Data-driven creativity in Requirements Engineering

Devising a data-driven approach for increasing creativity in requirements engineering

Utrecht University

Final version
15-12-2019

Jasper van Noordenburg
4195981
j.m.vannoordenburg@students.uu.nl

First supervisor
Dr. Fabiano Dalpiaz
f.dalpiaz@uu.nl

Second supervisor
Dr. Sietse Overbeek
s.j.overbeek@uu.nl

Acknowledgements

I mostly want to thank my supervisor, Dr. Fabiano Dalpiaz, for helping me conduct this research and for all his feedback on this thesis, even when my motivation was less than desirable. He provided me with a lot of guidance and insights I would never have had.

Furthermore, I want to thank Dr. Sietse Overbeek for agreeing to become my second supervisor on short notice, after my initial second supervisor was unable to do so.

Finally, a huge thank you to my family and friends that supported me, as well as telling me to get to work and get moving when I lacked motivation or was postponing writing the thesis.

Abstract

This research explores the possibility, usefulness and effectiveness of data-driven methods and techniques in supporting the creative process of requirements elicitation in projects with a highly adaptive and changing environment allowing little time for RE activities, such as agile software development. First, the concept of requirements elicitation is explored, as well as the various techniques used during elicitation. Afterwards a lightweight and accessible technique, called *Feature.Ly* is developed to aid users during idea generation. The application provides examples based on features extracted from app reviews retrieved from the App Store to trigger users into generating new ideas or adapt the examples to their situation. Evaluation of the application resulted in no significant effect when comparing idea generation without and with the application. However, a number of insights are gathered to guide future work and areas of research.

Keywords: *Requirements elicitation, agile, data-driven, creativity, app review*

Table of Contents

List of tables

List of figures

1. Introduction

In the current economy, creativity and innovation are ever so important as increasing globalisation and technological developments create new business opportunities but simultaneously increase competition and market crowding [1], [2]. As such, an increasing amount of pressure rests on Requirement Engineering (RE) professionals to tailor to the needs of stakeholders to provide creative and innovative software and systems solutions in order to gain a competitive advantage.

Even though the systematic study of creativity has been around since the 1950s and has been widely studied [3], creativity in requirements engineering is relatively novel. Until early research on creativity in RE, the process was seen as a series of defined and recurring activities that produces a set of quality requirements [4]. This view has shifted since Maiden and Gizikis reported that the process of RE is highly creative and more research should be conducted in order to support, and find the effects of creativity on, the engineering process [5]. Since then, researchers and practitioners have gradually moved to a creative process in which stakeholders and designers work together to create ideas for new systems that are eventually expressed as requirements [6].

Furthermore, requirements were traditionally considered to be within the mind of stakeholders in an implicit manner and could be extracted through various elicitation techniques [7]. Nowadays, the consensus is that stakeholders cannot know everything as new technologies evolve beyond what was thought possible [8]. As such, stakeholders need to relate their domain knowledge to these technologies to produce creative requirements and should make us of tools that support them in this process [8].

As these views shifted, a number of methods, tools and techniques have been researched and developed in the field of RE creativity to support the elicitation process. An example is creativity workshops [5] in which experts design sessions to support stakeholders in envisioning ideas and encourage creativity. Activities include random idea generation, presentations from experts from other domains and listening to music and discussing paintings [5], [6]. RESCUE [9] is another example. It integrates human activity modelling, system goal modelling, scenario walkthroughs, and best practice requirements management to determine stakeholder requirements. ART-SCENE [10] also works with scenario walkthroughs. It automatically generates scenarios and possible alternative courses based on a list of use cases to be used during discussions with stakeholders. A final example is iThink [11], a game aiming to improve collaboration and stakeholder involvement during requirements elicitation by introducing gamification into its tool.

Boden's three categories of creativity can be used to classify the developed methods according to the way they stimulate the occurrence of creativity. According to Boden, creativity happens by either combining familiar ideas to create new ones (Combinational creativity), by exploring the potential and limits of a conceptual space to come up with new concepts (Exploratory creativity), and by dropping or altering existing characteristics of a space in order to create new concepts and beliefs (Transformational creativity) [12].

These methods have been proven to be helpful in supporting creativity in requirements elicitation but have shortcomings too. They rely on the capabilities and knowledge of the requirement researchers and practitioners, or the information available in the scope of the project to improve their own creativity. Furthermore, they require a lot of time for preparation, planning and execution. Therefore, they are not suitable for projects in which there is little time for RE activities, such as projects that use agile development methods [13].

