the title, description, pictures, video (if available), and even in user reviews. A good practice would be for the annotator to assess the app page and after that first tag the goals and then list keywords that come to mind when assessing the app. Another good practice would be to provide an example, so that annotators know what to do. This may benefit the consistency of the results. Only goals that describe the reason for downloading an app best should be tagged. A guideline on how many goals to tag could be provided. For instance, our final tagging of 100 apps resulted in an average of 2.8 goals per app (SD = 1.3), which can be used as a guideline.

Tagging goals is subjective as goals may vary from person to person. However, we believe that each app still fulfils a specific set of goals. When many people would tag goals, agreement between the annotators could be used to discover the chief goals. It is also advisable to provide a diverse set of apps that cover a wide variety of goals to annotators. This may help annotators see better which goals may apply to a specific app and which may apply better to other apps.

For listing keywords, the app description may give the most information. However, persons may also come up with their own words to describe the app. There is not a specific upper bound for the number of keywords the annotators may come up with. In general, more keywords will describe an app in a more complete way. Although, it must be prevented that general terms are used that apply to many apps. Around fifteen to twenty keywords per app would already be a good starting point for determining the distance between those apps. Prescribing specific rules for listing keywords ensures that the results are more consistent and easier to process. Processing steps such as stemming or lemmatisation can control for multiple way of describing the same thing. It is assumed that the chance of making spelling and grammar mistakes is small, as keywords consist only of words or very small phrases. Still, it may be needed to correct mistakes while processing the keywords. To make the set of keywords more complete, additional keywords could be suggested to annotators (cf. Vu et al., 2015).

7.3.1 Pilot Workshop

A pilot workshop was held with researchers from Utrecht University to find out whether people are able to tag goals and list keywords for a set of apps. This workshop was also held to evaluate whether goals were clear enough or whether a final refinement iteration was needed.

The participants of the workshop were asked to perform the tagging exercise beforehand. In

total, eight persons participated in the workshop. They were split into two groups, to assess a wider variety of apps. Each group was assigned a set of ten different apps and each participant needed to perform the exercise individually beforehand. Two diverse sets of apps were provided in order to assess whether all goals were clear enough. As explained earlier, providing a diverse set of apps was also expected to help the annotators in their job. Apps were selected from the 100 apps that were used to create the list of goals, to compare the participants with those of the author. Some specific apps were selected, since those were assessed to be good candidates for discussion. The apps were then evenly distributed between the two groups. It was also ensured that each general theme of an app would only occur once. The instructions and the sets of apps for the exercise can be found in Appendix C. It must be noted that the list of goals the participants received slightly differs from the final set of goals. Those goals were the result of the second last iteration and were updated into the final set of goals after the workshop. During the workshop, the participants' experience with the exercise was discussed. For this, a set of questions were prepared to guide the discussion (Appendix C). Separate discussion sessions were held with each group to make the discussion more structured. The author and her first supervisor guided the discussion sessions.

Table 7.3: Results of the analysis of the workshop exercise. * = *Per participant.*

	Mean # goals*	**SD # goals***	**Mean # keywords***	**SD # keywords***	**Mean goal agreement**	**Mean % of keyword unigrams***
Group 1	2.73	0.98	7.68	1.58	0.55	73.4
Group 2	2.80	0.47	6.05	1.14	0.55	75.7
Total	2.76	0.71	6.86	1.54	0.55	74.5

Table 7.3 shows general outcomes from the analysis on the participants results. As can be seen, most results are very similar between the groups. The biggest difference between the groups is the average number of keywords provided. The total average number of goals is very similar to the average that was taken over the 100 tagged apps (i.e., 2.78). Participants were instructed to provide at least five keywords. On average, only few additional keywords were given by the participants. Only one participant provided more than nine keywords on average. The keywords that were provided were rather consistent amongst the members of the groups. This was probably due to the fact that they all assessed the app pages. Nevertheless, none of the sets of keywords completely overlapped between the participants. Most participants provided keywords consisting of only one word, however bigrams and trigrams were provided as well. The agreement that a goal should be tagged was rather low among all participants, with a weighted average of 0.55. For each goal, the agreement was determined to be 1.0 if all four participants tagged the goal for a specific app and 0.25 if only one of the participants tagged the goal. The weighted average agreement per goal (Appendix C) was taken over all apps of the two groups for which this goal was tagged. The total average agreement reported in Table 7.3, is the average over all goals. The low agreement can possibly be accounted to several things. First, some participants were stricter in tagging apps than others, which is reflected in the average number of tagged goals per participant. Second, as already touched upon, goals are subjective. A specific goal can for one person be a primary goal, whereas this same goal can be a secondary

goal for another person. Third, during the discussions it became clear that participants had different interpretations of the goals. This is not surprising, since many goals are formulated ambiguously. The following key takeaways summarise the discussion during the workshop:

- **Duration.** On average, participants needed about thirty minutes to tag the goals and list the keywords for a set of ten apps.
- **Strategy.** The strategy that the participants took somewhat differed. Most did the exercise in a linear fashion, while some went back to check or revise their answers of previous apps.
- **Instructions.** In general, the provided instructions were clear to the participants. Some found it difficult to determine how many goals to tag and to decide whether something was a primary goal or not. This was also seen in the analysis outcomes: one participant tagged on average only 1.5 goals, whereas another participant from the same group tagged 3.9 goals. The participants suggested to give an approximate guideline on how many goals to tag.
- **Difficulty.** Overall, participants did not find it difficult to perform the exercise. Most participants found it more difficult to list keywords than tagging goals, since the former required them more to be creativity. Some noted that it was sometimes difficult to adhere to the prescribed rules for listing keywords. Nonetheless, the quality of the resulting keywords was satisfactory. Also, coming up with more than five keywords was a challenge for some. Some noted that it could have helped to know the relation between the two parts of the exercise. Multiple participants mentioned that they had the feeling that at least one or two goals could be tagged without any difficulty for each app. Finally, it seemed that it is more difficult to tag goals of apps that either expose very few functionalities or that expose a wide variety of functionalities. Also, the quality of the description and the familiarity of the app were mentioned to influence the difficulty of the exercise.
- **Familiarity.** All participant noticed a difference between 1) apps they had experience with, 2) apps they had not used themselves but that were similar to apps they used or for which they could imagine its intended use, and 3) apps about which they had little to no knowledge about beforehand. Overall, apps of the first category were found to be easiest to process. Some participants noted that apps of the second category may cause a bias, since annotators may expect certain functionalities based on experiences with other apps, while these may in fact be absent. This could case faulty annotations. When letting a large group of annotators tag goals, familiarity with apps could, for instance, be taken into account by adjusting their weights in the agreement between annotators.
- **App page inspection.** The way in which participants assessed the app pages differed to some extent amongst the participants. Most of them read the title and description and looked at the pictures. Two looked at the reviews to either get inspiration for keywords or to determine goals. Videos were less often consulted by the participants. Most participants consulted the description during the keyword part of the exercise.
- **Clarity of goals.** Various goals were somewhat confusing for participants. Also, when asking about the participants about their interpretation, it became clear that not all goals were clear enough. This was mainly the case for the following goals: *Perform any action on physical and digital files*, *Personalise & customise*, *Organise & arrange matters*, *Create & design*, and *Get guidance & be assisted*. Some of those were adjusted, while others were removed from the main set of goals. Some participants also noted to experience an overlap between pairs of goals. However, since there was no widespread consensus on this, no

alterations were made regarding those experienced overlaps. Some other suggestions of participants regarding goals were taken into account in the final set.

- **Extra line of explanation.** Not all participants needed the extra line of explanation that accompanied some of the goals. Opinions about the usefulness of those lines were mixed. Since there was no widespread consensus on this, no large alterations were made to the extra lines of explanation. Only some small suggested changes were made to those lines after the discussion.

The results from the workshop indicate that manually annotating apps in terms of goals and keywords is a feasible task. The task does not seem very difficult and it does not require a substantial amount of time to tag a set of apps. An inspection of the tagged keywords indicated that keyword tagging can be done rather consistently. Also, the quality of the keywords was satisfactory. The average agreement for the goal tagging may not have been high, but this is not a major obstruction when many people tag a set of apps. If many people perform the exercise, the primary goals can be extracted through the use of agreement. Overall, we suggest that it would be possible and feasible to let a crowd annotate apps. This also suggest that the method is scalable, as required for the tool.

7.4 Finding Appropriate Example Apps

When the goals have been tagged and the keywords have been listed for a large set of apps, appropriate example apps can be determined and selected (Figure 5.2). The app designer first needs to determine the high-level goals for the app to fulfil. Then, the designer needs to specify a list of keywords. This list should be around fifteen to twenty keywords to determine the similarity. In case a tool is in place, less keywords could be required by using automatically supplemented keywords. In this section, we propose a method and underlying theory for classifying apps which can be used for selecting example apps for the eventual tool.

7.4.1 Example App Selection Matrix

According to our theory, an app can be roughly classified into four categories (i.e., quadrants) based on the similarity of the goals and the dissimilarity of the keywords between the example app and the design problem at hand. The combination of the goal similarity and the keyword dissimilarity determines how an example app can be classified. More specifically, it determines the place of an app in the example app selection matrix (Figure 7.2). A specific app can take different positions in the matrix for different design problems. This categorisation is suitable for automation, since both the goal similarity and keyword dissimilarity can be determined automatically.

Our theory is based on theories of analogical reasoning, design-by-analogy, and analogical distance. Hence, the matrix and the terminology that denotes the four quadrants are mainly derived from literature on those concepts. The matrix is mainly based on the works of Gentner (e.g., Gentner, 1983). Four unique combinations can be made when considering both goal similarity and keyword dissimilarity, leading to the four quadrants. The y-axis determines the distance between the domains, whereas the x-axis determines the strength of the analogy (e.g., Christensen and Schunn, 2009). Both the keyword dissimilarity and goal similarity are continua. That is, the borders between the four quadrants are not strict. K. Fu et al. (2013) also denotes the analogical distance as a continuum and stated that there is an optimal distance between

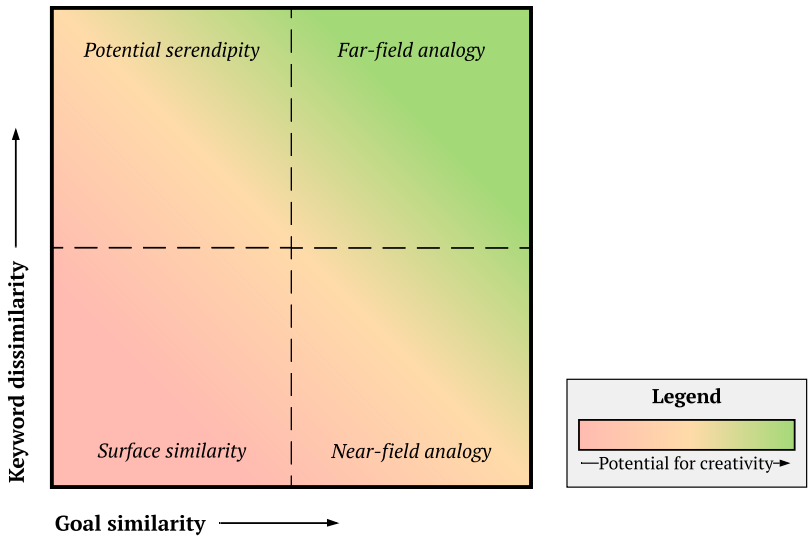

Figure 7.2: Example app selection matrix.

domains. This matrix thus also allows for searching for an optimum as both axes are continuous. The four quadrants entail the following:

- **Potential serendipity**. Apps that share few to no goals and surface elements. These apps are thus not analogous to the design problem at hand (Gentner, 1983). We hypothesise that the chance of finding appropriate solutions in those examples is low, as different problems are solved. Despite that, it may still be possible to come up with new functionality ideas by looking at these examples. Therefore, this quadrant is called Potential serendipity. According to Markman and Wood (2009), serendipity may cause the generation of new ideas. Since those new ideas coming from this quadrant are still within the app field, they may still be useful and thereby creative. We hypothesise that literally applying example elements to the design problem is less easy, since those elements may need to be adjusted first to be fitting for the design problem. In other words, since the domain is far, it is expected that app designers will less tend to use example elements in their own design (Christensen & Schunn, 2009).
- **Far-field analogy**. Apps that share many goals, but few (or no) keywords with the design problem at hand. As these apps have a low surface similarity with the design problem at hand, we hypothesise that it is more difficult for app designers to literally apply example elements to their own design (Christensen & Schunn, 2009). App designers may be forced to (analogically) reason about the relevance of the example apps to their own design problem, since the domain is different from the one of their app. Also, example elements may not be directly applicable and changes to those may be needed before they could be even used in the design for the app at hand. Hence, we expect that the design outcomes will be more original and creative by assessing these far-field analogies (Christensen & Schunn, 2009; Dahl & Moreau, 2002). Also, we expect that the creativity of the resulting ideas will be more manifest (Gentner & Markman, 1997).
- **Surface similarity**. Apps that share many keywords (i.e., surface elements) and few (or no) goals. Thus, these apps are not analogous to the design problem at hand (Gentner,

1983). We still hypothesise that these apps may inspire the app designer with new ideas. However, literally applying those ideas may be easy, as both are in the same domain (Christensen & Schunn, 2009). We thereby argue that the potential for true creativity of the design outcomes is low, as the ideas in the resulting designs may resemble those of the examples. This quadrant refers to the Appearance match of Gentner (1983). Gentner (1983) described that Appearance matches may seem applicable to certain contexts, but their actual usefulness may be low in many cases.

- **Near-field analogy**. Apps that share both many goals and many keywords with the design problem at hand. We hypothesise that it easy to literally apply example elements to the designer's own design, since the apps tend to solve the same problem in the same domain. Example elements are thus highly relevant to the design task and can thus be easily transferred to the new design. Hence, we argue that the outcomes will be less original and creative compared to the Far-field analogy quadrant (Dahl & Moreau, 2002). Also the creativity of the ideas may be less obvious (Gentner & Markman, 1997). This quadrant corresponds to the Literal similarity of Gentner (1983). Christensen and Schunn (2009) advice not to provide near-field examples when the goal is to generate novel outcomes that seem reasonably different from the examples shown.

When taking the recipe app design problem as the target domain, the following examples could be found for each of the quadrants. A Potential serendipity app could be an event management app, a Far-field app could be a interior inspiration app, a Near-field app would be another recipe app, and a Surface similarity app could be a calorie counting app. The real-world examples of apps from the four quadrants can be found in the next chapter.

According to Gentner et al. (1993), domains with a higher surface similarity are easier to retrieve. Their research also indicates that the higher the structural similarity, the more valid the domain appear to be. It may thus be easier for app designers to use of apps from two quadrants at the bottom of Figure 7.2. However, the apps from the quadrants on the right-hand side of Figure 7.2 may in fact be more appropriate, as they aim at solving the same underlying design problem. We hypothesise that the design fixation in terms of literally transferring example elements is prevented most by selecting apps from the upper quadrants (Christensen & Schunn, 2009). We believe that app designers need to reason more about the apps and that for those apps it may be harder to literally transfer the elements without adjustments. Also, since those apps are in further domains, we expect that those apps will broaden the designer's exploration space and thereby reducing design fixation in our second interpretation of the concept (Crilly & Cardoso, 2017).

K. Fu et al. (2013) suggested that some domains may be too close or too distant. We argue, however, that the issue of "too far" is confined in this context, since all analogous domains are still within the app field. Hence, the app designer is assumed to still being able to reason about the relevance of the shown examples. The issue of "too near" may in our case be confined by only showing examples with a high keyword dissimilarity (i.e., Potential serendipity or Far-field analogy apps). All in all, it is expected that the app designers will benefit most from apps from the Far-field analogy quadrant. Also, those apps are expected to limit the design fixation and foster creativity most (Christensen & Schunn, 2009; Dahl & Moreau, 2002). The next chapter further elaborates on the hypotheses and substantiations regarding our app selection theory.

7.4.2 Selecting Far-field Apps

To select apps from the Far-field analogy quadrant, a series of steps need to be taken. At this stage, app designers have already specified the high-level goals and listed the keywords for their app. The selection of appropriate example apps is a multi-objective optimisation problem (Isermann, 1982), as the selection needs to fulfil to a set of criteria. All apps need to meet the following preconditions to be selected:

- The overall rating of the app need to be at least 4.5 stars. As already discussed, the apps need to be of high-quality. This is in place to prevent transferring defective solutions to the app designers' own design (cf. Jansson and Smith, 1991). Rating is used as a measure for quality here, because the experience of users is essential in this research.
- The apps need to be downloadable for free, for the app designers' may need to further inspect the example apps.
- The examples shown should not be common, since more common examples may lead to less novel design outcomes (Chan et al., 2011). Commonness can for instance be measured by the number of downloads in the app market.
- Ultimately, the app designer should not be familiar with the example apps. Familiarity with products may prevent people to come up with new ideas regarding those products. Their familiarity thus may form a certain bias (Crilly & Cardoso, 2017; von Hippel, 1986). However, accounting for familiarity in the eventual tool may be difficult.

By taking applying these criteria, a large set of apps shall be filtered out. For the remaining apps, the goal similarity and keyword dissimilarity can be determined in a stepwise manner. Lexicographic optimisation (Isermann, 1982) is applied here, as it is a multi-objective optimisation problem that needs to be solved sequentially and not all objectives have the same priority. The following steps need to be taken for each app to determine the Far-field analogy apps:

1. Determine the recall of the goals with respect to the goals of the design problem at hand. If an app fulfils all the goals of the design problem at hand, then the recall is 1. If the app does not fulfil any of the goals of the design problem, then the recall is 0.
2. Select all apps with a recall of 1. If there are no apps with a recall of 1, then select the apps with the highest recall. These apps are most analogous to the design problem at hand.
3. Determine for each remaining app the recall of the goals with respect to all main goals (i.e., Goal 1 through 17 in Table 7.2). The recall is 1 if the app fulfils all available goals, whereas the recall is close to 0 if the app fulfils only one or two of those goals.
4. Select the apps with the highest recall. These apps tend to solve more design problems. It is reasoned that to solve those problems, a diverse set of functionalities is needed. Thereby, these apps may form a richer source of inspiration.
5. Determine of each of the remaining apps the keyword (dis)similarity. This can be done by for instance using (latent) semantic similarity (cf. K. Fu et al., 2013) or cosine similarity between the set of keywords listed for the app and those listed by the app designer.
6. Select the apps with the highest keyword dissimilarity.

A real-life application of the app selection method is discussed in the next chapter. The method is used there for selecting apps for the validation of our theory.

Chapter 8

Validation

The validation performed in this research was two-fold, as both main strands of the research were validated. The validation of the app review analysis strand focused on the performance of the feature opinion extraction. The validation of the analogical reasoning strand aimed at finding empirical evidence for the assumptions made and expectations formulated in the previous chapter.