Using other, external, data allows the creativity process to be influenced by the experiences of other practitioners, projects and experiences on a much larger scale than possible through peer-to-peer knowledge sharing. Additionally, this can improve the efficiency of the method and allow for its use in projects with little time for the RE process.

1.1 Context and Background

Before discussing the details, it is worth describing the context and background upon which this research is build. It also provides the scope for this research as to give a better understanding of the intended use of the method and its positioning in the complex and broad process of RE.

Requirements engineering has several core phases which are all intertwined and iterative [4], [14]. *Context/Domain analysis* is the preparation phase in which the focus lies on identifying and understanding the domain in which the system will be built and its stakeholders. In some projects this phase is made shorter or less effort is put in as the problem domain and its stakeholders are already clear. The next phase is *Elicitation*. Often regarded as the most important phase of them all, it aims to find what problems need to be solved and hence identify requirements [4], [14]. After identification, the *Analysis and modelling* phase is concerned with analysing the found requirements for any deficiencies and documenting them in such a way that stakeholders understand them and their rationale. *Evolution* is the final phase and focuses on managing changes in the requirements as a result of corrections, changes in the environment, and new objectives. It often requires revisiting previous phases and evaluating the effect of possible change on the entire project.

This research focuses only on the *Elicitation* phase of the RE process as this is where new and novel (i.e. ‘creative’) ideas are to be discovered [7]. Furthermore, this phase is most often revisited due to its iterative nature [15] and the fact that new insights in a project’s objectives and changes in stakeholder needs often lead to the addition or changing of requirements [4]. Thus, the *Elicitation* phase seems the most appropriate to develop a creativity approach that is applicable and useful in frequently changing environments with little time for planning.

Additionally, this research is a follow-up of the thesis project by Niels Wever during his Masters Business Informatics [16]. His thesis explored creativity in RE and created a program to synthesize requirements using natural language processing. Semantic role labelling was used to identify parts of user stories that can be reused and combined to create potentially new, novel, and useful requirements. This program has shown promising results in the production of new requirements but has some issues in ensuring they are meaningful. Rather than the automatic generation of meaningful requirements that are creative, this research focuses on the use of creativity triggers, so stakeholders create their own creative requirements.

1.2 Research questions

This research explores the possibility, usefulness and effectiveness of data-driven methods and techniques in supporting the creative process of requirements elicitation in projects with a highly adaptive and changing environment allowing little time for RE activities, such as agile software development. As a result, the main research question is formulated as follows:

> *RQ: How to devise a data-driven approach to increase creativity for requirements elicitation in agile software development environments?*

In order to properly understand the domain and existing solutions a few sub-questions need to be answered as preparation. As mentioned before, creativity in RE is relatively new and its definition is vague. A proper understanding of what creativity in requirements elicitation entails is required in order to determine whether new methods actually improve it. This results in the first sub-question:

> *SQ1: How can creativity be increased during the RE elicitation process?*

Additionally, an understanding of the elicitation process, its techniques and current efforts is required to construct a taxonomy to determine which techniques are suitable for a data-driven approach. This results in the third sub-question:

> *SQ2: Which elicitation techniques are suitable for a data-driven approach?*

These two sub-questions give a proper overview and understanding of the problem area and the current efforts. The next steps involve determining what kind of data can be used to potentially increase creativity and figuring out how this can be incorporated in creativity techniques and methods, resulting in the next two sub-questions:

> *SQ3: What data can potentially be used to increase creativity in requirements engineering?*

> *SQ4: How can this data be used to improve current or create new creativity-fostering techniques?*

Finally, the effectiveness and usefulness of the new methods need to be examined and compared against traditional techniques, which is formulated in the last sub-question:

> *SQ5: How effective are techniques enriched with data?*

1.3 Outline

The next chapters start with explaining the method used to answer the questions stated in the previous section after which the literature is explored. Subsequently, a technique is designed and evaluated, ensuring its feasibility for testing within the scopes of this research. Then an experiment is constructed to test the technique’s effectiveness and the results are presented. Finally, the results are discussed, conclusions are drawn, and recommendations for future work are made.