8.1 App Review Analysis Validation

This area of the validation focused on the upper two rows of figure Figure 6.3: the accuracy of the extraction of feature opinion pairs. This was done by comparing the automatically extracted feature opinion pairs with a golden standard. This golden standard was created by manually annotating feature opinion pairs for a set of reviews. The feature grouping and the creation of the visualisation were not considered in this validation. The quality of the extracted feature opinion pairs influences those last steps in the automatic review analysis approach. Hence, it is important that the quality is good enough for the intended purposes. The postprocessing step was included, because this is an important step of the approach which aims are improving the quality. Also, this step was taken instead of prefiltering irrelevant sentences (Section 6.2.2).

8.1.1 Validation Approach

Apps. Eight apps were selected, which can be found in Table 8.1. For the full name of the apps, please consult Appendix E. For consistency, we selected a subset of the apps that were used for the analogical reasoning validation (as discussed later). The apps represent the type of apps that could be presented in the tool, as they were sampled largely in line with the preconditions listed Section 7.4.2. Two apps were selected for each of the four quadrants. These specific ones were selected while trying to create a rather diverse set of apps. This was done to be able to generalise the findings better. Apart from that objective, two highly similar apps were selected (i.e., the Near-field analogy apps) for being able to compare results of highly similar apps. The Sports Tracker app was selected despite having a rating lower than 4.5. Including this app instead of Amino would lead to a more diverse set, as both Amino and Travello were about socialising with communities. Houzz was not selected, since reviews for this app were already used during the creation of the script. Only eight apps were sampled, due to time and resource restrictions.

Reviews. The fifty most relevant reviews (at the time of extraction) were extracted from the Google Play Store, using the tool mentioned in Section 6.2. This sampling method was used

Table 8.1: Apps used for the validation of the automatic app review analysis. * = *At the time of selection.*

Name	Play Store category	Rating*	# downloads*	# reviews*
Event Manager	Events	4.7	10,000+	438
Forest	Productivity	4.5	10,000,000+	223,045
Travello	Travel & Local	4.5	500,000+	3,362
Sports Tracker	Health & Fitness	4.4	10,000,000+	218,909
Food Checklist	Shopping	4.7	10,000+	488
Calorie, Carb & Fat	Food & Drink	4.5	1,000,000+	53,107
Easy Recipes	Food & Drink	4.7	1,000,000+	14,293
Kitchen Stories	Food & Drink	4.7	1,000,000+	29,328

Table 8.2: Automatic app review analysis validation data description * = *Some version numbers were missing.* ** = *Including spaces.*

Name	Mean rating	Mean # characters**	Time span (days)	# versions
Event manager	4.4	96	652	7*
Forest	3.6	340	16	2
Travello	4.1	150	569	15*
Sports Tracker	3.3	219	72	4
Food Checklist	4.5	211	107	7
Calorie, Carb & Fat	4.2	201	69	1
Easy Recipes	4.1	62	609	18
Kitchen Stories	4.4	146	157	7

(i.e., as opposed to the newest reviews), as the in-depth manual analysis showed that this would result in a higher number of feature opinion pairs. Also, a quick visual inspection indicated that the quality of the reviews of that type would be higher as fewer spelling mistakes and poorly formatted sentences would be present. A visual comparison of two word clouds created using the two different sampling methods also indicated that the most relevant reviews would result in higher quality word clouds (Section 6.2.6). Only English reviews were selected, as the script was trained on English reviews.

As can be seen in Table 8.2, the reviews were posted over different time span. The time spans per pair of each quadrant varies considerably, which is incidental. Moreover, the reviews of the various sets were written about different versions of the app. The average rating given to the reviews in the set was considerably lower for most apps than the actual rating in Google Play Store as reported in Table 8.1. In total, only 400 reviews were analysed, mainly due to the constraints put on this validation.

Protocol. The validation was performed by only one annotator. However, most steps were double checked by a second annotator. The following steps were taken during this validation:

1. Manually annotate each review to create a golden standard. List for each app the feature opinion pairs while taking into account the guidelines listed in Appendix D. Each feature and each opinion is assigned to an individual column, as this is equivalent to the script output. Each new pair is listed in a new row.
2. Run the feature opinion extraction script. Use the output up till and including the post-processing step.
3. Gather both results side by side in a sheet.
4. Assess per feature opinion pair the accuracy in terms of true positives (TP).
5. Compute the precision and recall per app.
6. Compute the precision and recall over all apps.

Only the author of this thesis tagged all the reviews manually to create the golden standard. Beforehand, guidelines were created (Appendix D) to perform the tagging consistently. These guidelines were partly based on the knowledge obtained during the manual review analysis (Section 6.1) and partly on own insights, while taking the objectives of this research into account. These guidelines were updated during the tagging process, since new cases required new rules. As a result, various iterations took place. The first supervisor tagged a subset of the data to assess the agreement and to align the annotation protocol. This subset contained five reviews per app (i.e., 10% of all reviews). No rules were agreed on beforehand, in order to annotate the reviews in an unbiased manner. For each pair an agreement of 1 was assigned if both annotators tagged the exact same pair. An agreement of 0.5 was assigned if the pairs of both annotators were slightly different but still captured the same meaning. If cases with conjunctions were annotated differently (i.e., as one pair vs. as separate pairs), then this was still counted as a case of agreement. Initially, the agreement about the annotated cases was low (i.e., 34%). Cases of disagreement were discussed and resolved, and the protocol was updated. After that, the initial annotation was revised. Cases of doubt were double checked by the second annotator. If there was still doubt after that, the specific case was not annotated to prevent incorrect annotations. We believe that the automatic feature opinion extraction cannot be expected to work better than human annotators.

The assessment of the precision and recall was done in a similar manner to the creation of the golden standard. Only the author assessed the accuracy of all extracted and annotated pairs and the first supervisor assessed all cases of doubt. A set of guidelines was used as well for the assignment of TP values. These guidelines can also be found in Appendix D.

Analysis Metrics. To validate the performance of the automatic feature opinion extraction, the quality of the output was analysed by comparing the extracted pairs with the annotated pairs. Each extracted pair was counted as a true positive if it was part of the golden standard. The entire assessment was performed twice. The difference between the two assessments was the way in which subsets were handled. In many cases subsets of the annotated pair were extracted. Either the feature was a subset of the annotated feature, the opinion was a subset of the annotated opinion, or both the feature and opinion were a subset of the annotated pair. In the first and stricter assessment, the subsets were assessed on whether they were still useful or meaningful regarding their eventual use in the tool. In this way, some pairs that were not fully extracted like the annotated pair and that would still be useful for our objectives would not be discarded fully. Put differently, partly correct extracted pairs would not be counted as completely accurate. In this assessment, TP values of 1 were assigned in case the annotated pair was extracted or in which the extracted subset was still as meaningful as the annotated pair. In

specific cases in which a reasonable subset of the annotated pair was extracted, a TP value of 0.5 was assigned (consult Appendix D for more detail on these cases). The second assessment was done in line with the approach of Shah et al. (2019). In this assessment, subsets were handled more leniently. A TP value of 1 was assigned in all cases in which the extracted pair was a subset of the annotated pair. Thus, if either the feature or the opinion was a subset or if both were a subset, then it would be counted as a full TP. In this second assessment it was not taken into account whether the extracted subset was meaningful or not.

The precision and recall were determined per app and over all apps. The overall precision per app and over all apps was determined as follows (Powers, 2011):

$$Precision = \frac{\text{Annotated pairs} \cap \text{Extracted pairs}}{\text{Extracted pairs}} = \frac{\text{TP}}{\text{Extracted pairs}}$$

The precision was determined by dividing the number of TPs by the number of extracted pairs per app or in total. The overall recall per app and over all apps was determined by dividing the number of TPs by the number of annotated pairs per app or in total. This is reflected in the following formula (Powers, 2011):

$$Recall = \frac{\text{Annotated pairs} \cap \text{Extracted pairs}}{\text{Annotated pairs}} = \frac{\text{TP}}{\text{Annotated pairs}}$$

General Observations. For some apps (e.g., Calorie, Carb & Fat) it was considerably easier to describe the pairs. The users expressed their opinion about specific features in a more straightforward and simple manner (e.g., "I love the recipes"). The easier reviews contained less ambiguity regarding what the users were talking about. Unsurprisingly, the quality of reviews thus influences the quality of the golden standard.

8.1.2 Results

In general, all recall and precision values are pretty low. This is also reflected in the differences between the number of extracted pairs and the number of pairs that were part of the golden standard. There is a difference in performance between the two assessments. However, this difference is only small, as the precision and recall are 0.07 and 0.08 higher for the second assessment. This difference is not substantial, as only in 27 cases (i.e., 7%) an extracted subset was assigned a TP value of 0.5 or 0 in the first assessment. That is, only a limited number of subsets was assessed to be less or not meaningful in the first assessment. In most cases, either a completely wrong pair (false positive) was extracted or the annotated pair was not extracted at all (false negative). In case of the stricter assessment (Table 8.3), only the precision for Easy recipes is above a threshold of 0.5. In the more lenient assessment (Table 8.4), three apps obtained values above a threshold of 0.5. In both assessments, the precision and recall are highest for three of the four food or recipe related apps. These apps were also found to be easiest to annotate. Most of those cases discussed short features and short opinions. The highest precision and recall values were obtained for the app with the shortest reviews. However, the performance of the app with the second shortest reviews (i.e., Event manager) is considerably lower, indicating that the overall review length does not per se affect the performance. Moreover, the overall app rating and the average rating in the set do not seem to affect the performance, since there is not clear distinction in rating between better performing apps and worse performing apps.

Table 8.3: Results of the strict feature opinion extraction validation.

Name	# annotated pairs	# extracted pairs	# TP	Precision	Recall
Event manager	10	21	3	0.14	0.30
Forest	22	42	5	0.12	0.23
Travello	16	33	7.5	0.23	0.47
Sports Tracker	25	24	5	0.21	0.20
Food Checklist	25	30	4.5	0.15	0.18
Calorie, Carb & Fat	57	41	17.5	0.43	0.31
Easy Recipes	45	37	22	**0.59**	0.49
Kitchen Stories	53	46	21	0.46	0.40
Total	**253**	**274**	**85.5**	**0.31**	**0.34**

Table 8.4: Results of the mild feature opinion extraction validation.

Name	# annotated pairs	# extracted pairs	# TP	Precision	Recall
Event manager	10	21	4	0.19	0.40
Forest	22	42	7	0.17	0.32
Travello	16	33	9	0.27	**0.56**
Sports Tracker	25	24	10	0.42	0.40
Food Checklist	25	30	7	0.23	0.28
Calorie, Carb & Fat	57	41	20	0.49	0.35
Easy Recipes	45	37	23	**0.62**	**0.51**
Kitchen Stories	53	46	25	**0.54**	0.47
Total	**253**	**274**	**105**	**0.38**	**0.42**

8.1.3 Discussion of the Results

The overall low recall value is not to our surprise, since only a limited set of patterns was used to create the script. The feature patterns only covered 57% of all cases and the opinion patterns only covered 62% of all cases in our manual analysis (Section 6.1.2). Taking the product of both would indicate that at best a recall of approximately 0.35 could be obtained. Section 6.1.2 explained that some of the longer patterns could also be shortened to the ones that were used to create the script. Hence, the recall could be a bit higher than 0.35, but certainly not near 1. Currently, extracted pairs with a neutral sentiment are filtered out. It could be the case that actual, correct pairs are assigned a wrong sentiment and are mistakenly excluded from the set. This was for instance the case for opinion words such as 'simple' as this word was assigned a neutral sentiment score. In some cases, it initially seemed surprising that a pair was not extracted. A visual inspection showed that the automatic annotation contained flaws. Hence, the quality of the automatic annotations (e.g., dependency parsing and sentiment analysis) have influence on the recall. All in all, obtaining a high recall was not aimed for as the quality of the

extracted pairs was of higher importance for the eventual tool. So, the low recall is not seen as a large limitation here.

Unfortunately, the precision was below expected. This low value can be attributed to several issues. The foremost reason for the low precision is the fact that the general POS patterns cause the script to extract all aspect opinion pairs. Only feature opinion pairs were of interest, which is a stricter subset of the former. Furthermore, using patterns only comprising unigrams and bigrams for the features and opinions caused that a substantial amount of pairs could not be (fully) extracted or that too much contextual information was missing (as foreseen in Section 6.1.2). Even in some cases, the annotated opinion was extracted as a complete pair. The sentiment analysis could also cause that some pairs are extracted, while they should not have been because of an actual neutral sentiment. This was already partly taken into account in the post-processing dictionaries. However, all written dictionaries are currently small, meaning that many irrelevant sentences are still analysed. These irrelevant sentences cause that many aspect opinion pairs or other things are extracted, while those are not of interest. Faulty annotation of some review sentences or phrases is also of influence on the precision, as some incorrect pairs were extracted in line with the predefined patterns. Lastly, the quality of the review itself was in various cases low. Even in some cases it was hard for the human annotators to extract pairs from some reviews. This low quality also influenced the eventual quality of the results.

Comparing the raw numbers, one can observe that the performance is considerably lower than the results obtained in other studies. However, there are differences in the approaches that each study took. The performance is most substantially lower than the results of Gu and Kim (2015) (i.e., the most comparable app review analysis research). They obtained average F1-scores for the extracted aspects and extracted opinions of 0.85 and 0.84, respectively. In their work, they use the term aspects and not features. It seems that they focus on a less strict set of things about which the user has an opinion. However, this is not completely clear, as we could not find a clear definition of the aspects in their work. Furthermore, their approach for extracting those pairs was more advanced than ours. This is mainly attributable to the difference in research scope. Furthermore, the irrelevant review sentences that introduced noise are filtered out automatically beforehand in their approach. In the validation, they only focused on sentences labelled as “aspect evaluation”. We expect that when those irrelevant review sentences are filtered out similarly in our approach, that the results would substantially improve. Our results are closer to those of Guzman and Maalej (2014). Their approach for extracting features managed to achieve an average precision of 0.59 and an average recall 0.51. Again, our values are reasonably lower. This could partly be attributed to the fact that we focus on pairs of features and opinions, which is more challenging than extracting individual features. Our results are closest to those of the reproduction the SAFE approach validation of Johann et al. (2017) performed by Shah et al. (2019). Their extraction of 2-to-4 word features achieved an average precision and recall of 0.12 and 0.54 respectively. Their extraction of all features achieved an average precision and recall of 0.21 and 0.42 respectively. So, the quality of our extracted pairs is on average higher than their extracted features, whereas the average recall is somewhat comparable.

All in all, the results from this validation and the word clouds presented in Section 6.2.6 give a first indication of the feasibility of analysing app review in the context of fostering creativity.

8.1.4 Threats to Validity

Only one researcher created the golden standard and performed the full annotation of the extracted pairs. Hence, other researchers might have annotated it differently and mistakes could go unnoticed. As a mitigation, a second researcher was involved in the annotation process. All guidelines were double checked and cases of doubt were resolved in discussion. Furthermore, having the second researcher also annotating a subset of reviews also improves the reliability of the results. Furthermore, several iterations were performed and all outcomes were double checked to ensure that there were no prominent mistakes left. The creation of the golden standard was a human and subjective task, meaning that this could lead to possible inconsistencies. The quality of various app reviews was poor, making it hard to assess to what something was referring. On top of that, conjunctions or poor formatting introduced ambiguity, making it harder to determine which parts need to be together in a feature or opinion. The guidelines were in place to prevent inconsistencies and wrong annotations as much as possible. We recognise the fact that the script was not tested on reviews of low-quality apps (i.e., apps with a low overall rating) that could affect the quality of the output. This is not seen as a threat here, since the script is only used for showing examples of apps with high ratings. Finally, a limited number of reviews of a limited number of apps was assessed. Hence, the results have a limited generalisability. This effort is only seen as an initial validation and an initial proof of concept. Future work could help to further mitigate this threat.

8.2 Analogical Reasoning Validation

Section 2.2.5 touched upon the fact that a small experiment was conducted for the validation of the analogical reasoning strand. A small experiment was chosen for this for several reasons. This approach was assessed to be most feasible given the available resources. Also, an experiment would allow best to answer the research questions. Furthermore, compared to, for instance expert interviews, this method would allow for obtaining quantitatively analysable empirical data. The design of the experiment is described in this section based on the experiment planning steps of Wohlin et al. (2012).

8.2.1 Experiment Design

Context. The experiment was conducted at the end of the academic year 2019-2020. It was chosen to conduct the experiment online through a questionnaire. In this way, potentially many people could be reached. Also, due to circumstances[1], inviting participants in a real-life setting was not possible. As the experiment could not take place in a controlled environment, the experiment design was adapted to make it appropriate for the context in which it would take place. Instead of letting participants actually execute a specific task, it was decided to let them assess example apps with a specific design task in mind and ask about their perceptions regarding those examples (as discussed later). The questionnaire was created in Qualtrics, the default survey tool within Utrecht University.

Hypotheses & Variables. The main goal, questions, and metrics of this study are captured in the GQM model of Basili, Caldiera, and Rombach (1994) (Figure 8.1). As can be seen, it was

[1]While the execution of this research took place, the global pandemic COVID-19 emerged.

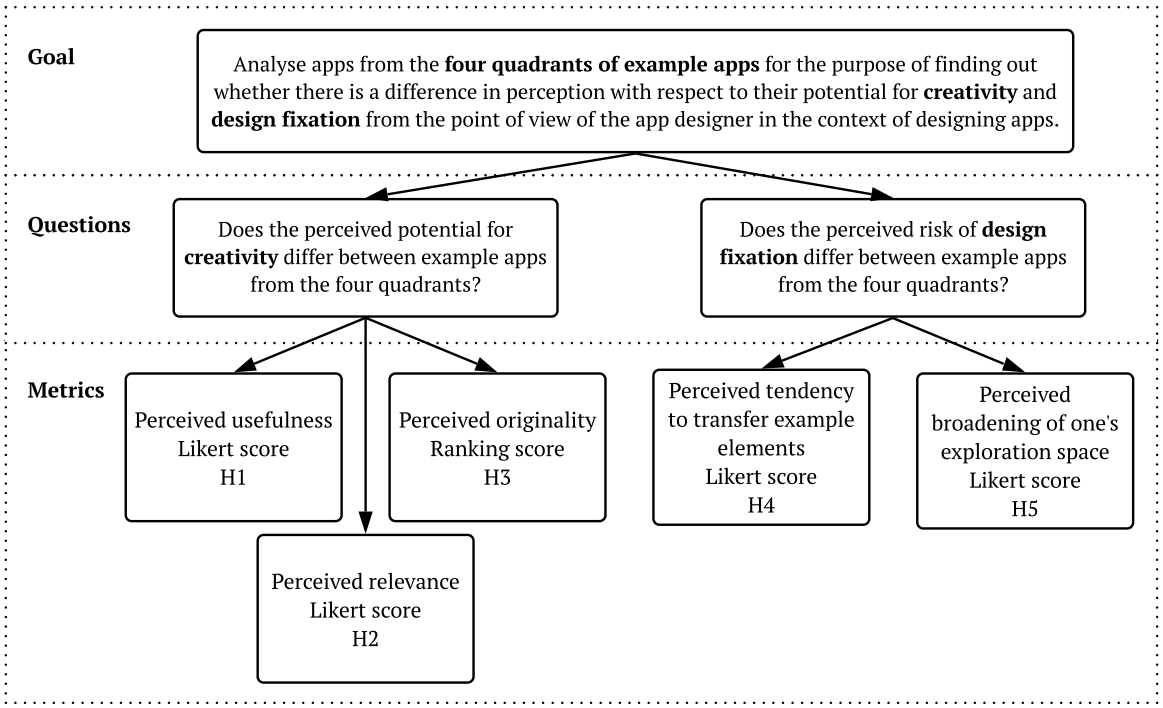

Figure 8.1: The GQM model of this study. Based on the work of Basili et al. (1994).

chosen to use the terms perception, potential, and risk, since the design of the experiment did not allow for measuring the actual creativity and design fixation of app designers. Chapter 3 described that there is little consensus on both constructs and on how to measure them. In total five variables were selected that intend to measure the two constructs. Relevance, usefulness, and originality were derived from the definitions of creativity given in Section 3.1. However, different phrasings were used for novelty and appropriateness to make them more understandable for the participants. Originality was an approximation of novelty and relevance was an approximation of appropriateness. Originality is also used in related work on creativity (see e.g., Mohanani et al., 2014; Christensen and Schunn, 2009). Relevance is often discussed in the context of analogical reasoning (see e.g., Gentner, 1989; Gick and Holyoak, 1980). The aim to study the influence of relevant examples on creativity is not unique, since already various research efforts have been devoted to that (Christensen & Schunn, 2009). Relevance and usefulness of the presented examples may at first seem redundant. However, we argue that examples may be relevant in the context the design task at hand, while not being useful for generating novel ideas for that task (and vice versa).

The risk for design fixation was operationalised with two variables that correspond to our two interpretations of the construct, namely the tendency to literally transfer example elements and the perceived broadening of one's exploration space. The latter must be interpreted inversely in the final conclusions, meaning that when the perceived broadening of one's exploration is high the risk for design fixation is low. As can be seen in Figure 8.1 each variable is associated with one set of hypotheses.

The hypothesis for the perceived relevance is as follows:

- $H1_0$: Apps from the Surface similarity and Near-field analogy quadrants do not have a higher *perceived usefulness* than apps from the Potential serendipity and Far-field analogy quadrants.
- $H1_A$: Apps from the Surface similarity and Near-field analogy quadrants have a higher *perceived usefulness* than apps from the Potential serendipity and Far-field analogy quadrants.

This set of hypotheses was phrased following the results from the experiments of Reed (1987), which indicated that people tend to assess surface similar examples to be more useful than examples that are not similar on the surface in the context of problem solving (i.e., word problems). In that research, they also compared four classes of examples that were also categorised on structural and surface similarity. We hypothesised that their results also apply to the app design field and that participants would judge the usefulness on the basis of surface elements. Thus, the two lower quadrants were expected to be perceived more useful than the two upper quadrants.

- $H2_0$: Apps from the Surface similarity and Near-field analogy quadrants do not have a higher *perceived relevance* than apps from the Potential serendipity and Far-field analogy quadrants.
- $H2_A$: Apps from the Surface similarity and Near-field analogy quadrants have a higher *perceived relevance* than apps from the Potential serendipity and Far-field analogy quadrants.

This set of hypotheses was composed on a similar basis as the previous set, namely based on surface similarity. According to Gick and Holyoak (1980), people may be unsuccessful in noticing relevant analogies. The study of Gentner et al. (1993) indicates that examples with a higher surface similarity are more easily accessible than examples with lower surface similarity. Christensen and Schunn (2009) also argue that people may assess far fields as less relevant as a result of low surface similarity. This was also indicated by the results of the study of K. Fu et al. (2013). Thus, we hypothesised in line with the previous set of hypotheses that subjects focus more on the surface elements than on the analogical structure in assessing the relevance of examples. Hence, the apps from the two lower quadrants from the matrix were expected to obtain a higher perceived relevance than apps from the two upper quadrants. Note that we expect that actual usefulness and actual relevance would be influenced more heavily by the structure than by the surface, since those solve the same underlying problem as the design case at hand (Christensen & Schunn, 2009; Gentner et al., 1993).

- $H3_0$: Ideas resulting from assessing apps from the Potential serendipity and Far-field analogy quadrants are not *perceived* to be more *original* than ideas resulting from assessing apps from the Surface similarity and Near-field analogy quadrants.
- $H3_A$: Ideas resulting from assessing apps from the Potential serendipity and Far-field analogy quadrants are *perceived* to be more *original* than ideas resulting from assessing apps from the Surface similarity and Near-field analogy quadrants.

In contrast to the previous hypotheses, we hypothesised that presenting examples that are less surface similar will lead to more original results. Results from the study of Dahl and Moreau (2002) indicated that originality of design outcomes is positively influenced by making far-field

analogies. When near domains are presented, it seems that it is more difficult to make those far-field analogies (Christensen & Schunn, 2009). In other words, presenting domains that are near may negatively impact the originality. Gentner and Markman (1997) described that people may be better able to observe the creativity of farther domains. We expect that this principle also applies to originality.

- $H4_0$: The *perceived tendency to literally transfer elements* from apps from the Surface similarity and Near-field analogy quadrants is not higher than the perceived tendency to literally transfer elements from apps from the Potential serendipity and Far-field analogy quadrants.
- $H4_A$: The *perceived tendency to literally transfer elements* from apps from the Surface similarity and Near-field analogy quadrants is higher than the perceived tendency to literally transfer elements from apps from the Potential serendipity and Far-field analogy quadrants.

In line with the reasoning above, we expect that it will be easier for people to literally transfer ideas from apps with a similar theme (i.e., surface similar apps) than apps with a different theme. It may be easier for people to come up with ideas that are similar to the ones they saw in the examples of close domains. For instance, when assessing examples of food and recipe apps when creating a recipe app, we expect that people may tend to apply the functionalities of the examples (e.g., a calorie log) to their own design. Christensen and Schunn (2009) also suggested presenting far-field domains when it is one's intention to come up with ideas that do not resemble the examples. They argued that near-field examples may lead to an increased amount of example elements used in the design outcome. Therefore, we hypothesised that apps from the two lower quadrants will have a higher perceived tendency of transferring example elements than the upper quadrants.

- $H5_0$: Apps from the Potential serendipity and Far-field analogy quadrants are not *perceived to broaden one's exploration* space more than apps from the Surface similarity and Near-field analogy quadrants.
- $H5_A$: Apps from the Potential serendipity and Far-field analogy quadrants are *perceived to broaden one's exploration space* more than apps from the Surface similarity and Near-field analogy quadrants.

Lastly, we hypothesised that examples that are further away from their own design problem may help people to break away from their own thinking patterns. Presenting distant domains may allow to explore a broader range of options (Christensen & Schunn, 2009). As the distance between the domain and the design problem at hand is measured through the keyword similarity, we expected that the upper two quadrants will help people to explore more options than they would have without any examples. Also, whereas near-field analogies may help in successfully solving the design problem, they may not be helpful in finding novel ways of doing that.

The independent variables of this experiment were 1) the four quadrants and 2) two sets of example apps. The first independent variable has four levels, namely apps from each of the four quadrants. The second independent variable was solely used to counterbalance the possible, unwanted impact of a specific selection of apps. The dependent variables can be found in Figure 8.1 and in the hypotheses above.

Subject Selection. Beforehand, we aimed at getting a response rate of at least thirty or forty, as this would seem feasible given the time and resource constraints. At first, only undergraduate and graduate Information Science students were recruited. These subjects were assessed to be appropriate, as most had gained some experience in app design. However, due to the limited response rate, additional subjects needed to be sampled. For this, the online platform Reddit was used to recruit more subjects. Several communities about android apps, app design, or android development were used for recruiting subjects. These communities were assessed to be fitting, since we reasoned that our target audience could be member of this community. As can be inferred, convenience sampling was used in both cases as a sampling method. Two different invitation hyperlinks were used, to be able to differentiate between the samples afterwards if needed. These invitation links to the questionnaire were anonymous links to maintain the subjects' privacy. The responses were automatically anonymised by the tool to further ensure the participants' privacy. Thus, no personal data was gathered. The participation was on a voluntary basis and the subjects did not receive any reward for their participation. Unfortunately, in the end only twelve subjects (n = 12) participated in the experiment.

Experimental Design & Procedure. Having two independent variables with two and four factors resulted in a 2 x 4 Set of apps (between) X App quadrant (within) design. Each participant received two example apps from each of the four quadrants. This would allow participants to compare results between the quadrants and this would require less participants to be recruited. In total, sixteen examples were selected for the experiment. Beforehand, subjects were asked to rate their level of app design experience as either *novice*, *advanced*, *intermediate*, or *expert*. A question was asked about expertise, since literature indicated that expertise may be of influence on analogical reasoning (see e.g., Ball, Ormerod, and Morley, 2004). Also, this variable was useful for describing the sample. All subjects were then asked to read a description of a toy design case and their task instructions. The participants did not actually need to execute the design task, but they were asked to assess example apps as if they needed to list ideas for the design task. No hints were provided regarding the possible analogies in the task instructions, since we aimed at understanding the unaffected perception of subjects. However, high-level requirements that the app of the toy design task needed to fulfil were provided to give subjects boundaries for their design task. It was explicitly noted that their ideas did not need to be limited to those requirements. To address a possible evaluation apprehension, it was also specifically stated beforehand that the questions were about perception and that no wrong answers could be given. After reading the instruction and design case, each subject was randomly assigned by the tool to one of the groups. The tool presented links to two examples with four related questions for each quadrant, which was repeated four times. Links to the app pages on the Google Play Store were provided, instead of creating a custom representation. The reason for this was to ensure that each app would be presented in the same format. Moreover, as could be read in Section 3.4.1, the way in which examples are presented may be of influence. We decided not to focus on finding the most optimal example representation, as this was outside the scope of our research. The order in which the subjects received the examples from the quadrants was randomised as well to counterbalance a learning effect. Finally, subjects were asked to rank the four quadrants on originality of generated ideas and to provide the apps they were already familiar[2] with. Again, familiarity with examples could influence creativity and design fixation (Section 7.4.2).

[2]Familiarity constituted here either having used the app or having encountered the app before.

Instrumentation. Subjects were provided with an information sheet at the beginning of the questionnaire. They were required to read this and give their informed consent before being able to participate in the study. As mentioned above, each participant received the same design task and instructions. The sets of apps differed between the two groups. The questions the subjects needed to answer were equivalent. In total, each participant was asked to answer nineteen short questions. The questionnaire, the set of selected apps, and the app selection procedure can be found in Appendix E.

For the sixteen questions about the examples from the four quadrants, seven-point Likert scales were used. Seven-point scales are found to be best option compared to shorter or longer scales (Krosnick & Presser, 2010). Likert scales were chosen, since those are often used in questionnaires and may thereby be easy to use for the subjects. Only one question per variable was used in this questionnaire for a couple of reasons. We found that adding more questions about the same variable would lead to redundant rephrasings, which was expected to confuse the subjects. Also, using one question per variable is found to be acceptable in specific cases and a better option than using two or three questions per variable (Diamantopoulos, Sarstedt, Fuchs, Wilczynski, & Kaiser, 2012). We believe that our case was in line with the guidelines of Diamantopoulos et al. (2012). It was chosen to use a ranking question for the originality, since originality must be compared to a base reference. Comparing the originality of the generated ideas for one example to those of other examples was thereby suitable.

Pretest. A small pretest was conducted to assess whether the experiment would be understandable for participants and whether the questionnaire was of good quality. The pretest was conducted with four researchers of Utrecht University. In general, no big issues arose. Some suggestions were given regarding the format of the questionnaire. These were incorporated accordingly.

8.2.2 Experiment Results

As can be seen in Table 8.5, the two groups (i.e., relating to the two different sets of apps) were fairly equally distributed. In general, the participants completed the experiment in around six minutes. The median duration is reported here, since there was one outlier who took several days to complete the experiment. Probably, this participant opened the questionnaire days before actually working on it. This median duration was substantially lower than the prescribed and expected duration (i.e., ten to fifteen minutes). Most participants from both groups were not very experienced in designing apps. 50% of both the students and the Reddit users reported to be a novice app designer. Furthermore, in each sample of participants, only one reported to be an advanced app designer.

Seven of the participants reported not to be familiar with any of the apps. In total, ten of the apps were familiar to the participants. Six of those were only familiar to one person and four of those to two persons. Three participants reported to be familiar with one or two of the apps. Those two apps were not from the same quadrant. The two remaining participants reported to be familiar with either four or five apps. In both cases, they were familiar with both apps from the Near-field analogy and Surface similarity quadrants.

Since there were very few responses (i.e., twelve) it was decided not to conduct any statistical tests. Therefore, only the descriptive statistics and charts of the results are reported and

Table 8.5: Descriptive statistics of the experiment sample and execution.

	Median duration (min)	**# novice**	**# intermediate**	**# advanced**	**# expert**	**# student**	**# Reddit user**
Group 1	5.3	3	2	1	0	3	3
Group 2	6.2	3	2	1	0	5	1
Total	5.8	6	4	2	0	8	4

assessed to gather evidence for our hypotheses. We shall not draw formal conclusions regarding the rejection of the hypotheses, since there is not enough evidence for this. Therefore, only tentative conclusions are drawn here. Furthermore, a more in-depth analysis of the influence of experience and familiarity was not conducted, due to the small sample size. For the same reason, a separate analysis of the two groups was not conducted. This does not have negative influence, since the characteristics of both groups were fairly even.

Table 8.6: Median results per quadrant. Value assigned to each label between brackets.

Variable	**Potential serendipity**	**Far-field analogy**	**Surface similarity**	**Near-field analogy**
Perceived usefulness	Somewhat disagree (3)	Somewhat agree (5)	Somewhat agree (5)	Agree (6)
Perceived relevance	Disagree (2)	Somewhat agree (5)	Somewhat agree/ Agree (5.5)	Agree/ Strongly agree (6.5)
Perceived example element transfer	Somewhat disagree/ Disagree (2.5)	Somewhat disagree (3)	Somewhat agree (5)	Somewhat agree (5)
Perceived exploration space broadening	Somewhat disagree/ Disagree (2.5)	Somewhat agree/ Agree (5.5)	Somewhat agree/ Agree (5.5)	Agree (6)

Perceived usefulness. The medians reported in Table 8.6 show some difference between the four quadrants for this variable. The difference between the Potential serendipity quadrant and the other three quadrants is substantial. The median for the Far-field analogy and Surface similarity quadrants are equivalent and their difference with the Near-field analogy quadrant is small. Figure 8.2 does indicate a slight difference between the Far-field analogy and Surface similarity quadrants. The two upper quadrants seem to be perceived a bit less useful than the two lower quadrants. Looking only at the more outspoken attitudes (i.e., strongly (dis)agree and (dis)agree) does not show a clearer difference between the upper quadrants and lower quadrants. Those results do indicate that participants perceived the Potential serendipity apps not very useful, with 42% of the participant disagreeing and only 8% agreeing. In contrast, partici-

pants perceived the Surface similarity and Near-field analogy apps as useful, with an agreement of 42% (vs. 17% disagreement) and 67% (vs. 8% disagreement) respectively. However, the attitudes regarding the Far-field analogy quadrant are in this case mixed, with both 33% positive and negative attitudes. The aggregated results of the upper and lower quadrants (Figure 8.3) seem to provide some evidence that apps from surface dissimilar quadrants are perceived as less useful than apps from the surface similar quadrants. However, the results of the surface dissimilar quadrants are most strongly influenced by Potential serendipity quadrant. So, the results indicate some differences between the two upper and two lower quadrants. However, there is not enough evidence to reject the null hypothesis (H1). Nevertheless, the findings are not contrasting with our expectations.

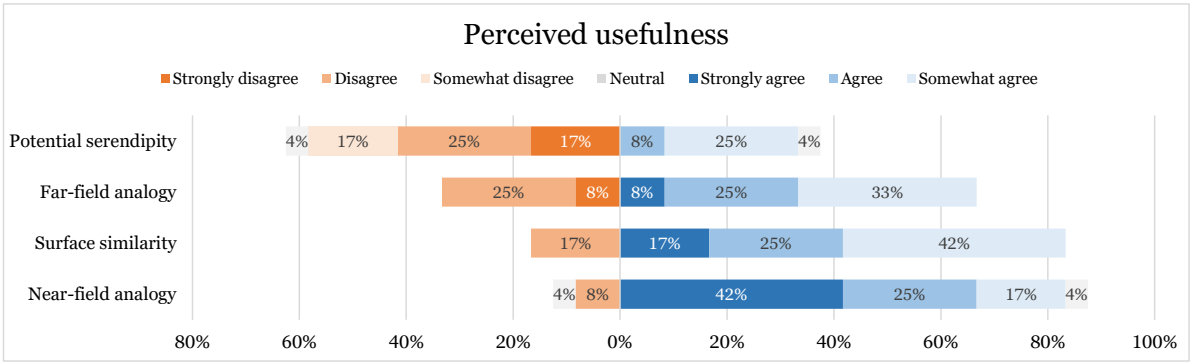

Figure 8.2: Results on the perceived usefulness[3].

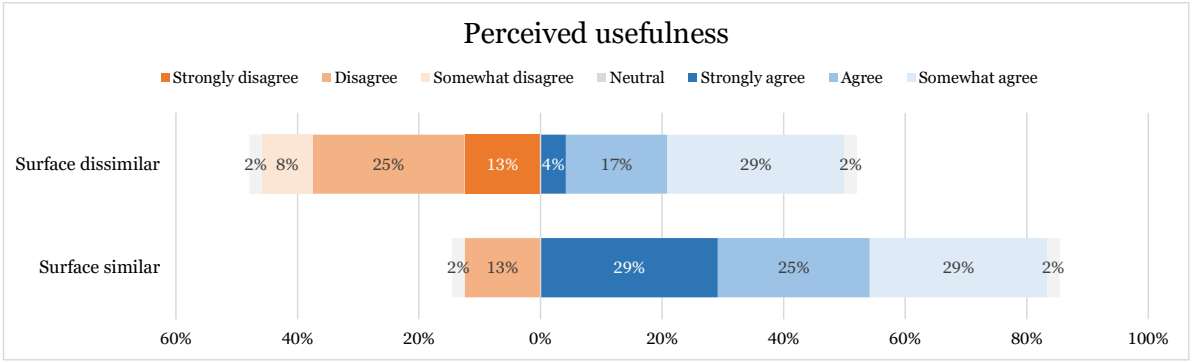

Figure 8.3: Aggregated results on the perceived usefulness[3]

Perceived relevance. The median values reported in Table 8.6 again indicate that there is some difference between the four quadrants. Again, the main difference is found between the Potential serendipity quadrant and the other three quadrants. At a glance, Figure 8.4 shows a difference between apps of the four different quadrants. Mainly the Potential serendipity quadrant differs from the other three quadrants. Again, the Far-field analogy quadrant shows more mixed re-

[3]The strongest attitudes are presented in the center for comparability purposes.

sponses than the other quadrants. When looking at the more outspoken responses, a possible difference between the upper quadrants and lower quadrants becomes more apparent. Participants perceived the Potential serendipity overall not very relevant for the design task at hand, with a disagreement of 66%. In contrast, most participants perceived the Surface similarity and Near-field analogy as relevant, with an outspoken agreement of 50% and 83% respectively. The Far-field analogy falls in between, with 25% of the participants outspokenly disagreeing and 17% outspokenly agreeing. The less strong opinions (i.e., somewhat (dis)agree), show an even less clear case for the Far-field analogy quadrant. Thus, the participants seem to have mixed opinions regarding the relevance of the apps from the Far-field analogy quadrant. By aggregating the results of the upper and lower quadrants (Figure 8.5), one could say that the evidence towards rejecting the null hypothesis becomes stronger. The difference between the upper and lower quadrants are clearer. However, the results on surface dissimilar apps is again influenced most strongly by the Potential serendipity quadrant. Thus, there is not enough evidence towards rejecting the null hypothesis (H2). Still, these initial findings are in line with our expectations that the apps from the lower quadrants are perceived to be (slightly) more relevant.