2. Research method

In order to answer the five sub-questions, a couple of phases are needed. First, a thorough understanding of the problem is required to be able to extract characteristics to be used in the creation of a new method or selection of an existing method to be altered. Afterwards, the method needs to be validated to determine whether or not it actually improves creativity in requirements elicitation.

These required steps are a good fit for the design science method. The design science method is the design and validation of solution proposals to practical problems [17]. It combines *knowledge problems* – the difference of stakeholders' current knowledge and what they would like to know – and *practical problems* – the difference of how stakeholders experience the world and how they would like it to be [18]. The method is an iterative process, called the *regulative cycle*, and comprises of four phases: *Problem investigation*, *Solution design*, *Design validation*, and *Solution implementation* [18]. The cycle, adapted to this research, is visualized in Figure 1.

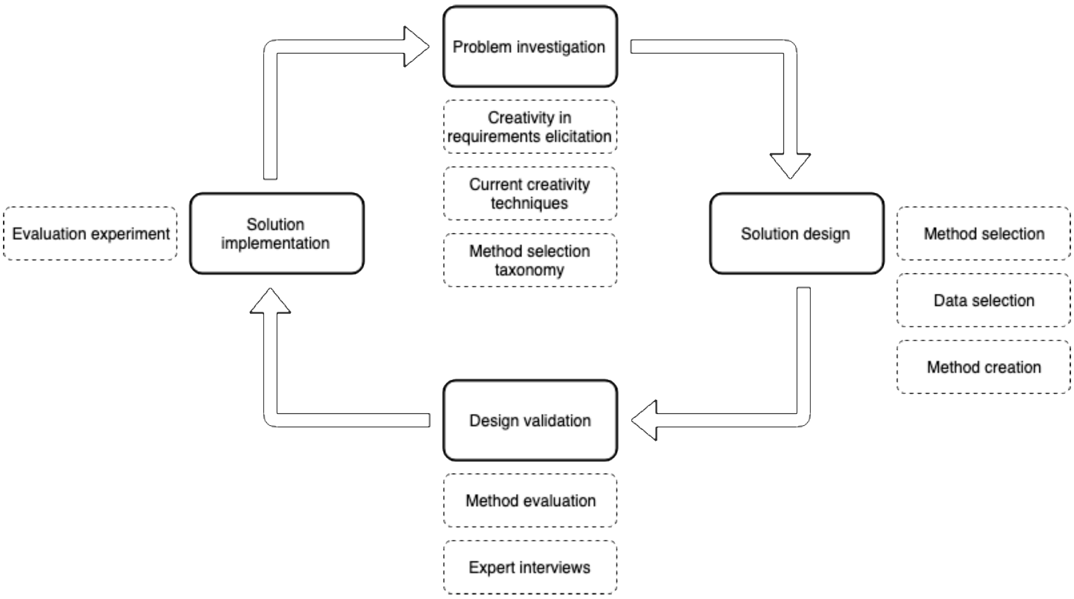


Figure 1 Research method design cycle

Problem investigation is concerned with understanding the problem and explaining it based on knowledge problems. For this research, this phase will answer SQ1 through the study of previous work and current efforts in creativity by means of a literature study. Additionally, a literature study of current elicitation techniques and the creation of a taxonomy to select a method suitable for a data-driven approach will be used to answer SQ2. This phase's results will serve as a basis of method selection during the conclusion of method selection and idea generation during SQ3.

Solution design comprises of designing the artefact that will help in solving the stated problem. SQ3 & SQ4 will be answered in this phase by generating ideas on data generation by analysing the use of data in creativity in other fields. These will be related to the previously examined elicitation techniques to select the best combination of method and data use.

Design validation and *Solution implementation* are concerned with asking if the designed solution will actually bring stakeholders closer to their goals. This research combines these two phases to answer SQ5. First, the created method will be assessed through critical analysis to determine the viability of gathering the data required in a timely manner that does not exceed traditional elicitation techniques. Afterwards, the method will be tested by means of an experiment to see whether or not it actually improves creativity during requirements elicitation.

3. Literature

This chapter covers the existing literature about requirements elicitation and how it can be executed. Additionally, it describes creativity as a concept and how other work has created methods and techniques to improve it in RE. The strengths and shortcomings of these techniques are described and related to this research's goals to identify its relevance and gap in the literature.

3.1 Requirements elicitation

Requirements elicitation is concerned with learning, uncovering, extracting, surfacing, or discovering the needs of customers, users, and other potential stakeholders [19] and is generally viewed as one of the critical activities in requirements engineering and a key factor in the potential success of a software project [15], [20].