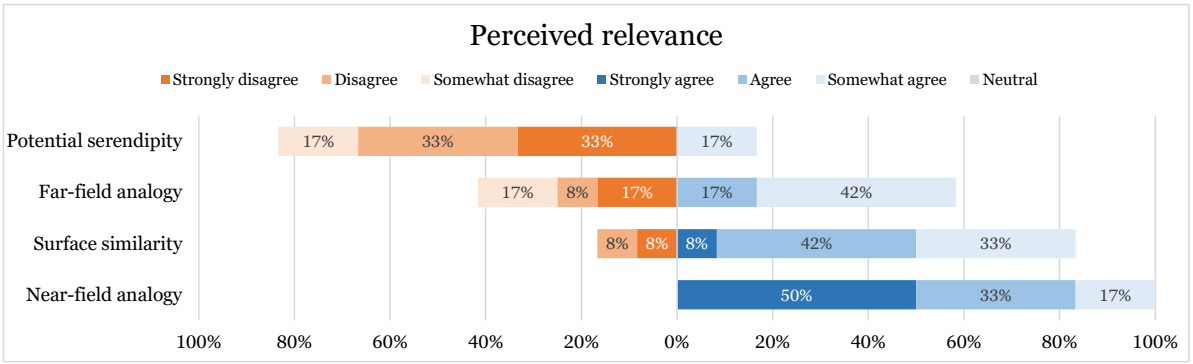

Figure 8.4: Results on the perceived relevance[3].

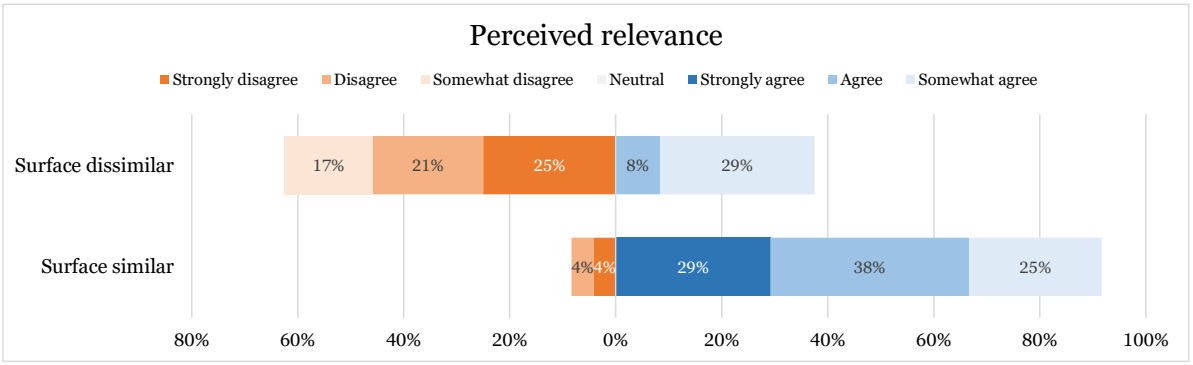

Figure 8.5: Aggregated results on the perceived relevance[3].

Perceived originality. When analysing the rank of each quadrant regarding the originality of the ideas (Table 8.7), the initial results do not seem to be in line with our expectations. The ideas coming from assessing apps of the Near-field analogy quadrant are perceived as most original (i.e., both median and mode), whereas the ideas coming from the Far-field analogy and Potential serendipity quadrant are perceived as least original (i.e., both median and mode). Only the difference in the median rank between the Near-field analogy quadrant and the other three quadrants is reasonable.

Table 8.7: Originality ranking per quadrant.

Measure	Potential serendipity	Far-field analogy	Surface similarity	Near-field analogy
Median	3	3	2.5	1
Mode	4	3	2	1

Figure 8.6 and Figure 8.7 generally indicate that assessing apps from the two upper quadrants (i.e., surface dissimilar quadrants) is perceived to generate less original ideas compared to the two lower quadrants. However, the opinions on the Far-field analogy quadrant are mixed as each rank was assigned by roughly a quarter of the participants. These results seem thus contrary to our expectations and give some indications towards a failure to reject the null hypothesis (H3). However, since evidence is only limited here, this is not formulated as a conclusion.

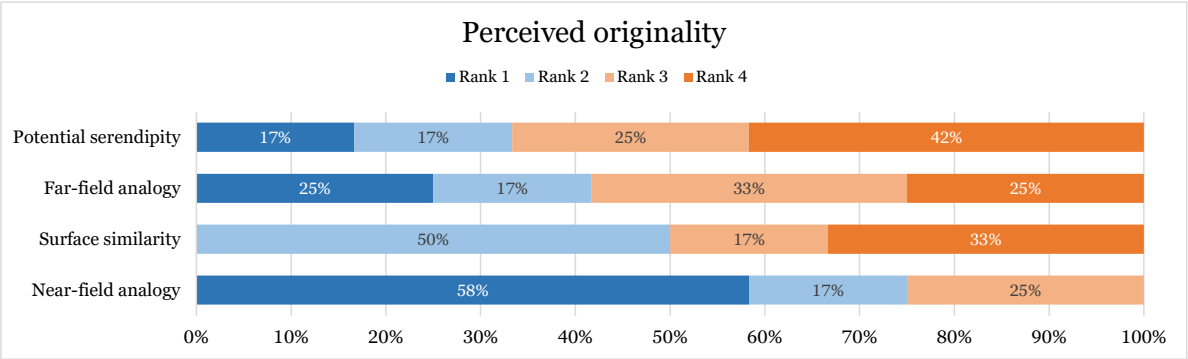

Figure 8.6: Perceived originality ranking results.

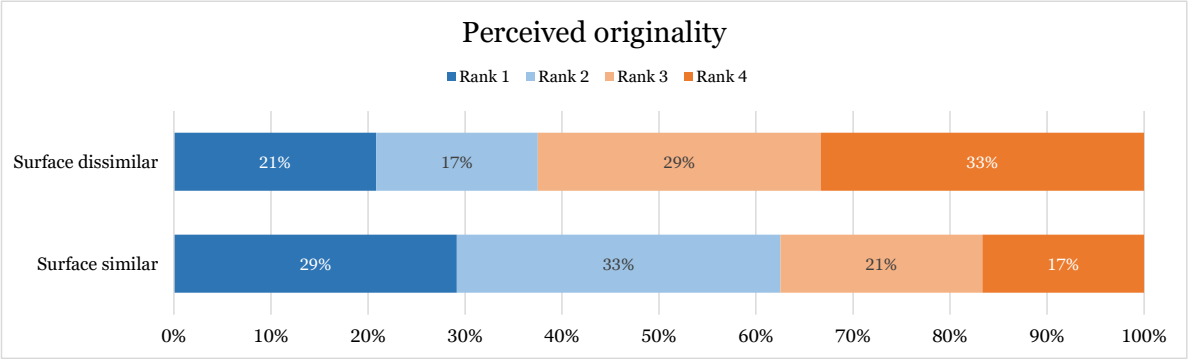

Figure 8.7: Aggregated perceived originality ranking results.

Perceived transfer of example elements. The median values reported in Table 8.6 indicate that there is a difference between the surface similar quadrants and the surface dissimilar quadrants. For this variable, the gap between the upper quadrants and lower quadrants is more substantial. The values indicate here that participants tend less to literally transfer example elements from the two upper quadrants compared to the two lower quadrants. These differences can also be observed in Figure 8.8. Analysing the stronger attitudes leads to less clear differences between the quadrants. Participants only reported quite strongly that they did not perceive to literally transfer example elements after assessing the Potential serendipity quadrant, with a disagreement of 50% and an agreement of 8%. In contrast, participants expressed more mixed attitudes for the other three quadrants. The strong positive and negative attitudes even outweigh each other for both the Far-field analogy and Near-field analogy quadrants. The aggregated results (Figure 8.9) also show some, but not strong, evidence that people tend to literally transfer more example elements from the apps of the lower quadrants. However, it seems that in this case the Surface similarity quadrant influences the results for the surface similar quadrants most strongly. All in all, there is again too little evidence for rejecting the null hypothesis (H4). Nevertheless, the findings are largely in line with our expectations.

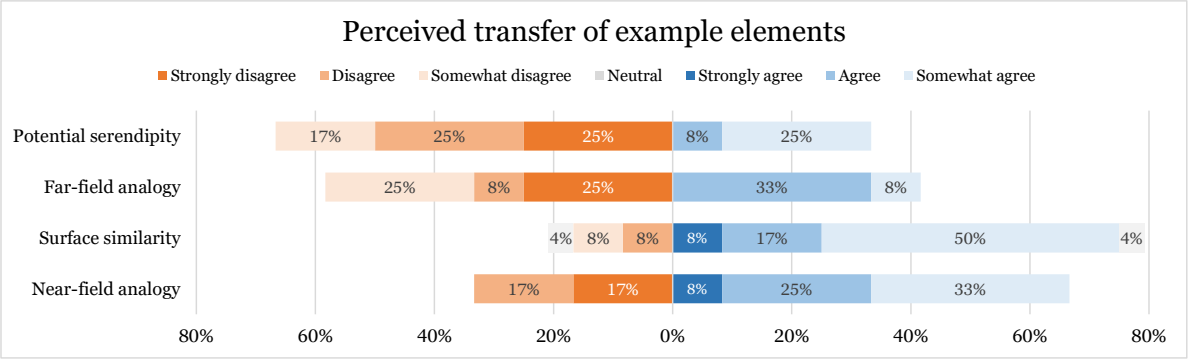

Figure 8.8: Results on the perceived tendency to transfer of example elements[3].

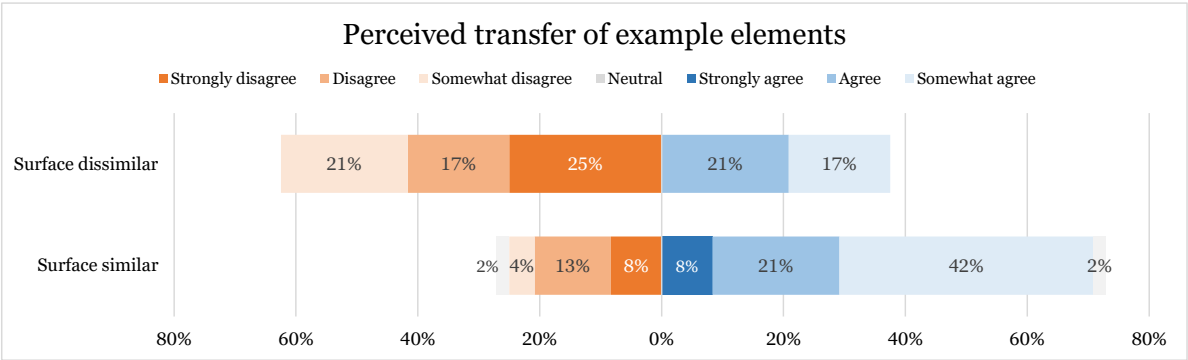

Figure 8.9: Aggregated results on the perceived tendency to transfer of example elements[3].

Perceived broadening of one's exploration space. The median values reported in Table 8.6 are again less distinctive, which similar to the results on the perceived usefulness and relevance. Again, the difference between apps from the Near-field analogy quadrant and the Potential serendipity quadrant is reasonable. However, the same median values are found for the Far-field analogy and Surface similarity quadrants. Also Figure 8.10 shows only small differences between the four quadrants. The strong attitudes indicate that apps from the Potential serendipity quadrant did not really broaden the participants' exploration space, with a disagreement of 50% and an agreement of 17%. The strong attitudes indicate that, contrary to our expectations, apps from the Near-field analogy quadrant do broaden one's exploration space, with a majority agreeing (i.e., 75%) and a minority disagreeing (i.e., 8%). However, again, the perceptions on the Far-field analogy and the Surface similarity quadrants are less distinctive. The aggregated results (Figure 8.11) seem to provide evidence that there may be a difference between the surface dissimilar and surface similar quadrants. However, these are most strongly influenced by the Potential serendipity and Near-field analogy quadrants. Hence, there is too little evidence for rejecting the null hypothesis (H5). Despite that, these tentative findings are contrary to our expectations.

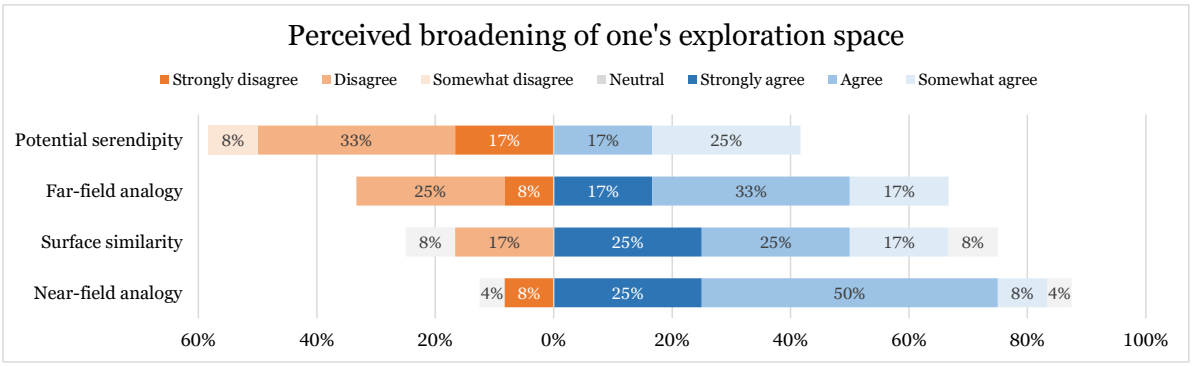

Figure 8.10: Results on the perceived broadening of one's exploration space[3].

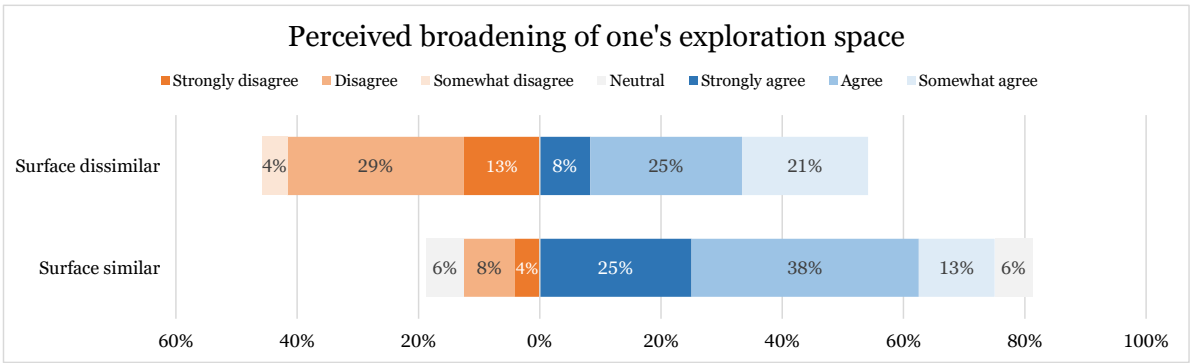

Figure 8.11: Aggregated results on the perceived broadening of one's exploration space[3].

8.2.3 Discussion of the Results

For none of the dependent variables enough strong evidence could be accumulated to give strong indications about the null hypotheses. We assume that because only few responses were gathered, differences may be less articulated. However, in most cases the findings seemed to be (somewhat) in line with our expectations or at least not contrary to our expectations. For the perceived relevance, perceived usefulness, and perceived tendency to literally transfer example elements, evidence was found that is consistent with our expectations. The small differences in the results for the perceived relevance and usefulness indicates that indeed the relevance and usefulness are not perceived as exactly the same. The findings for the perceived originality with respect to the generated ideas were the opposite of our expectations. There could be several reasons for this. For instance, at first, we thought that the order in which the quadrants were listed could be of influence. However, this order was randomised and participants could not continue until at least one bar was moved (a bit). So, we do not expect that these findings are a result the order in which the quadrants were listed. The participants only spent a limited amount of time to answer the questions. This may have influenced these results in particular, as they may only have generated few ideas when assessing the apps. It could also have been the case that the participants perceived their ideas to be more original than they actually might have been. It could, for instance, have happened that the participants assessed the originality in the light of the design task at hand, instead of in general. The findings on tendency to literally transfer ideas and perceived originality also seem contradicting, but can be explained by this assumption. Also, assessing originality may be difficult for a hypothetical task without a clear reference. Currently, the participants assessed the originality of ideas compared to the generated ideas from other quadrants. An experiment in which design outcomes are actually assessed is needed to give an indication about the actual originality instead of the perceived originality. The findings on the broadening of the exploration space were also different from our expectations. Again, it could be the case that the participants assess this question in the light of the specific context. So, potentially, participants were specifically searching for suitable ideas for their design task. This assumption is reflected by the fact that most participants found the surface similar apps more relevant and useful. It could also be the case that the familiarity influenced the results of this variable. However, most of the familiar apps came from the quadrants than were found to broaden the exploration space most (e.g., Surface similarity and Near-field analogy).

The results with respect to the quadrant that was of main interest (i.e., Far-field analogy) are mixed. For most of the dependent variables, the disagreement and agreement are highly similar. When assessing the stronger attitudes, the results seem even more mixed. The stronger attitudes on perceived usefulness and perceived tendency to transfer example elements even fully outweigh each other. It is at least positive to observe that the results on this quadrant are not fully the opposite of our expectations.

Finally, the results need to be discussed in the light of creativity and design fixation and further implications for the tool. It is difficult to draw conclusions about those constructs based on the gathered evidence. Some tentative conclusions can be drawn for individual quadrants. For instance, it seems that one may need to refrain from providing apps from the Surface similarity quadrant, when one wants to prevent design fixation in terms of copying example elements. Participants report in this case to not even unconsciously transfer example elements. If it is one's intention to give examples without giving explanations one may need to refrain

from providing Potential serendipity apps. These apps are perceived to be less relevant and useful and may thereby be confusing for the app designer. All in all, these findings do not give any positive or negative indications towards the actual creativity or potential for creativity. So, even though the ideas coming from the Far-field analogy quadrant were not perceived to most original, the actual originality of ideas coming from the assessment of that quadrant may be highest. Hence, presenting apps from the Far-field analogy and Potential serendipity quadrants may still lead to more creative ideas compared to presenting apps from the Near-field analogy and Surface similarity quadrants.