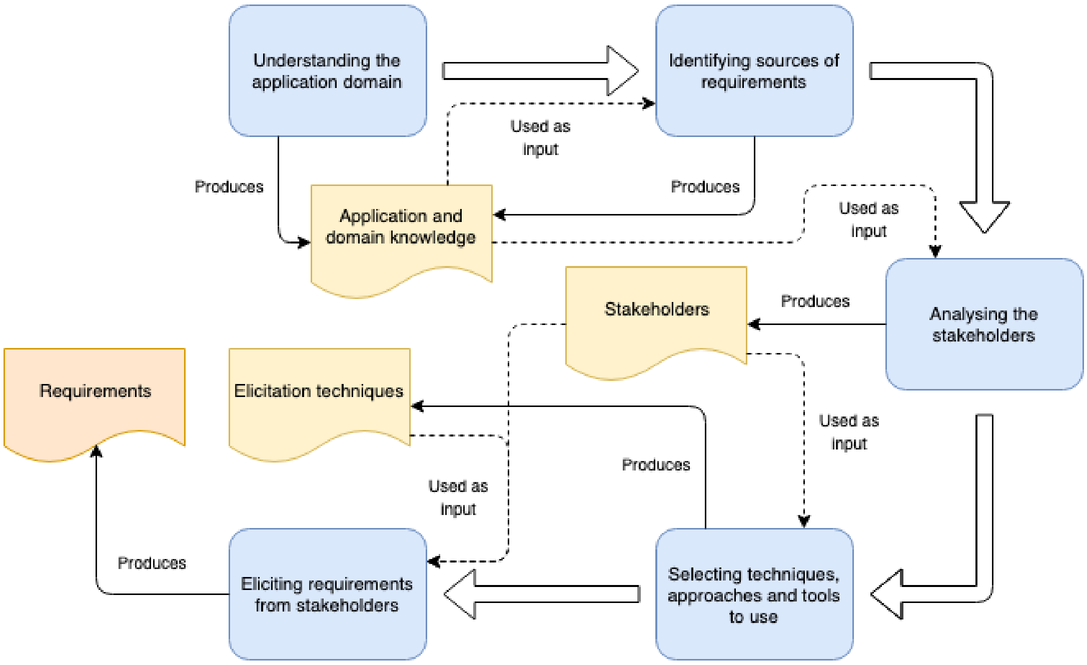
The elicitation process is highly complex and involves many different activities, each with their own methods and techniques, which can be divided into five fundamental types of activities [15], as visualized in Figure 2. The first type of activity is *Understanding the application domain. It* involves the detailed analysis and examination of the domain which the to-be system will occupy and is typically carried out at the start of the RE process. It explores all aspects related to the current system and takes into account existing processes and business rules that need to be incorporated into the to-be system.

Using the knowledge gained about the application domain, it is possible to *Identifying the sources of requirements*. This phase comprises of activities to identify stakeholders, documents, existing systems, regulations, etc. that could provide analysts with requirements or with information that could lead to requirements.

Next is *Analysing the stakeholders*, a type of activity that involves analysing the people that have an interest in the system and involving those that are relevant to the software development process. These people typically include employees of the organisation but can also include the users of the to-be system, if they are external and are typically already identified in the previous type of activity.

When the sources of potential requirements are discovered and involved, *Selecting the techniques, approaches and tools to use* follows. There is never one method or technique that can be applied to all situations and result in quality requirements. The selection of techniques is highly dependent on the context of the projects, the knowledge and experience of the analyst, and how well they complement each other [19]. Furthermore, the selected techniques depend on the stage of the project and can be swapped out when elicitation goals change as the projects progresses.

Eliciting the requirements from stakeholders and other sources is the final type of activity. The previous activities come together as the requirements are elicited from the stakeholders and other sources of requirements, using the selected tools from the previous activity.



Understanding the application domain
Identifying sources of requirements
Used as input
Produces
Produces
Application and domain knowledge
Used as input
Analysing the stakeholders
Stakeholders
Produces
Requirements
Elicitation techniques
Used as input
Produces
Used as input
Produces
Eliciting requirements from stakeholders
Selecting techniques, approaches and tools to use

Figure 2 The five fundamental activities of requirement elicitation

As mentioned before, elicitation is often revisited during the RE process as changing project objectives, new stakeholder insights, evolving technology, or the discovery of other requirements can result in the need for the addition or changing of existing requirements [4]. Therefore, it has no definitive place in the RE process but is often performed after initial domain analysis and revisited whenever new knowledge and understanding of the to-be system arises. Due to its iterative nature and frequent revisiting, elicitation takes up the bulk of the time spent in the RE process [19], as illustrated in Figure 3.