Even though many results and differences were not strong, there is one interesting finding in favour of our theory. Namely, for almost all dependent variables (except for the perceived tendency to transfer example elements), the results for the Near-field analogy and Potential serendipity quadrants were most extreme. These quadrants can be seen as the extremes in the matrix regarding the goal similarity and keyword dissimilarity. This indicates that our mechanism for selecting apps, based on the goal similarity and keyword dissimilarity, has an influence on the perception of creativity and design fixation. This mechanism could thus potentially also have an influence on the actual creativity and design fixation.

8.2.4 Threats to Validity

The findings and discussion above need to be assessed in the light of various threats to validity. The main threat to validity is the limited number of responses. Further threats to validity are discussed using the description of Wohlin et al. (2012).

- *Conclusion validity.* Since only few subjects participated, no statistical tests were executed as a low statistical power was expected. Hence, no statistical significance of the results could be obtained. The reliability of the measures may also influence the conclusion validity. For instance, to the best of our knowledge, no questionnaire existed for specific experiment. Hence, the questions needed to be specifically formulated for this research. This threat was addressed by taking into account guidelines for questionnaire design (e.g., Krosnick and Presser, 2010). Also, the pretest indicated that the questionnaire was comprehensible and usable.
- *Internal validity.* Since subjects needed execute the experiment from a distance, not every aspect of the experiment could be controlled for. It could be the case that different subjects assess the app pages differently. However, all apps were presented fairly equally, which could reduce this effect. Moreover, we do not expect that the way in which participants assess the app pages heavily influences the experiment results. In the goal keyword workshop (Section 7.3.1), participants seemed to assess the app pages fairly similar. We believe that this observation also applies to this context. It could not be controlled that subjects could perform other tasks in between. However, a quick visual inspection showed that most subjects performed the experiment in a short amount of time, indicating that they finished the experiment in one sitting. However, participants did spend substantially less time on the task than was prescribed and expected. This threat can only be accepted due to the nature of the experiment. Furthermore, participants may not have actually generated ideas when assessing the apps, unlike they were asked to. However, we believe that most questions were still answerable despite that. The influence of the sampling of subjects and division of subjects into groups was reduced by applying randomisation. Finally, it could be the case that one subject may attempt to fill in the questionnaire

more than once. This was prevented as much as possible by letting the tool automatically prevent multiple answers by the same participant.

- *Construct validity.* The nature of the subjects' task was hypothetical, since they needed to reason about the task instead of actually executing the task. These threats needed to be accepted due to constraints put on this study. Furthermore, the variables might not measure the underlying constructs. For instance, approximations of certain concepts were used. We argue that this risk is mitigated as the variable selection and hypotheses formulation were rooted in literature. Furthermore, in the provided definitions, usefulness and appropriateness are about the creative product and not about the means to foster creativity. Assessing the appropriateness and usefulness of the means is still suitable. First, the aim of the study is to understand the potential for creativity and not about the actual creativity, making the assessment of the means for fostering creativity fitting. Second, we argue that something must be appropriate or useful for it to foster creativity effectively. Confounding variables could be influencing the outcomes of the results. It was taken into account that familiarity or expertise could influence the outcomes. Furthermore, the selection of apps could influence the results in an unwanted way. For instance, some apps may be inherently more creative or better at fostering creativity. This was accounted for by selecting two sets of apps. All subjects were assigned all four treatments, leading to treatment interaction. To counteract this effect, the order of the treatments between the subjects was randomised.
- *External validity.* The choice for students as subjects can be seen as threat to the external validity. However, students were told that they could only participate if they had some experience in app design. Thereby, this threat was mitigated. Also, it could not be fully controlled whether the subjects recruited on Reddit were actually app designers. However, this threat was limited by distributing the questionnaire only in relevant communities. The setting in which the experiment took place may not fully replicate the real-world setting. However, it is assumed that app designers nowadays mainly work on a computer behind a desk, making the experiment resemble the real-life situation. Lastly, due to the constraints on this study, only perceived and expected measures could be measured, not actual values. Future research may be needed to supplement the initial findings of this study.

Chapter 9

Discussion

As already many apps exist to choose from, app designers need to find a way to differentiate their work from that of others. Since creativity may be a resource for this, it may be worth fostering it. In this research, we set off to find out how the creativity of app designers can be fostered. We proposed a conceptual design for a tool that aims to achieve this objective. The design incorporates two important concepts from different fields, namely analogical reasoning and app review analysis. The interdisciplinary approach intended to tailor general creativity research to the field in question to effectively aid the app designer. In essence, the proposed tool shall present analogous apps accompanied with visualisations of app features that are discussed in app reviews. Thereby, this research proposed a way to describe apps in analogical terms and a method and associated theory to select analogous apps. The approach behind the tool combines both human and machine processing to foster the creativity of app designers. Initial, tentative results give a first proof of concept and first indications towards the feasibility of the creativity support tool.

9.1 Conclusions

Before addressing the implications and limitations of this research, first the research questions that were formulated in Chapter 1 shall be answered. The main question of this research can only be answered after answers have been given to the sub-research questions. We start by answering those first as those set out the scope of the entire research.

9.1.1 SQ1. Current State of App Design Practice

The research started by trying to understand and identify what the app design process looks like. Thereby the aim was to answer to the first sub-research question: *What is the current state of practice in app design?* This question was addressed at multiple levels (i.e., field level vs. process level). In general, there are many app developers. Most of them tend to only release one app, while some of them release many (Wang et al., 2019). One reason why this research specifically focused on app design and not software development in general is that apps are different from traditional software (Section 3.6). For instance, apps are smaller and provide fewer functionalities than traditional software (Minelli & Lanza, 2013). This suggests that different requirements may be needed and that a different design approach is required compared to traditional software development processes. Even though apps are different from traditional software, the phases of the development process are highly similar, if not the same (Inukollu et al., 2014; Nagappan & Shihab, 2016; Wang et al., 2019). The interviews indicated that the steps taken during the app design process vary per project and per company. Nevertheless, steps such

as collecting project information, sketching, prototyping, and testing seem to be recurring. The interviews also indicated that, nowadays, many app designers work in an Agile manner. During the process of designing and creating apps, many different (digital) tools are already in place for various purposes (e.g., sketching, modelling, documentation). Furthermore, oftentimes a variety of stakeholders are involved in app design processes. The most important ones seem to be the client and the end user.

9.1.2 SQ2. Current Role of Creativity in the App Design Process

The second sub-research question focused on understanding the role of creativity in the app design process. The research question was as follows: *What role does creativity currently play in the app design process?* It seems that app designers value creativity as important, even though it often does not get explicit attention. The role of creativity in the app design process differs per person, project, and company. This does not come as a surprise as, for instance, the app design process was also noted to be context dependent. Also, the importance of creativity is not assessed equivalently by different companies. Ideas emerge is various ways during the process of designing apps. It was noticed that ideas oftentimes come up by interacting and empathising with the end user. In the end, these are an important source for requirements as the app is created for them. Besides the end user, also other sources form an inspiration for app designers. The most frequently mentioned one is the work of others. App designers look at the works of others not per se to copy ideas, but to learn, to identify common practices, and to see what already exists on the market. App designers seems to not solely look at the works of direct competitors, but also to works in more distant domains. Apart from that, app designers face various challenges regarding creativity during the design process of which many are in line with the factors that influence creativity in work environments in general (Amabile et al., 1996). One of the challenges is getting stuck in the creative process and in generating new ideas. All in all, the four-stages model of Wallas (1926) appears also to be applicable to the app design domain. App designers first gather information (i.e., preparation), which is then processed consciously and unconsciously (i.e., incubation). From this, app ideas emerge (i.e., illumination) which are finally tested and evaluated with, for instance, end users (i.e., verification). So, while this research specifically focuses on the app design field, more general mechanisms also seem to be in place.

9.1.3 SQ3 & SQ4. Fostering Creativity by Using a Tool and App Markets

One of the main objectives of this research was to find out how app markets could be used for fostering the creativity of app designers. Hence, the third research question was formulated broadly as: *How can app markets be used for fostering creativity of app designers?* The use of the app markets is incorporated into the conceptual design of the tool. Therefore, the fourth sub-research questions can best be answered simultaneously with the third. This last sub-research questions was formulated as *How can creativity of app designers be fostered through the use of a tool?* The entire research was centred around the design of the creativity support tool. As described at the beginning of this chapter, two important concepts of different fields were combined in the design of a tool to foster the creativity of app designers in a targeted way. On the one hand, analogical reasoning or design-by-analogy were applied to help the app designers in finding a creative solution to their design problems. App designers already mentioned to look at the works of others to become inspired. A tool could make this process more directed

and structured, by presenting well-functioning apps as example solutions to a design problem at hand. This can be seen as providing knowledge, which is way of fostering creativity (Amabile, 1983; Boden, 2004, 2009). However, it is important that appropriate apps are selected and presented given a specific design task, for the creativity to be fostered and design fixation to be prevented (e.g., Christensen and Schunn, 2009; Dahl and Moreau, 2002; Jansson and Smith; 1991). The concept of analogical reasoning and analogical distance were found to be useful for this. In order to apply these concepts to the app domain, apps need to be described in both a structural and superficial way (Gentner, 1983). This leads to the second use of app markets in the process of fostering creativity. Pages of specific apps were found to be a useful source to describe apps structurally and superficially. Human annotators are able to tag goals (i.e., app structure) and list keywords (i.e., app surface), based on their assessment of app pages. These goals and keywords allow for classifying apps in a matrix and thereby for determining which apps are appropriate. To help the app designer in assessing the analogous solutions further, app review analysis could be applied. From these app reviews, a wide variety of topics can be extracted, such as feature requests, bug reports, and opinions about features (Pagano & Maalej, 2013). This research has given a first proof of concept that the features and associated opinions can be automatically extracted and be presented in a dense visualisation (i.e., word clouds). The features can be seen as low-level solutions to analogous design problems. Since the visualisation forms an additional source of knowledge, it is expected that these further foster the creativity (Boden, 2009). In short, this research indicates that the creativity of app designers can be fostered through a tool which presents analogous design problems in the app design field. Additional information about the apps in the form of word clouds are expected to further foster the app designers' creativity. App pages in app markets can be used to describe the structure and surface of apps and can be used to provide knowledge to app designers.

9.1.4 Answering the Main Research Question

Accumulating the previous answers allows for answering the main research question. This question was formulated as follows: *How can creativity be fostered in the process of designing mobile applications?*. In general, creativity can be fostered in many ways, with many techniques, and with many different tools. This research proposed the design of a tool which is specifically targeted at the app design field. We argue that this tool will foster the creativity of app designers as it provides knowledge and different perspectives to app designers. The tool allows app designers to evaluate creative ideas, which will further foster the creativity (Boden, 2009). Furthermore, by applying the concept of analogical distance we argue that design fixation is prevented or at least limited. This is in turn seen as beneficial for the app designers' creativity. To be short, we argue that the creativity of app designers can be fostered by applying a combination of analogical reasoning and app review analysis.

9.2 Implications

This research has taken the opportunity to aid app designers in their strive to differentiate themselves from their numerous competitors. This research has proposed a conceptual design for a creativity support tool specifically targeted at app designers. To the best of our knowledge, no research had been done so far that was actively focused on fostering the creativity of app designers and no CSTs for app designers have been proposed so far. Many CSTs have already

been proposed in the past, of which some also make use of analogical reasoning (see e.g., Zachos and Maiden, 2008) and design-by-analogy (e.g., Idea-Inspire of Chakrabarti et al., 2005 and DANE of Vattam et al., 2011). The former was also proposed in the context of RE. However, none of these tools was specifically aimed at the app design field. Platzer and Petrovic (2011) proposed a tool for app designers in a preliminary work. However, their work had an unspecific focus on creativity and was not substantiated in creativity literature. We therefore do not consider their work as a CST. Their work shows some similarities with ours, such as using app review analysis, presenting example apps, and the use of motives (cf. goals). However, our approach suggests appropriate apps based on analogies, while theirs presents apps based on frequency of motives in the app reviews. Furthermore, their approach uses general motives, while our goals are tailored to the app domain, making them more appropriate. Finally, the app review analysis had different purposes in both researches. In theirs, it was used for assigning motives to apps, while in ours the opinion of users regarding features were extracted to foster creativity. All in all, this work is significantly different from ours. To sum up, we believe that the approach taken to design our tool and the combination of concepts underlying this design are unique compared to other works.

This research contributes to the body of knowledge of multiple fields as it combines theories and concepts of various fields (e.g., creativity research and RE research). Besides the contribution of a novel design for a tool, this research also proposed a method and related theory for selecting appropriate example apps. Furthermore, this research has shown that there is a limited amount of end-user goals underlying apps. The goal and keywords approach to capture apps is simple and usable even if some apps are updated frequently. We argue that both keywords and high-level goals are stable characteristics of apps that are not affected by small updates in the design of the app. With the approach to analogical reasoning and the app review analysis, this research has given first indications that a mixture of human and machine processing is suitable in the process of fostering creativity of app designers.

Already various efforts have been put in determining the analogical distance. The most similar approach to ours it that of K. Fu et al. (2013), as they also made use of textual similarity. In line with their research, we address the analogical distance as a continuum. However, our approach is, to the best of our knowledge, unique as it exploits specific characteristics of the app domain. Furthermore, it decouples the surface elements from the structure. We argue that this allows for finding true analogies. The decoupling of the surface and structure is not unique as this is rooted in the work of Gentner (1983). Therefore, positioning domains based on surface similarity and structural similarity is also not new. For instance, Gentner et al. (1993) already researched the influence of surface similarity and structural similarity on retrievability and subjective soundness of the examples. They also made use of a matrix to categorise stories based on surface and structural similarity. Reed (1987) also classified problems in a matrix based on surface and structural similarity. However, our approach of classifying and delineating the surface and structure is to our knowledge new. We believe that this approach may also be applicable for other domains (e.g., software development domain), since user goals and keywords are generalisable concepts.

The app review analysis approach of this research builds further on the works of others (e.g., Gu and Kim, 2015; Guzman and Maalej, 2014). The performance is not yet comparable or satisfactory, but it shows that the approach is feasible and usable. Furthermore, comparing the result is difficult, if not impossible, for the aim in this study was different. Our approach is different from others as it uses both the features and opinions in a new context (i.e., fostering

creativity) and in a new way (i.e., word clouds). Furthermore, different steps were taken in the creation of our approach.

Finally, one may object by saying that already many CSTs exist. However, we believe that this tool will foster creativity more effectively, since it is specifically created for a specific field (Shneiderman, 2002). Also, specific domain knowledge is provided to the app designer which is expected to be beneficial for the creativity (Amabile, 1983; Boden, 2004). Finally, one might note that providing examples may have negative implications on, for instance, the design fixation and creativity of app designers as described in Chapter 3. However, we expect that by presenting surface dissimilar apps this issue is limited, if not prevented. Next to that, app designers already noted to look at the works of others. This research aims at making this process more structured and at preventing design fixation as much as possible. It is suggested, by for instance Christensen and Schunn (2009, p. 59), to present both near and far examples in the context of problem solving. However, we argue to only present far-field examples in this context, since all examples are somewhat near as they are still within the app field. This last argument thus means that the examples cannot be "too far" as K. Fu et al. (2013) warned for.

9.3 Limitations

The findings of this research need to be assessed in the light of several limitations, apart from the ones already mentioned in Chapter 8. The first and foremost limitation of this research is the preliminary validation. It would have been wishful to validate the creativity of actual design ideas of designers who had assessed the different example apps. However, this was not feasible due to the restrictions in time and resources. Furthermore, the initial analogical reasoning validation received only few responses despite our efforts. However, the initial validations has already given some positive evidence regarding the design of the tool and our theory. However, we cannot yet give conclusions with respect to the effect of our design on creativity and design fixation.

The performance validation of the app review analysis was done on a small scale, which limits the generalisability. We do not see this as a substantial limitation, since the validation served towards a first proof-of-concept. The initial validation and the resulting word clouds have given an initial evidence of the feasibility of the approach. Furthermore, despite the low precision, the word clouds may still be satisfactory as only the names of the most frequently occurring (grouped) features are presented. Thereby, further noise can be filtered out.

In this research, only a limited number of apps was analysed for the creation of the high-level goal set. Despite that, we still believe that most apps are captured by the set of goals, since they were created on a diverse set of apps and since they are high-level. Furthermore, the manual app review analyses were done on a small set of apps. However, this is not seen as a threat, since many findings were in line with literature. Besides that, this research only focused on the Google Play Store, which may limit the generalisability of some results. However, as Pagano and Maalej (2013) also suggested, most results are expected to be generalisable to other app markets.

Lastly, only five interviews were conducted to understand the state of the app design practice and the role of creativity in the app design process. Not all findings may be generalisable for that reason. Nevertheless, many findings were noted by multiple interviewees, sometimes even without explicitly asking them. This indicates that those findings are probably generally applicable.

9.4 Future Work

As with many researches, this research also leaves a variety of opportunities for future research. First of all, a prototype and eventually a working tool should be developed for the proposed conceptual design. This design may need to be done careful as, for instance, the representation of examples may influence the creativity and design fixation of app designers. Furthermore, as explained, it may be difficult for people to form or identify analogies by themselves (Gentner et al., 1993; Gick & Holyoak, 1980). Analogies that are readily presented are easier for people to use (Markman et al., 2009). So, to facilitate the analogical reasoning of app designers when using the tool, hints or instructions should be given towards the use of analogies (Gick & Holyoak, 1980; Christensen & Schunn, 2009). In line with that, explanations for the suggested apps may further improve the systems. The tool can be seen as a sort of recommender system, as it recommends certain apps to assess. Explanability of recommendations is an important aspect for those systems (Tintarev & Masthoff, 2007).

The design currently only allows to make analogies within one domain, namely the app domain. So, the analogies are not very far, meaning that designers are not enabled to get inspiration from other domains. Potentially, examples from other domains could also be incorporated into the tool. Again, the use of goals and keywords would allow for other design problems to be captured in the same way, making the use of other domains feasible. In the Implications section, it was mentioned that the examples cannot be "too far". It would be valuable to do more research on the optimal distance and to determine whether the Far-field analogy apps are not "too near". Furthermore, this research did not focus on mobile games. In future research, the scope may be expanded to also include those.

We believe that some apps may be inherently more creative than others. For instance, apps with very confined functionality or that are very familiar or common (see also Chan et al., 2011; von Hippel, 1986) could be less suitable for fostering creativity. Future research could be done to determine whether certain characteristics, apart from goal similarity and keyword dissimilarity, influence the suitability of apps for fostering creativity.

More research should also focus on further validating the proposed design and theory. We advise to focus on validating the effect on the creativity of products and on the design fixation of app designers. It is suggested that longitudinal studies in combination with a variety of validation methods are most appropriate for this (Shneiderman et al., 2006). It would be worthwhile to perform similar validations to the one conducted in this research in an offline setting, with people in a room working on the design task.

Lastly, besides further validating the automatic feature opinion extraction, the approach itself and the visualisation could be extended. The automatic extraction script could be enhanced with more advanced measures and patterns to improve both the precision and recall. The dictionaries could also be expanded further after performing further analyses. As already discussed, it would be valuable to also show the opinion words in the word clouds as they capture an important part of the extracted information. Before doing that, it would be highly valuable to validate the word clouds with respect to fostering creativity and the attitude of app designers. Finally, we advise to use the newest review instead of using the most relevant reviews. In the end, the tool should present up-to-date information to the app designer. This may not always be the case with the most relevant reviews that tend to cover a larger time frame (Section 8.1).

References

Al-Subaihin, A. A., Sarro, F., Black, S., Capra, L., Harman, M., Jia, Y., & Zhang, Y. (2016). Clustering mobile apps based on mined textual features. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.* Association for Computing Machinery.

Amabile, T. M. (1982). Social psychology of creativity: A consensual assessment technique. *Journal of Personality and Social Psychology*, *43*(5), 997–1013.

Amabile, T. M. (1983). The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology*, *45*(2), 357–376.

Amabile, T. M. (1998). How to kill creativity. *Harvard Business Review*, *76*(5), 77–87.

Amabile, T. M., Conti, R., Coon, H., Lazenby, J., & Herron, M. (1996). Assessing the work environment for creativity. *Academy of Management Journal*, *39*(5), 1154–1184.

App Annie. (2019). *App annie releases annual state of mobile 2019 report.* Retrieved from https://www.appannie.com/en/about/press/releases/app-annie-releases-annual-state-of-mobile-2019-report/

Ball, L. J., Ormerod, T. C., & Morley, N. J. (2004). Spontaneous analogising in engineering design: A comparative analysis of experts and novices. *Design Studies*, *25*(5), 495–508.

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The goal question metric paradigm. In J. Marciniak (Ed.), *Encyclopedia of software engineering* (pp. 528–532). Wiley.

Besemer, S. P. (1998). Creative Product Analysis Matrix: Testing the model structure and a comparison among products - three novel chairs. *Creativity Research Journal*, *11*(4), 333–346.

Besemer, S. P., & O'Quin, K. (1986). Analyzing creative products: Refinement and test of a judging instrument. *The Journal of Creative Behavior*, *20*(2), 115–126.

Besemer, S. P., & O'Quin, K. (1999). Confirming the three-factor Creative Product Analysis Matrix model in an American sample. *Creativity Research Journal*, *12*(4), 287–296.

Besemer, S. P., & O'Quin, K. (2011). Creativity products. In M. A. Runco & S. R. Pritzker (Eds.), *Encyclopedia of creativity* (2nd ed., pp. 273–281). Elsevier Academic Press.

Besemer, S. P., & Treffinger, D. J. (1981). Analysis of creative products: Review and synthesis. *The Journal of Creative Behavior*, *15*(3), 158–178.

Boden, M. A. (1996). Creativity. In M. A. Boden (Ed.), *Artificial intelligence* (pp. 267–291). Academic Press.

Boden, M. A. (2004). *The creative mind: Myths and mechanisms.* Routledge.

Boden, M. A. (2009). Creativity: How does it work? In M. Krausz, D. Dutton, & K. Bardsley (Eds.), *The idea of creativity* (pp. 237–250). Brill.

Brown, T. (2008). Design thinking. *Harvard Business Review*, *86*(6), 84.

Burnay, C., Horkoff, J., & Maiden, N. (2016). Stimulating stakeholders' imagination: New creativity triggers for eliciting novel requirements. In *2016 IEEE 24th International Requirements Engineering Conference (RE)* (pp. 36–45). IEEE.

Chakrabarti, A., Sarkar, P., Leelavathamma, B., & Nataraju, B. (2005). A functional represen-

tation for aiding biomimetic and artificial inspiration of new ideas. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *19*(2), 113–132.

Chakrabarti, A., Siddharth, L., Dinakar, M., Panda, M., Palegar, N., & Keshwani, S. (2017). Idea Inspire 3.0 — a tool for analogical design. In A. Chakrabarti & D. Chakrabarti (Eds.), *Research into design for communities* (Vol. 2, pp. 475–485). Springer.

Chan, J., Fu, K., Schunn, C., Cagan, J., Wood, K., & Kotovsky, K. (2011). On the benefits and pitfalls of analogies for innovative design: Ideation performance based on analogical distance, commonness, and modality of examples. *Journal of Mechanical Design*, *133*(8), 081004.

Chen, N., Lin, J., Hoi, S. C., Xiao, X., & Zhang, B. (2014). AR-miner: Mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 767–778). Association for Computing Machinery.

Cherry, E., & Latulipe, C. (2014). Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction*, *21*(4).

Christensen, B. T., & Schunn, C. D. (2007). The relationship of analogical distance to analogical function and preinventive structure: The case of engineering design. *Memory & Cognition*, *35*(1), 29–38.

Christensen, B. T., & Schunn, C. D. (2009). "Putting blinkers on a blind man": Providing cognitive support for creative processes with environmental cues. In A. B. Markman & K. L. Wood (Eds.), *Tools for innovation* (pp. 48–74). Oxford University Press.

Crilly, N., & Cardoso, C. (2017). Where next for research on fixation, inspiration and creativity in design? *Design Studies*, *50*, 1–38.

Cropley, A. (2006). In praise of convergent thinking. *Creativity Research Journal*, *18*(3), 391–404.

Cropley, D. H. (2016). Creativity in engineering. In G. Corazza & S. Agnoli (Eds.), *Multidisciplinary contributions to the science of creative thinking* (pp. 155–173). Springer.

Cropley, D. H., & Cropley, A. (2005). Engineering creativity: A systems concept of functional creativity. In J. C. Kaufman & J. Baer (Eds.), *Creativity across domains: Faces of the muse* (pp. 169–185). Lawrence Erlbaum.

Cropley, D. H., Kaufman, J. C., & Cropley, A. J. (2011). Measuring creativity for innovation management. *Journal of Technology Management & Innovation*, *6*(3), 13–30.

Cybulski, J., Nguyen, L., Thanasankit, T., & Lichtenstein, S. (2003). Understanding problem solving in requirements engineering: Debating creativity with is practitioners. In *PACIS 2003: Proceedings of the Seventh Pacific Asia Conference on Information Systems* (pp. 465–482). University of South Australia.

Dahl, D. W., & Moreau, P. (2002). The influence and value of analogical thinking during new product ideation. *Journal of Marketing Research*, *39*(1), 47–60.

Dallman, S., Nguyen, L., Lamp, J., & Cybulski, J. (2005). Contextual factors which influence creativity in requirements engineering. In *Information systems in a rapidly changing economy: ECIS 2005, 13th European Conference on Information Systems* (pp. 1–12). University of Regensburg.

Dalpiaz, F., Franch, X., & Horkoff, J. (2016). *iStar 2.0 language guide.* arXiv e-prints.

Dalpiaz, F., & Parente, M. (2019). RE-SWOT: From user feedback to requirements via competitor analysis. In G. M. Knauss E. (Ed.), *Requirements engineering: Foundation for software quality* (pp. 55–70). Springer.

Davidson, J. L., & Jensen, C. (2013). Participatory design with older adults: an analysis of creativity in the design of mobile healthcare applications. In *Proceedings of the 9th ACM Conference on Creativity & Cognition* (pp. 114–123). Association for Computing Machinery.

Diamantopoulos, A., Sarstedt, M., Fuchs, C., Wilczynski, P., & Kaiser, S. (2012). Guidelines for choosing between multi-item and single-item scales for construct measurement: a predictive validity perspective. *Journal of the Academy of Marketing Science*, *40*(3), 434–449.

El-Sharkawy, S., & Schmid, K. (2011). A heuristic approach for supporting product innovation in requirements engineering: A controlled experiment. In D. Berry & X. Franch (Eds.), *Requirements engineering: Foundation for software quality* (pp. 78–93). Springer.

Evans Data Corporation. (2016). *Mobile Developer Population Reaches 12M Worldwide, Expected to Top 14M by 2020.* Retrieved from https://evansdata.com/press/viewRelease.php?pressID=244

Fellows, I. (2018). wordcloud: Word clouds [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=wordcloud (R package version 2.6)

Ferreira, D. J. (2013). Fostering the creative development of computer science students in programming and interaction design. *Procedia Computer Science*, *18*, 1446–1455.

Frich, J., MacDonald Vermeulen, L., Remy, C., Biskjaer, M. M., & Dalsgaard, P. (2019). Mapping the landscape of creativity support tools in HCI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1–18). Association for Computing Machinery.

Frich, J., Mose Biskjaer, M., & Dalsgaard, P. (2018). Twenty years of creativity research in human-computer interaction: Current state and future directions. In *Proceedings of the 2018 Designing Interactive Systems Conference* (p. 1235—1257). Association for Computing Machinery.

Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., & Sadeh, N. (2013). Why people hate your app: Making sense of user feedback in a mobile app store. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1276–1284). Association for Computing Machinery.

Fu, K., Chan, J., Cagan, J., Kotovsky, K., Schunn, C., & Wood, K. (2013). The meaning of "near" and "far": The impact of structuring design databases and the effect of distance of analogy on design output. *Journal of Mechanical Design*, *135*(2), 021007.

Gabriel, A., Monticolo, D., Camargo, M., & Bourgault, M. (2016). Creativity support systems: A systematic mapping study. *Thinking Skills and Creativity*, *21*, 109–122.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155–170.

Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 199–241). Cambridge University Press.

Gentner, D., Brem, S., Fergunson, R., Wolff, P., Markman, A. B., & Forbus, K. (1997). Analogy and creativity in the works of Johannes Kepler. In T. B. Ward, S. M. Smith, & J. Vaid (Eds.), *Creative thought: An investigation of conceptual structures and processes* (pp. 403–459). American Psychological Association.

Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. *American Psychologist*, *52*(1), 45–56.

Gentner, D., Rattermann, M. J., & Forbus, K. D. (1993). The roles of similarity in transfer:

Separating retrievability from inferential soundness. *Cognitive Psychology*, *25*(4), 524–575.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, *12*(3), 306–355.

Goel, A. K. (1997). Design, analogy, and creativity. *IEEE Expert*, *12*(3), 62–70.

Groen, E. C., Kopczyńska, S., Hauer, M. P., Krafft, T. D., & Doerr, J. (2017). Users — the hidden software product quality experts?: A study on how app users report quality aspects in online reviews. In *2017 IEEE 25th International Requirements Engineering Conference (RE)* (pp. 80–89). IEEE.

Gu, X., & Kim, S. (2015). "What parts of your apps are loved by users?" (t). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 760–770). IEEE.

Guzman, E., & Maalej, W. (2014). How do users like this feature? A fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)* (pp. 153–162). IEEE.

Hall, R. P. (1989). Computational approaches to analogical reasoning: A comparative analysis. *Artificial intelligence*, *39*(1), 39–120.

Harman, M., Jia, Y., & Zhang, Y. (2012). App store mining and analysis: MSR for app stores. In *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)* (pp. 108–111). IEEE.

Hehn, J., Mendez, D., Uebernickel, F., Brenner, W., & Broy, M. (2019). On integrating design thinking for a human-centered requirements engineering. *IEEE Software*, *37*(2), 25–34.

Herrmann, A., Mich, L., & Berry, D. M. (2018). Creativity techniques for requirements elicitation: Comparing four-step EPMcreate-based processes. In *2018 IEEE 7th International Workshop on Empirical Requirements Engineering (EmpiRE)* (pp. 1–7). IEEE.

Holyoak, K. J., & Thagard, P. (1997). The analogical mind. *American Psychologist*, *52*(1), 35–44.

Horn, D., & Salvendy, G. (2006). Consumer-based assessment of product creativity: A review and reappraisal. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *16*(2), 155–175.

Horn, D., & Salvendy, G. (2009). Measuring consumer perception of product creativity: Impact on satisfaction and purchasability. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *19*(3), 223–240.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 168–177). Association for Computing Machinery.

Huang, J., Etzioni, O., Zettlemoyer, L., Clark, K., & Lee, C. (2012). RevMiner: An extractive interface for navigating reviews on a smartphone. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (pp. 3–12). Association for Computing Machinery.

Inukollu, V. N., Keshamoni, D. D., Kang, T., & Inukollu, M. (2014). Factors influencing quality of mobile apps: Role of mobile app development life cycle. *International Journal of Software Engineering & Applications (IJSEA)*, *5*(5), 15–34.

Isermann, H. (1982). Linear lexicographic optimization. *OR Spektrum*, *4*(4), 223–228.

Jansson, D. G., & Smith, S. M. (1991). Design fixation. *Design Studies*, *12*(1), 3–11.

Jesson, J., Matheson, L., & Lacey, F. M. (2011). *Doing your literature review: Traditional and systematic techniques*. Sage.

Jha, N., & Mahmoud, A. (2019). Mining non-functional requirements from app store reviews. *Empirical Software Engineering*, *24*, 3659–3695.

Jiang, H., Ma, H., Ren, Z., Zhang, J., & Li, X. (2014). What makes a good app description? In *Proceedings of the 6th Asia-Pacific Symposium on Internetware on Internetware* (pp. 45–53). Association for Computing Machinery.

Jobe, W. (2013). Native apps vs. mobile web apps. *International Journal of Interactive Mobile Technologies*, *7*(4), 27–32.

Johann, T., Stanik, C., Alizadeh B., A. M., & Maalej, W. (2017). SAFE: A simple approach for feature extraction from app descriptions and app reviews. In *2017 IEEE 25th International Requirements Engineering Conference (RE)* (pp. 21–30). IEEE.

Karlsen, I. K., Maiden, N., & Kerne, A. (2009). Inventing requirements with creativity support tools. In *Requirements Engineering: Foundation for Software Quality* (pp. 162–174). Springer.

Kipper Schuler, K. (2005). *Verbnet: A broad-coverage, comprehensive verb lexicon.* ProQuest. (Doctoral dissertation, University of Pennsylvania.)

Krosnick, J. A., & Presser, S. (2010). Question and questionnaire design. In P. V. Marsden & J. D. Wright (Eds.), *Handbook of survey research* (2nd ed., pp. 263–314). Emerald Group Publishing Limited.

Lemos, J., Alves, C., Duboc, L., & Rodrigues, G. N. (2012). A systematic mapping study on creativity in requirements engineering. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 1083–1088). Association for Computing Machinery.

Linsey, J. S., Tseng, I., Fu, K., Cagan, J., Wood, K. L., & Schunn, C. (2010). A study of design fixation, its mitigation and perception in engineering design faculty. *Journal of Mechanical Design*, *132*(4), 041003.

Linsey, J. S., Wood, K. L., & Markman, A. B. (2008). Modality and representation in analogy. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *22*(2), 85–100.

Lombriser, P., Dalpiaz, F., Lucassen, G., & Brinkkemper, S. (2016). Gamified requirements engineering: Model and experimentation. In *Requirements Engineering: Foundation for Software Quality* (pp. 171–187). Springer.

Maalej, W., & Nabil, H. (2015). Bug report, feature request, or simply praise? On automatically classifying app reviews. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)* (pp. 116–125). IEEE.

Mahaux, M., Nguyen, L., Gotel, O., Mich, L., Mavin, A., & Schmid, K. (2013). Collaborative creativity in requirements engineering: Analysis and practical advice. In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)* (pp. 1–10). IEEE.

Mahaux, M., Nguyen, L., Mich, L., & Mavin, A. (2014). A framework for understanding collaborative creativity in requirements engineering: Empirical validation. In *2014 IEEE 4th International Workshop on Empirical Requirements Engineering (EmpiRE)* (pp. 48–55). IEEE.

Maher, M. L., & Fisher, D. H. (2012). Using AI to evaluate creative designs. In *DS 73-1 Proceedings of the 2nd International Conference on Design Creativity* (Vol. 1, pp. 45–54). The Design Society.

Maiden, N., Gizikis, A., & Robertson, S. (2004). Provoking creativity: Imagine what your requirements could be like. *IEEE Software*, *21*(5), 68–75.

Maiden, N., Jones, S., Karlsen, K., Neill, R., Zachos, K., & Milne, A. (2010). Requirements engineering as creative problem solving: A research agenda for idea finding. In *2010 18th IEEE International Requirements Engineering Conference* (pp. 57–66). IEEE.

Markman, A. B., & Wood, K. L. (2009). The cognitive science of innovation tools. In A. B. Markman & K. L. Wood (Eds.), *Tools for innovation* (pp. 3–22). Oxford University Press.

Markman, A. B., Wood, K. L., Linsey, J. S., Murphy, J. T., & Laux, J. P. (2009). Supporting innovation by promoting analogical reasoning. In A. B. Markman & K. L. Wood (Eds.), *Tools for innovation* (pp. 85–103). Oxford University Press.

Marsh, R. L., Landau, J. D., & Hicks, J. L. (1996). How examples may (and may not) constrain creativity. *Memory & Cognition*, *24*(5), 669–680.

Marsh, R. L., Ward, T. B., & Landau, J. D. (1999). The inadvertent use of prior knowledge in a generative cognitive task. *Memory & Cognition*, *27*(1), 94–105.

Mich, L., Anesi, C., & Berry, D. M. (2005). Applying a pragmatics-based creativity-fostering technique to requirements elicitation. *Requirements Engineering*, *10*(4), 262–275.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Minelli, R., & Lanza, M. (2013). Software analytics for mobile applications – Insights & lessons learned. In *2013 17th European Conference on Software Maintenance and Reengineering* (pp. 144–153). IEEE.

Mohanani, R., Ralph, P., & Shreeve, B. (2014). Requirements fixation. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 895–906). Association for Computing Machinery.

Mumford, M. D. (2003). Where have we been, where are we going? Taking stock in creativity research. *Creativity Research Journal*, *15*(2-3), 107–120.

Nagappan, M., & Shihab, E. (2016). Future trends in software engineering research for mobile apps. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)* (Vol. 5, pp. 21–32). IEEE.

Nakakoji, K. (2005). Seven issues for creativity support tool researchers. In B. Shneiderman, G. Fisher, M. Czerwinski, M. Resnick, & B. Myers (Eds.), *Creativity support tools* (pp. 67–70). (Workshop report)

O'Quin, K., & Besemer, S. P. (2006). Using the Creative Product Semantic Scale as a metric for results-oriented business. *Creativity and Innovation Management*, *15*(1), 34–44.

Pagano, D., & Maalej, W. (2013). User feedback in the appstore: An empirical study. In *2013 21st IEEE international requirements engineering conference (RE)* (pp. 125–134). IEEE.

Platzer, E., & Petrovic, O. (2011). Learning mobile app design from user review analysis. *International Journal of Interactive Mobile Technologies*, *5*(3), 43–50.

Powers, D. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Purcell, A. T., & Gero, J. S. (1992). The effects of examples on the results of a design activity. *Knowledge-Based Systems*, *5*(1), 82–91.

Purcell, A. T., & Gero, J. S. (1996). Design and other types of fixation. *Design Studies*, *17*(4), 363–383.

R Core Team. (2020). R: A language and environment for statistical computing [Computer software manual]. Retrieved from https://www.R-project.org/

Reed, S. K. (1987). A structure-mapping model for word problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*(1), 124–139.

Rhodes, M. (1961). An analysis of creativity. *The Phi Delta Kappan*, *42*(7), 305–310.