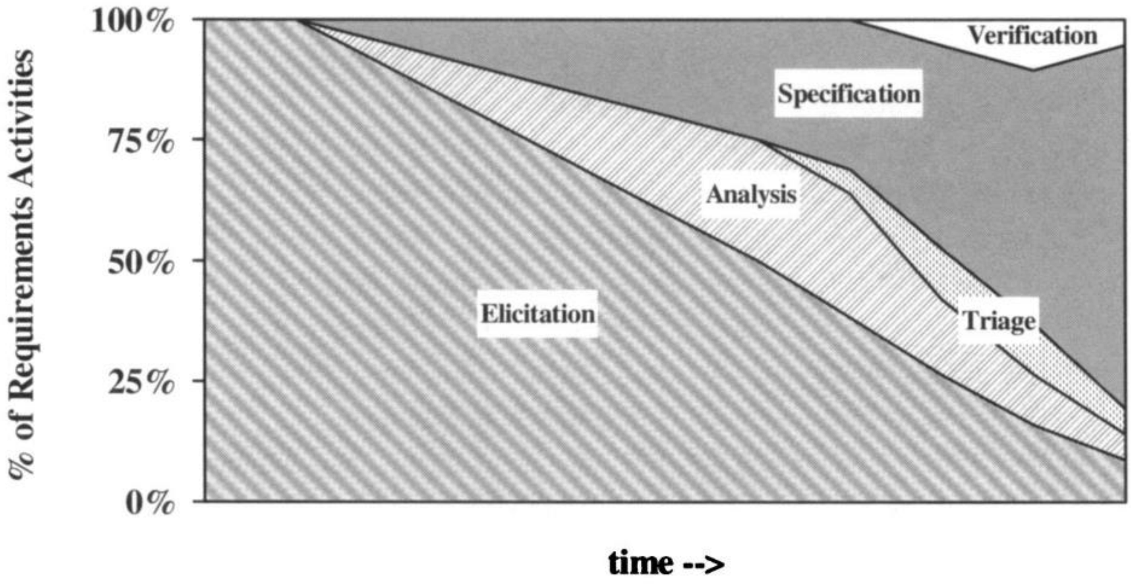


Figure 3 Time spent per activity as time progresses through the RE process [19]

3.2 Problems in elicitation

Since stakeholders often face challenges in identifying and articulating their needs, and thus the requirements, cooperation, communication, and mutual understanding between RE analyst and stakeholders are key during the elicitation process [20]. As such, these are often the cause for problems during the elicitation process. There are three categories of problems that can typically occur and may have a major impact on the time and cost of the project [21].

Problems of scope is concerned with the problem in which requirements are created with the wrong scope in mind or without proper understanding of the application domain. This may result in requirements that either provide too much or too little information and those that do not reflect users' needs. For instance, focusing on and specifying the exact details of a requirement may take needed time away from other areas that need to be addressed, resulting in poor quality requirements. Furthermore, if the domain is not properly understood, analysts may make assumptions that result in unusable requirements due to environmental or organizational factors and constraints.

Problems of understanding are problems created due to poor communication between stakeholders and analysts. This can lead to requirements that are ambiguous, incomplete, and incorrect. Problems of understanding can be divided into three issues. One of the causes of these problems is the difference in backgrounds and experience of the various groups. One may see a concept as an everyday issue while another may not even know of its existence or describes it in a completely different manner. This makes it difficult for the analyst to integrate the information and prioritize it accordingly. Another cause is the language being used and their complexity. Some groups may not understand the requirement due to the formality of the language and thus may perceive it as less important. The same issue occurs when requirements become very complex, resulting in groups no longer understanding the focus and necessity of the requirement.

Problems of volatility occur due to the constantly changing environment of requirements. Environmental or organisational factors may change during a project causing a need for changing or adding requirements. If this change is not embraced, or at least taken into account, the final set of requirements may describe a system that is no longer feasible in the current domain. Additionally, users may not have clearly understood or expressed their needs in the initial phases. If these evolving needs are not taken into account as the project progresses, the system's functionality may not be usable or may provide only a marginal improvement over current process.