Rinker, T. W. (2019). sentimentr: Calculate text polarity sentiment [Computer software manual]. Retrieved from http://github.com/trinker/sentimentr (R package version 2.7.1)

Rinker, T. W. (2020). qdap: Quantitative discourse analysis package [Computer software manual]. Retrieved from http://github.com/trinker/qdap (R package version 2.4.1)

Sadler-Smith, E. (2015). Wallas' four-stage model of the creative process: More than meets the eye? *Creativity Research Journal*, *27*(4), 342–352.

Sakhnini, V., Mich, L., & Berry, D. M. (2012). The effectiveness of an optimized EPMcreate as a creativity enhancement technique for web site requirements elicitation. *Requirements Engineering*, *17*(3), 171–186.

Sarro, F. (n.d.). *The UCLAppA repository: A repository of research articles on mobile software engineering and app store analysis.* CREST Centre, UCL. Retrieved from http://www0.cs.ucl.ac.uk/staff/F.Sarro/projects/UCLappA/UCLappArepository.html

Scott, G., Leritz, L. E., & Mumford, M. D. (2004). Types of creativity training: Approaches and their effectiveness. *The Journal of Creative Behavior*, *38*(3), 149–179.

Shah, F., Sirts, K., & Pfahl, D. (2018). Simple app review classification with only lexical features. In *Proceedings of the 13th International Conference on Software Technologies (ICSOFT 2018)* (pp. 112–119). SCITEPRESS.

Shah, F., Sirts, K., & Pfahl, D. (2019). Is the SAFE approach too simple for app feature extraction? A replication study. In G. M. Knauss E. (Ed.), *Requirements engineering: Foundation for software quality* (pp. 21–36). Springer.

Shimada, H., Nakagawa, H., & Tsuchiya, T. (2019). Goal model construction based on user review classification. In P. Spoletini et al. (Eds.), *Joint proceedings of REFSQ-2019 workshops, doctoral symposium, live studies track, and poster track co-located with the 25th international conference on requirements engineering: Foundation for software quality (REFSQ 2019)* (Vol. 2376). CEUR-WS.org.

Shneiderman, B. (2002). Creativity support tools. *Communications of the ACM*, *45*(10), 116–120.

Shneiderman, B. (2007). Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM*, *50*(12), 20–32.

Shneiderman, B., Fischer, G., Czerwinski, M., Resnick, M., Myers, B., Candy, L., ... others (2006). Creativity support tools: workshop report. *International Journal of Human-Computer Interaction*, *20*(2), 61–77.

Smith, G. F. (1998). Idea-generation techniques: A formulary of active ingredients. *The Journal of Creative Behavior*, *32*(2), 107–134.

Smith, S. M., Ward, T. B., & Schumacher, J. S. (1993). Constraining effects of examples in a creative generation task. *Memory & Cognition*, *21*(6), 837–845.

Solis, C., & Ali, N. (2010). Distributed requirements elicitation using a spatial hypertext wiki. In *2010 5th IEEE International Conference on Global Software Engineering* (pp. 237–246). IEEE.

Thagard, P., Holyoak, K. J., Nelson, G., & Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial intelligence*, *46*(3), 259–310.

Tintarev, N., & Masthoff, J. (2007). Effective explanations of recommendations: User-centered design. In *Proceedings of the 2007 ACM Conference on Recommender Systems* (pp. 153–

156). Association for Computing Machinery.

Tschimmel, K. (2012). Design thinking as an effective toolkit for innovation. In *Proceedings of the XXIII ISPIM Conference: Action for Innovation: Innovating from Experience* (pp. 1–20). The International Society for Professional Innovation Management (ISPIM).

Valentim, N. M. C., Silva, W., & Conte, T. (2017). The students' perspectives on applying design thinking for the design of mobile applications. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)* (pp. 77–86). IEEE.

Vattam, S., Wiltgen, B., Helms, M., Goel, A. K., & Yen, J. (2011). DANE: Fostering creativity in and through biologically inspired design. In T. Taura & Y. Nagai (Eds.), *Design creativity 2010* (pp. 115–122). Springer.

Verhaegen, P.-A., D'hondt, J., Vandevenne, D., Dewulf, S., & Duflou, J. R. (2011). Identifying candidates for design-by-analogy. *Computers in Industry*, *62*(4), 446–459.

Vetterli, C., Brenner, W., Uebernickel, F., & Petrie, C. (2013). From palaces to yurts: Why requirements engineering needs design thinking. *IEEE Internet Computing*, *17*(2), 91–94.

Vieira, E. R., Alves, C., & Duboc, L. (2012). Creativity patterns guide: Support for the application of creativity techniques in requirements engineering. In M. Winckler, P. Forbrig, & R. Bernhaupt (Eds.), *Human-centered software engineering* (pp. 283–290). Springer.

Voigt, M., Niehaves, B., & Becker, J. (2012). Towards a unified design theory for creativity support systems. In K. Peffers, M. Rothenberger, & B. Kuechler (Eds.), *Design science research in information systems. advances in theory and practice.* (pp. 152–173). Springer.

von Hippel, E. (1986). Lead users: A source of novel product concepts. *Management Science*, *32*(7), 791–805.

Vu, P. M., Nguyen, T. T., Pham, H. V., & Nguyen, T. T. (2015). Mining user opinions in mobile app reviews: A keyword-based approach. In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 749–759). IEEE.

Vu, P. M., Pham, H. V., Nguyen, T. T., & Nguyen, T. T. (2016). Phrase-based extraction of user opinions in mobile app reviews. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering* (pp. 726–731). Association for Computing Machinery.

Wallas, G. (1926). *The art of thought.* Jonathan Cape.

Wang, H., Wang, X., & Guo, Y. (2019). Characterizing the global mobile app developers: A large-scale empirical study. In *Proceedings of the 6th international conference on mobile software engineering and systems* (pp. 150–161). IEEE.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering.* Springer-Verlag.

Wijffels, J. (2019). udpipe: Tokenization, parts of speech tagging, lemmatization and dependency parsing with the 'UDPipe' 'NLP' toolkit [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=udpipe (R package version 0.8.3)

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering.* Springer-Verlag.

Zachos, K., & Maiden, N. (2008). Inventing requirements from software: An empirical investigation with web services. In *2008 16th IEEE International Requirements Engineering Conference* (pp. 145–154). IEEE.

Zachos, K., Maiden, N., Pitts, K., Jones, S., Turner, I., Rose, M., ... MacManus, J. (2013). A software app to support creativity in dementia care. In *Proceedings of the 9th ACM Con-*

ference on Creativity & Cognition (pp. 124–133). Association for Computing Machinery.

Zeng, L., Salvendy, G., & Zhang, M. (2009). Factor structure of web site creativity. *Computers in Human Behavior*, *25*(2), 568–577.

Appendix A

Interview Protocol

Fostering creativity in app design

Interview protocol

Introduction

"You have been selected for this interview, because you have been identified as someone who has knowledge about the process of designing apps within a company. For my research, I investigate creativity in this process. The main goal of the research is to identify how the creativity of app designers can be fostered. With the term app design, I refer to the creation and design of apps in a broad sense. This interview is for exploratory purposes, meaning that I am interested in your view on creativity in your work."

"We have planned this interview to last no longer than one hour. During this time, I have several open questions that I would like to ask. If you need any clarifications; you may ask me at any time. The outcomes of the interview will be shared with you, if you are interested in this."

"Do you have any further questions you would like to ask?"

"To facilitate our notetaking, I would like to audio-record our conversation. For your information, only the interviewer and her supervisors on the project will be privy to the tapes. Please sign the consent form. Essentially, this document states that: (1) all information will be held as confidential as possible, and (2) your participation is entirely voluntary and you may stop at any time. Thank you for agreeing to participate."

Let participant sign informed consent.

Questions

The numbered questions are the main questions to be answered. The alphabetically indicated questions are probing questions. These questions may or may not be explicitly asked depending on the answers given by the interviewee and on the flow of the conversation. If needed, additional questions could be asked during the interview for clarification or other purposes.

Introductory questions

1. Could you tell something about your job?
    a. What is your job?
    b. What is your (previous) experience with app design?

Company details

1. Could you tell something about the company in two to three sentences?
    a. What types of products does the company create/deliver?
    b. Is there one or are there multiple teams/departments creating apps or other software?
2. Can you describe the team/department that is working on app design (and that you are working in)?
    a. How big is the team/department?
    b. What type of products does the team/department create/deliver?

Creativity

Explain what creativity is to the interviewee. Explain that creativity is not only about aesthetics, but also about more functional ideas (e.g.., problem solving). For instance, one may come up with a functionality of an app that is novel and valuable. "Ideas here may for instance relate to functionalities, qualities, and services delivered by the app".

1. How do new ideas appear to emerge?
    a. Do new ideas arise from individuals or do they arise in collaboration?
2. Where do you get your inspiration from?
    a. Do you for instance look at the works of others (i.e., both competitors and non-competing companies) to get inspired?
    b. Do you look at the reviews in the app store to generate ideas or to improve the app design?
3. Is creativity an implicit part or explicit part of the design process?
    a. Do you make use of some sort of creativity techniques (e.g., brainstorming, mind mapping, etc.)?
4. Could you give an example of a creative idea in the app design process?
    a. Could you explain why you think this is a good example?
5. How important is creativity in the company you are working for?
    a. How new or innovative should the design ideas be?
    b. What drives the generation of new ideas (e.g., culture and strategy)?
6. Do you experience any difficulties in coming up with creative or new ideas?
    a. What kind of difficulties do you encounter?
    b. Do you have any systematic ways of coping with those difficulties?

App design process

1. Could you describe what the app design process looks like in your company (in case of a small company) or team?
    a. Are there specific phases, steps, or milestones in this process?
    b. What are the novel ideas that are created in this process? What do they look like and when do they arise?
    c. Do you follow any methods or frameworks with predefined steps?
    d. What kind of tools are currently used in the app design process (e.g., design prototyping tools)?
2. Who are the stakeholders? So, who is involved in the app design process?
    a. Who is involved in the elicitation of the requirements?
    b. Could you tell who is making the decisions regarding the functionalities, qualities, and services of the app?
    c. Are users and customers involved in the process? If yes, why and how? If no, why?
3. How is the app design process influenced by the companies' strategy or company culture?

End interview

Thank interviewee for participating. Ask if there are any questions. Share what is going to be done with the information from the interview. Mention that if interviewee is interested, the results could be shared.

Appendix B

Goal Analysis App Selection

This appendix describes the criteria and procedure for selecting apps for the creation of a generic set of goals. The following criteria were adhered to:

- The rating of the apps needed to be 4.5 or higher, similar to in Section 6.1. This criterion was also maintained in this manual goal analysis to ensure that the set of apps would represent apps that could be presented in the tool.
- Only apps that were downloadable for free were selected, for the reason mentioned above.
- Games were not included in the set, since those are outside the scope of the research.
- An app cannot be included if a highly similar app was already selected, since an as diverse set of apps as possible needed to be selected
- The app description should be available in English to understand the app and to extract goals.

Note, the decision for inclusion was made subjectively based on the diversity of the set and the quality of the to-be selected app. Also, the Google Play Store seems to adapt the shown apps to the user, causing that different users could be presented different apps in the same category.

The following steps were taken to create a diverse set of apps:

1. Select at least one app from each app category in the Google Play Store. The apps were taken from the *Top apps* overview. We reasoned that apps in the different categories expose different characteristics and may serve different goals. The following categories were not consulted for apps, since the apps in those categories were assigned by the Play Store to other main categories: Wear OS by Google, Daydream, and Augmented reality.
2. Select additional apps from the categories within the app market. Categories were either accessed via the menu or via the search bar. Some categories seemed to be more homogeneous in terms of exposed functionalities (e.g., *Dating* and *Communication*) than other categories. In that case, less additional apps were selected from those categories. Thus, the more diverse a category, the higher the number of apps selected from that category.
3. Add the app to the set if it complies to the specified inclusion criteria.

Appendix C

Goal & Keyword Workshop

This appendix provides the documents used before and during the workshop. Also, the apps that were assigned to the workshop participants can be found in this appendix.

Table C.1: Sets of apps assigned to the workshop participants.

Group 1	**Group 2**
Drivvo – Car management, Fuel log, Find Cheap Gas	Socratic by Google
ReadEra - book reader pdf, epub, word	Voice changer with effects
Cookpad - Create your own Recipes	Lloyds Bank Mobile Banking: by your side
Home Workout - No Equipment	Woebot: Your Self-Care Expert
Lullaby for Babies	Reddit
Waze - GPS, Maps, Traffic Alerts & Live Navigation	BeautyPlus - Easy Photo Editor & Selfie Camera
ColorNote Notepad Notes	Forest: Stay focused
Instagram	Stocard - Rewards Cards Wallet
YI Home	Polar steps
AR Ruler App – Tape Measure & Camera To Plan	New Emojis Stickers 3D Animated WAStickerApps

The following two pages show the instruction file that was provided to the workshop participants. The template used in the example was also given to the participants for providing the results of their exercise.

Mobile application analysis

First of all, thank you for participating in this workshop! In this exercise, you are asked to describe mobile applications (hereafter "apps") in terms of goals and keywords. This exercise needs to be performed on **10 apps**, which you can find in the other file. During the exercise you may access the page of the app in the Google Play Store. The link to the page of the app is also provided in the other file. You may consult the app title, pictures, videos (if available), and description to get more information about the app.

Goal analysis

The first column of the provided tables lists general goals of end users regarding apps. These goals are high-level reasons for downloading specific apps. One app can be linked to one or multiple goals. The goals are specified at a high level to make it possible that they can be linked to many types of apps.

Some goals may be fulfilled by many apps (e.g., share), but oftentimes they are not the main reason for downloading the app. Please make sure that you only **select the main goals**. So, only tick the boxes that you think that apply best to this app. Do NOT tick the boxes of goals that are (too) far-fetched or irrelevant according to you. An additional line of information is given for some goals to provide some direction of thinking. However, this is not done for each goal to give you freedom of interpretation and tagging.

The task

- Tick the boxes of goals that apply to the app. If the goal applies to the app, write down an 'x' in the column called "Applicable?".
- You may select multiple goals, but select at least 1 goal.
- Only select the main goals that describe the purpose of the app best.
- If you are in doubt between two goals, you can tag them both.
- Note that there are no wrong answers in this exercise; goals are a personal matter. However, still try to tag the best goals that make sense to you.
- *Tip: try to think about why you would download the app and what purposes the app would fulfil*.

Keyword tagging

Each app can be described with a set of keywords that distinguish it from other (types of) apps. These words may describe the theme of the app or the elements that are typical for this app. When thinking of an app, these words will come to mind first. *Example: A recipe app may be described with the following keywords: recipe, food, ingredient list, cooking, instructions, etc. A photo editing app may be described with keywords such as: picture editing, filters, retouch, etc.*

The task

- List the keywords for each app.
- Provide at least 5 keywords and do not provide more than 20 keywords.
- A keyword will mainly consist of one word (e.g., motivation), but may consist of multiple words (e.g., goal achievement).
- Keywords will mainly be described in terms of nouns. If needed, adjectives and adverbs are also allowed to complement the noun. If a verb is needed, please write it down as a gerund (e.g., picture editing, book writing). Write down keywords in singular.
- Please try to write down keywords carefully to prevent typos.
- *Tip: the app page at the Google Play Store could be helpful in finding keywords.*

EXAMPLE: *Duolingo: Learn Languages Free*

This example will give you an idea on how to perform the task. The list of keywords provided in this example is to give you an idea. It is not necessarily a complete set of keywords. So, it could be the case that you would provide additional keywords.

Link: https://play.google.com/store/apps/details?id=com.duolingo

Goal	**Applicable?**
Log, monitor & track *E.g., events, activities, behaviour, trends, experiences, progress, etc.*	x
Be informed & obtain knowledge	
Learn & practice skills	x
Stay mentally and physically healthy	
Get guidance & be assisted	
Arrange matters & organise	
Replace physical tools and objects with an app	
Control IoT or physical devices	
Perform any action on physical or digital files *E.g., open, explore, remove, edit, scan, etc.*	
Make & complete transactions	
Save money	
Search & discover	
Be entertained	x
Communicate & share	
Create & design	
Personalise & customise	
Get inspiration for new ideas	
Preserve security & privacy	

Keywords
Language
Education
Speaking skills
Reading skills
Listening skills
Language skills
Vocabulary
...

Mobile application analysis - discussion

General

1. How long did it take you to finish the entire exercise? *(Please note down the estimated duration for each member).*

1.
2.
3.
4.

2. Did you only go from one app to the next or did you also move back to adjust?

3. Was the exercise description clear enough for you? If not: why? *E.g., Is it too long or too short? Did you miss any essential information?*

4. Overall, which part was more difficult for you: the goal part or the keyword part?

5a. For which app(s) was this assignment most difficult? Why?

5b. For which app(s) was this assignment easiest? Why?

6. Did you experience any differences between 1) apps you have used yourself, 2) apps you haven't used yourself but that you know or could imagine what it comprises, 3) apps about which you had little to no knowledge about beforehand.

Goal analysis

7. Was this exercise difficult or intuitive for you? Why?

8. How extensively did you use the app page in the Google Play Store for this part of the exercise?

9a. Were some goals confusing or unclear? If yes: which ones and why? How would you improve them?

9b. What did you think of the following goals? How did you interpret them? Were they confusing? (*Only the ones that were not discussed yet.)*

Be informed & obtain knowledge:

Log, monitor & track:

Get guidance & be assisted:

Arrange matters & organise:

Perform any action on physical or digital files:

Learn & practice skills:

Search & discover:

Create & design:

Replace physical tools:

Personalise & customise:

10. Did you miss any goal(s) when tagging the apps? If yes: which one and for which app?

11a. Have you used any of the two extra lines of explanation that complement those goals?

11b. Did you miss/need an additional line of explanation for any goal? If yes: for which goal?

Keyword tagging

12. Was this exercise difficult or intuitive for you? Why?

13. How extensively did you use the app page in the Google Play Store for this part of the exercise?

Final questions

14. Do you have additional tips or comments?

15. Discuss some apps individually. *(If time allows.)*

The table below shows the weighted average agreement of participants per goal. The agreement for each goal was only taken from the apps for which the goal was tagged at least once. The weighted average was taken, since the number of apps for which the goal was tagged differed per goal.

Table C.2: Average agreement between participants per goal.

Goal	**Average agreement**
Log, monitor & track	0.70
Be informed & obtain knowledge	0.50
Learn & practice skills	0.40
Stay mentally and physically healthy	0.69
Get guidance & be assisted	0.53
Arrange matters & organise	0.46
Replace physical tools and objects with an app	0.68
Control IoT or physical devices	0.63
Perform any action on physical or digital files	0.36
Make & complete transactions	0.63
Save money	0.45
Search & discover	0.50
Be entertained	0.64
Communicate & share	0.69
Create & design	0.42
Personalise & customise	0.30
Get inspiration for new ideas	0.58
Preserve security & privacy	0.75

Appendix D

App Review Analysis Validation

Feature Opinion Extraction Coding Guide

These guidelines were created and used for the creation of a golden standard for the validation of the performance of automatic feature opinion extraction. Note: these guidelines are based on knowledge obtained in the manual review analyses and on own insights, while considering the objectives of this research. These guidelines give direction on what to annotate as a feature opinion pair. Keeping in mind the overall purpose and use of the extracted feature opinion pairs can aid the annotator in deciding whether it is wishful to extract a pair or not. The extracted pairs need to give app designers an indication about which elements of the app are good and which are not. These elements represent low-level solutions to certain design problems and can aid app designers in generating creative solutions to their own design problem. Thus, the extracted features and opinions are there for better understanding the app. They should give app designers directions where to look at when assessing the app and should ultimately trigger creative thought processes. In best case, both the feature and opinion are shown to app designers. Hence, the importance of the quality of the opinion words should not be overlooked.

General guidelines

- While annotating try to determine per case whether it is an opinion or evaluation about the app, an opinion about a feature, or something else (e.g., bug report, referral to other app, thank you note to developer, feature request, etc.). Only opinions about features need to be included in the golden standard. Users can still express opinions regarding those other topics or regarding the app in general. However, these should not be included in the golden standard. While annotating, one needs to be aware about whether somethings is about the app in general or about a specific feature. This is in various cases very difficult due to the structure of the sentences or review in general.
- We see a **feature** something a user can do with the app (e.g., “tracking progress”), a ‘physical’ or visual characteristic of the app (e.g., “the UI”), or a quality of the app (“simplicity of the app”). Note: we make a distinction between features and aspects. We see features as a more restricted subset of aspects. So, not all aspects (i.e., things about which a user has an opinion) that are identified should be annotated as features.
- We see **opinions** as words that express a positive or negative statement (i.e., positive or negative sentiment). These do not only have to be verbs (e.g., “love”, “like”, “hate”), but can, for example, also be adjectives (e.g., “beautiful”, “nice”, “accurate”) or phrases (e.g., “could not be more satisfied with”).
- It is easiest to start searching for opinions. If a feature is mentioned without an opinion, it does not need to be considered.

Rules Applicable to Both Features and Opinions

- Feature opinion pairs need to be annotated per sentence and not per review. Opinions that are given in a different sentence from that of the feature should not be annotated.
- Only pairs need to be annotated; individual opinions or features (e.g., "simple to use" or "it works great") only are not sufficient.
- Features and opinions may comprise one or multiple words here. These words do not need to be adjacent (e.g., "works well offline" becomes "well + work offline").
- Some very long features/opinions with much contextual detail could sometimes also be listed in a shorter way (e.g., "good to log your activities with pictures and text" for "good + log your activities"). In those cases, list the feature/opinion in its entirety.
- When there is an opinion about a feature and a quality, then the combination of the feature and the quality should annotated as the feature (e.g., "love the simple UI" is annotated as "love + simple UI").
- In general, conjunctions should be split into multiple pairs, unless they are part of a larger feature (e.g., "the way it shows pictures"). Opinions with two or more adjacent adjectives need to be split into two pairs (e.g., "good simple recipes" becomes "good + recipes" and "simple + recipes").
- Spelling mistakes should not be corrected when annotating. We also do not expect the tool to do this.
- Symbols should not be annotated as features or opinions (e.g., '+' instead of 'plus').
- In case of doubt or in case it is not clear to what an opinion or feature is referring, then the specific case should not be annotated (e.g., when 'it' or 'that' are used).

Feature Rules

- We do not consider general words that refer to the app in general as features. Examples of these words are: platform, website (i.e., often mistakenly used to refer to the app), tool for. When an opinion is provided about those specific cases, then we do not recognise this as a feature opinion pair.
- We do not see specific words referring on their own to the general concept of the app or idea behind the app as features. Also, we do not see words referring to the business behind the tool or the business practices of the developer as features. Examples of these words are: concept, ideas, experience, customer service. Some of these words could be features when supplemented with more contextual information (e.g., "user experience"). Even though the word 'content' is somewhat general, it still a feature, since it is a specific part of the app.
- General words referring to features or results on their own should not be annotated as features, since they do not describe the feature itself. Examples include: feature, service, results. When supplemented with more information, these could be part of a feature (e.g., "editing feature").
- Also, we do not see general results in daily life or side effects of the use of the app (e.g., makes it easier to organise my closet) as features. Only if a specific feature results in something positive, then we see it as a pair (e.g., " scanner makes the process easier" becomes "makes the process easier + scanner").
- We do not see the lack of something or something the app does not do as a feature. We consider this a feature request or bug report.

Opinion Rules

- We consider words expressing a certain quality that is not a fact, but for which the value of it may be dependent of the person expressing it, as opinions. Examples of these words are: useful, expensive, (in)accurate, bad, calming.
- We do not see general observations or facts as opinions. Moreover, words that get their specific sentiment polarity by the context they are used in (e.g., "limited number of features" vs. "limited costs") are also not considered as opinions. Examples of words that are not opinions are: healthy, wide, big, short, brief, vast, limited, a variety of, lots of, many. These words may be annotated as opinions in case they are combined with other words (e.g., "a huge plus").
- It could happen that words are reinforced by certain words that are added to the phrase, in those cases the polarity is made specific (e.g., "took forever", "too short"). Those cases should be annotated as opinions.
- Opinion words that contain spelling mistakes should not be annotated as opinions, even when it is incorrectly spelled on purpose. The tool is not expected to work better than humans and to determined sentiments of words with spelling mistakes.
- Sometimes it is expressed that features need to be improved or need to be worked on. Only when this is stated explicitly, it should be annotated as an opinion (e.g., "the search bar needs to be improved" becomes "needs to be improved + search bar"). Examples include: needs to be improved, can be enhanced, needs help.
- In some cases the user can express that the feature is problematic or that it forms a problem (e.g., "it is a problem that users can insert text" becomes "problem + users can insert text"). Only in those explicit cases, the problem could be annotated as an opinion.

Other Specific Cases

- We do not see [adjective] app for [gerund] (e.g., "great app for communicating") as a feature opinion pair, but as an app evaluation.
- [adjective] way: way is part of feature but may be omitted (e.g., "I like the way it shows pictures" becomes "like + way it shows pictures" or "like + show pictures").
- Adjectives or verbs may be followed by 'to' or 'for'. We decided that these cases need to be treated differently. '[adjective] to [verb]' is seen as a feature opinion pair in case neither a feature nor the app is explicitly mentioned (e.g., "great to keep track" or "it helps to analyse the results"). Then it is considered to be short for "it is [adjective] to", indicating a positive or negative statement about the verb in question. '[adjective] for [verb]' should not be annotated as feature opinion in case no explicit feature is mentioned before the adjective (e.g., "great for keeping track"). In that case, it is considered as short for "the app is [adjective] for", which is an evaluation of the app. If a feature is explicitly mentioned (e.g., "the search bar is helpful for looking for things"), then it is considered to be a feature opinion pair (e.g., "helpful for looking for things + search bar").

Feature Opinion Extraction Performance Assessment Guide

The following guide is used for the assessment of the performance of the feature opinion extraction compared to a golden standard. Note: these guidelines are based on knowledge obtained in the manual review analyses and on own insights, while considering the objectives of this research. This guide treats subsets of feature opinion pairs in a stricter way, by counting not all subsets as (full) true positives (TP).

1. Precision and recall are assessed per pair. However, both feature and opinion can have influence on the score. If either one is too incomplete, then it could mean that the pair as a whole is not good enough and that the score is adjusted to that one.
2. For some extracted pairs either the feature or opinion is a subset of the annotated one. There can be multiple scenarios for this, which are handled differently (Table D.1).
3. It could also happen that both the feature and opinion are a subset of the annotated ones. In that case, both the feature and opinion are assessed separately using the rules of Table D.1. If both would be assigned the same value (i.e., either TP of 0.5 or 1 or FN and FP), then that value applies for the entire pair. If both are assigned different values (e.g., feature a TP of 0.5 and opinion TP of 1), then the lowest value is assigned. It is then also assessed if the entire pair still retained its meaning (to some extent). If not, assigned value is decreased accordingly.
4. When a completely wrong opinion is extracted for a right feature it is still counted as a FN and FP.
5. Splitting is handled leniently. If a feature was annotated in one pair and it was extracted in multiple pairs, then the individual pairs are accepted. When calculating the precision and recall, the initial number of annotated pairs is updated accordingly, to prevent miscalculations.

Table D.1: Specific scenarios and associated rules for extracted subsets.

Scenario	**Examples**	**Assigned Value**
1. The extracted feature is still relevant and useful. It does not miss essential information and it is still understandable.	Annotated: contacting travelling friends, way to upload pictures. Extracted: contacting friends, upload pictures.	This pair is assigned a TP value of 1.
2. The extracted feature is (somewhat) relevant, but it misses important information. Part of the essence/meaning is gone or when presenting the feature in the word cloud, the feature seems somewhat incomplete but still useful.	Annotated: photo book feature, suggestions for activities, personal interaction. Extracted: photo book, suggestions, interaction.	This pair is assigned a TP value of 0.5.
3. The extracted feature is too incomplete to be understandable or usable. The pair misses too much essential information and thereby loses its complete meaning.	Annotated: variety of recipes, editing option. Extracted: variety, option.	This pair is counted as a FP and FN.
4. The opinion is still understandable and the polarity is the same. The sentiment score may be a bit less strong, but that is accepted	Annotated: really useful, very nice, extremely helpful. Extracted: useful, nice, helpful.	This pair is assigned a TP value of 1.
5. Some essential information is missing. The meaning of the opinion is affected by that. However, the polarity is not opposite.	Annotated: user friendly, almost accurate. Extracted: friendly, accurate.	This pair is assigned a TP value of 0.5.
6. The polarity is completely the opposite of the annotated opinion or too much information is missing, causing the meaning of the opinion is completely changed.	Annotated: accurate, not like. Extracted: inaccurate, like.	This pair is counted as a FP and FN.

Appendix E

Experiment Instrumentation

This appendix presents all the material that presented to the experiment subjects. Also, the protocol for selecting appropriate apps is provided here. Several different apps needed to be be selected for the experiment, based on different criteria. However, each app needed to at least fulfil the following criteria:

- The rating of the apps needed to be 4.5 or higher.
- Only apps that were downloadable for free were selected.
- Games were excluded in the set, since those are outside the scope of the research.
- For each quadrant, except for the Near-field analogy quadrant, an as diverse set as possible set needed to selected. So, an app could not be selected if already a highly similar app was selected.
- The app description should be available in English.
- The app should not have a higher download rate than 10 million+. We reasoned that an app with a higher download rate would have a higher chance of being familiar to the subject.

Since some criteria were already introduced earlier, these are not explained again here. For some apps, a lower rating was accepted as well, as it would be hard to find a more appropriate substitution. Also, app ratings vary over time, so even during the experiment these values could change. We do not expect this to have a big impact. The following steps needed to be taken to create sets of apps for each quadrant:

1. Tag each apps with the final list of goals (Table 7.2).
2. Select the goals for the app of the toy design case of the experiment. These were for this experiment: *Search & discover*, *Communicate & share*, and *Get inspiration for new ideas.*
3. For each app, determine the recall with respect to the goals of the design case. Recall = {tagged relevant goals} / {relevant goals}[1].
4. Depending on the quadrant, filter out irrelevant apps. For *Surface similarity* and *Potential serendipity*, select apps where recall is 0. For *Near-field analogy* and *Far-field analogy*, select apps where recall is 1.
5. For each app, determine the recall with respect to all available goals. Recall = {goals} / {all app goals}[1].
6. For the remaining set of apps, select the ones with the highest recall.
7. Retain only apps that adhere to the criteria listed above.
8. List about fifteen to twenty keywords per app.
9. Determine the cosine keyword similarity[2] between each app and the design problem.

[1] Derived from Powers (2011).

[2] Cosine was used here, since this was the most feasible option given the constraints on the study.

10. Select the final set of apps based on the keyword similarity. For *Potential serendipity* and *Far-field analogy*, select the apps with the lowest similarity. For *Surface similarity* and *Near-field analogy*, select the apps with the highest similarity.

For the experiment, apps were sampled from the 100 apps that were used for the goal analysis. However, it quickly became apparent that other apps needed to be selected from the Google Play Store, as the set did not contain enough apps for each quadrant. Apps could only be sampled from the 100 apps for the Potential serendipity quadrant and partly for the Far-field analogy quadrant. For these apps, the steps above applied. For the other quadrants, apps were selected in a more targeted manner form the Google Play Store. For the Near-field analogy quadrant, recipe apps were selected that at least fulfilled all three design problem goals. For the Far-field analogy quadrant, additional apps were selected that at least fulfilled the three goals and were not about food or recipes. Apps were selected for the Surface similarity quadrant if they were about food, but did not fulfil any of the three goals. After that, the steps listed above were taken to evaluate their actual suitability.

After all sixteen apps had been selected, they needed to be divided into two groups. When grouping the apps, the following aspects were roughly taken into account to balance the sets as much as possible. The apps in the two groups needed to be relatively similar in terms of the recall of step five, the average rating, number of downloads, and keyword similarity among the apps. Each set needed to cover an as diverse as possible theme (i.e., overall app functionality) and a diverse set of goals covered by the apps. The characteristics of the selected apps and the groups to which they were assigned can be found in Table E.1 and Table E.2.

Table E.1: Sets of apps assigned to the two experiment groups.

	Group 1	**Group 2**
Potential Serendipity	Woebot: Your Self-Care Expert Money Manager Expense & Budget	Event Manager - AllEvents.in Forest: Stay focused
Far-field analogy	Houzz - Home Design & Remodel Travello Travel From Home	Amino: Communities and Chats Sports Tracker Running Cycling
Surface similarity	Macros - Calorie Counter & Meal Planner Food Diary	Food Checklist - Groceries Expiration and Shopping Calorie, Carb & Fat Counter
Near-field analogy	Cookpad - Create your own Recipes Easy Recipes	Kitchen Stories - Recipes & Cooking Craftlog Recipes - daily cooking helper

Table E.2: Characteristics of the sixteen apps selected for the experiment. * = *At the time of selection.*

Name	**Recall - Design problem**	**Recall - All goals**	**Keyword similarity**	**# downloads***	**Rating***
Woebot	0.00	0.24	0.05	10,000+	4.5
Money Manager	0.00	0.24	0.00	10,000,000+	4.7
Event Manager	0.00	0.24	0.00	10,000+	4.7
Forest	0.00	0.24	0.00	10,000,000+	4.6
Houzz	1.00	0.24	0.10	10,000,000+	4.7
Travello	1.00	0.24	0.00	500,000+	4.4
Amino	1.00	0.24	0.00	10,000,000+	4.6
Sports Tracker	1.00	0.29	0.00	10,000,000+	4.4
Macros	0.00	0.29	0.25	1,000,000+	4.5
Food Diary	0.00	0.24	0.28	100,000+	4.3
Food Checklist	0.00	0.18	0.32	10,000+	4.5
Calorie, Carb & Fat	0.00	0.24	0.32	1,000,000+	4.5
Cookpad	1.00	0.24	0.39	10,000,000+	4.7
Easy Recipes	1.00	0.24	0.47	1,000,000+	4.7
Kitchen Stories	1.00	0.35	0.46	1,000,000+	4.7
Craftlog Recipes	1.00	0.18	0.46	1,000,000+	4.4

The following pages display the questionnaire. As can be seen, each separate element was displayed on a separate page in the online questionnaire. Furthermore, only one general version of the questionnaire is shown here, since in general the questions were equivalent for both groups. Questions two through seventeen are displayed here in a general manner and only once. In the actual questionnaire, these questions were adapted to the specific sets of apps and were presented four times. For the two last questions, specific instances of the questionnaire are shown to show how they were actually represented.

Utrecht University

Questionnaire

Q1. Experience

How experienced are you in designing apps?

- ○ Novice
- ○ Intermediate
- ○ Advanced
- ○ Expert

New page

Case description

Please, read the following case description first.

A client asked you to design a recipe app from scratch. You have the freedom to come up with your own ideas and suggestions. Your client has only given you a few high-level requirements for the app to at least fulfil:

- *the end user needs to be able to search for new recipes;*
- *the app should give the end user inspiration for new cooking and recipe ideas;*
- *the end users need to be able to share recipes they created themselves with the community.*

You were asked to provide a set of ideas regarding functionalities of the app. The ideas should not be about the visual layout of the app, since that is not yet relevant for your client at this stage.

Press the “next” button to receive instructions on your task.

New page

Instructions

You will be shown **four sets of two apps**. These apps are meant to **inspire** you and help you come up with new functionality ideas for your client's app.

Steps for each set of two apps:

- **Assess the apps and try to come up with functionality ideas for the design task.** To do so, you need to click the provided links that will open the corresponding app page in Google Play Store. You may assess the entire app page of the specific apps (e.g., title, images, video, description, and comments) to get inspiration. *Tip: keep the webpages open, as you may need them for later questions.*
- Try to assess each app as if you actually needed to get inspiration for the app design and as if you needed to list new ideas and requirements afterwards. So, try to reason about new ideas and requirements when assessing each app. The ideas you may come up with do not have to be solely limited to the high-level requirements specified on the previous page.
- Answer **four short questions** regarding how the apps inspired you.

You will repeat this process four times with different sets of apps. After that, you need to answer **two final questions**.

While answering the questions, you can go back to this page and to the case description in case you need to refresh your memory.

New page

Q2 – Q17. Relevance, usefulness, tendency to transfer example elements & broadening of one's exploration space

These questions below are about the apps [App name 1] and [App name 2]. You can acess their page here: [Link app 1] and [Link app 2].

	Strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
1. These example apps seemed **relevant** to me **given my design task**.	○	○	○	○	○	○	○
2. These example apps were **useful** to me as a **source of inspiration** for my design task.	○	○	○	○	○	○	○
3. When generating ideas, I tended to **literally apply elements I saw** in these example apps to the design of the client's app.	○	○	○	○	○	○	○
4. These example apps **helped me to explore more options** regarding functionality ideas for client's app **than I would have without them**.	○	○	○	○	○	○	○

New page

Q18. Originality

When assessing the example apps above, for which apps did you find the ideas that came to mind most original? Please rank the sets based on **originality** of generated ideas from most original (highest rank) to least original (lowest rank). Note: this is a drag and drop question.

"Calorie, Carb & Fat Counter" and "Food Checklist - Groceries Expiration and Shopping"

"Kitchen Stories - Recipes & Cooking" and "Craftlog Recipes - daily cooking helper"

"Amino: Communities and Chats" and "Sports Tracker Running Cycling"

"Forest: Stay focused" and "Event Manager - AllEvents.in"

New page

Q19. Familiarity

Please select the apps you were already familiar with (i.e. only apps you have used yourself or that you have encountered before).

- ☐ Cookpad - Create your own Recipes
- ☐ Easy Recipes
- ☐ Woebot: Your Self-Care Expert
- ☐ Money Manager Expense & Budget
- ☐ Macros – Calorie Counter & Meal Planner
- ☐ Food Diary
- ☐ Houzz - Home Design & Remodel
- ☐ Travello Travel From Home
- ☐ None of them

New page

We thank you for your time spent taking this survey.
Your response has been recorded.

End of questionnaire