3.3 Elicitation techniques

The elicitation process uses a multitude of methods and techniques, each with their own use and effectiveness based on situational context, as mentioned before. Nuseibeh and Easterbrook [4] distinguish between six classes of elicitation techniques, which were extended by Tuunanen [22].

Traditional techniques are a broad class of early elicitation techniques and includes typical methods of data gathering that were used as the process of elicitation developed. Examples are interviews, surveys, questionnaires, and analysis of existing documentation, process models, and existing systems. With the exception of interviews, these techniques are typically single-directional, meaning the analyst studies existing material to elicit requirements.

In contrast to traditional techniques, *Group elicitation techniques* are almost by definition two-directional as a result of free communication between participants. The wide range of methods target groups of stakeholders and users to elicit requirements. They aim to foster stakeholder agreement and buy-in, while simultaneously using team dynamics to gain a better understanding of user needs. Group techniques allow for the involvement of more people in the development process, aiming towards more representative requirements and user needs. Methods used in group techniques include brainstorming, focus groups, rapid application development (RAD), and joint application design (JAD).

Prototyping is typically used when analysts are uncertain about requirements and need to validate them, as well as moments when early feedback from stakeholders and end-users is required. Prototypes can be used as an aid in other classes of elicitation techniques, such as the basis of questionnaires and interviews or provoking a discussion during a group elicitation technique.

Cognitive techniques were originally developed for the acquisition of knowledge for knowledge systems and have been adapted for, or fitted into, the RE process. Techniques typically include the analyst as an observer to a task being performed to gain knowledge about the cognitive processes required for the task at hand. Examples include protocol analysis, in which an expert thinks aloud while performing a task and an analyst observes to gain insights into the cognitive process required, and laddering, in which the analyst uses probes to elicit the structure and content of stakeholder knowledge.

Similar to group techniques, *Contextual techniques* aim to obtain requirements information from stakeholders by enriching communication. They emerged as an alternative to traditional and cognitive techniques in the 1990s [23] and are steered by three general guidelines provided by Holtzblatt and Beyer [24]. Examples include ethnographic techniques, ethnomethodology, and conversation analysis, which all use fine-grained analysis to identify patterns in stakeholder conversations and interactions.

Rather than the relatively unstructured classes of eliciting requirements, *Model-driven techniques* provide a specific model of the type of information to be gathered. Examples of techniques are goal-based methods, such as KAOS [25], and scenario-based methods [26]. They require a thorough understanding of the application domain or knowledge of work practices to be applied successfully. As a result, these techniques can only be applied to user representatives rather than end-users, as they will almost never possess this required level of know-how.

The various techniques belonging to these classes will not be discussed in detail, as there are too many and new techniques are constantly being developed. Table 1, however, provides an overview of a selection of the most used techniques and their characteristics.

Table 1 Overview and characteristics of various elicitation techniques based on literature.

Method	Description	Characteristics
Introspection [15], [23]	Method in which the analysts imagines the system and its requirements by placing themselves in the users' shoes.	+ Very useful as a starting point + Can be very powerful in combination with other data gathering/domain understanding techniques - Can be, and often is, highly inaccurate - Later in development, introspection of an expert in a different field, such as RE, is unlikely to directly match what a user deems necessary
Interviews [15], [27]–[30]	Verbal method in which data is gathered through a conversation between two or more people. Can be divided into three types: unstructured, structured, and semi-structured interviews.	+ Accessible and efficient way of collecting large amounts of data quickly + Good for exploration of complex topics and when there is a limited understanding of the domain. + Gives immediate feedback + Allows for the elicitation of information not previously expected - Can limit the production of new concepts if too structured - Quality of data depends on interviewer (i.e. loss of focus may result in certain areas being skipped) - Require a lot of effort and time
Questionnaires [15], [28], [30]	Technique used to collect requirements from a large group of a population by defining a list of questions and sending these to participants to be answered.	+ Reaches a large number of people in a short timespan + Economical + Useful for uncovering relationships and uncertainties - Domain must be properly understood by participants - Questions need to be formulated carefully to avoid confusion among participants - No possibility for clarification questions
Task analysis [31], [32]	Technique used to analyse a system in terms of user goals and sub-goals during the task. Focuses on and aims to model the structure of the tasks that need to	+ Allows for the analysis of inconsistencies between user actions and identifying unnecessary functionality + Ideal for checking completeness of a design - Doesn't allow for novel ways of carrying out tasks if a system's design is based purely on how a task was performed previously

	be performed in the system.	
Domain analysis [27], [33], [34]	Analysing existing documentation and related applications to gather requirements through a systematic approach within an organization's field of operation.	+ Provides opportunity to reuse requirements. + Information is readily available - Previous systems development has to be consciously made reusable for it to be effective and efficient - Relies heavily on the analyst's experience, which is hard to obtain and takes considerable time - Lower creativity and novelty due after many iterations of systems in the same industry
Observation [15], [28]	Technique in which the user's environment is observed by the analyst to elicit requirements without directly interfering.	+ Very useful when stakeholders are not able to articulate/ do not know what they want in the system + Provides early insights in how users will interact with the system - Very time consuming - Success depends heavily on skill of analyst - Users tend to change their ways when knowingly being watched
Brainstorming [15], [28], [35]	Informal discussion sessions where participants are free to express their ideas without criticism. In a different part of the session, the generated ideas are evaluated and discussed.	+ Easy to implement and execute + Promotes free thinking + Allows for discovery of new solutions to existing problems - A high quantity of ideas does not mean they are also high quality - Groups in brainstorming settings generate fewer ideas than groups in 'normal' collaborative settings - Participants may get intimidated or become hesitant due to dominant people in the group
Prototyping [15], [28]	Iterative process in which a mock-up or dummy version of the system is given to users to acquire feedback	+ Early user involvement and feedback + Encourages stakeholders and users to actively participate in development - Generally, has to be used in conjunction with other techniques, such as interviews to process the feedback properly - Expensive and time consuming to produce - Users may become attached to a certain solution and become resistant to alternative implementations

Scenarios [9], [28], [36]	Uses representations of users’ interactions with the system as they would be in the real world. Can be used to generate new requirements by generalizing the real-world examples or as test scripts during evaluation.	+ Focuses on reality and forces analysts to go into detail, ensuring that abstract models are valid + Gives proper clarification regarding the natural flow of a task or activity, improving understanding for stakeholders + Easy to understand, use, and implement - Often encourage confirmation-bias where error handling is not covered in the written scenarios - Obsession on details in scenarios can make them too large and unwieldy - Cannot be used without understanding current work flow
Group work [15], [28], [37]	Technique where stakeholders are included in a meeting to elicit requirements in collaboration which is moderated by analysts to keep focus and encourage participants.	+ Saves costs compared to conducting interviews with all participants + Produces quality requirements in a shorter period of time - May be difficult to organise and schedule due to number of participants - Time consuming - Only useful for small projects
Protocol analysis [15], [23], [28]	Users are asked to think aloud while performing their tasks while an analyst observes.	+ Useful in understanding how each person solves problems + Is useful at any stage in development + Easy and low-cost - Time consuming - Having to keep prompting a person for explanation can take away from the natural flow of the process - Requires introspection during explanation, making the technique unreliable

3.4 Creativity in requirements elicitation

As mentioned before, researches and practitioners have shifted towards a view in which RE is seen as a creative process in which stakeholders and designers work together to create ideas for new systems that are eventually expressed as requirements [6]. It was also mentioned that the systematic study of creativity has existed long before this view was adopted and has been very extensive in developing creativity theories and models [3]. These theories and models can then be integrated into existing RE processes or adapted to fit the needs for RE activities.

Boden defines *Creativity* as: "The ability to create ideas that are new, surprising and valuable" [38]. In this definition, *Ideas* are any artefacts that can be produced. Furthermore, *valuable* describes the idea in terms of it being useful for the task at hand or the project. The second characteristic, *new*, is more complex and has two different meanings based on its context. *Psychological*, or *P-creative, ideas* describe ideas that are new to the person who thought of it. An idea or concept can be thought of thousands of times but still be creative when looking at it from a psychological perspective. *Historical*, or *H-creative, ideas* on the other hand are ideas that have never occurred in history before. As a result, *H-creativity* is automatically also *P-creativity* as the idea cannot be historically creative if the person has thought of it before. *Surprising* is the final characteristic and has three different meanings. Ideas may be surprising when they are unfamiliar or unlikely, such as winning the lottery. Another meaning of surprise is the realisation that an idea fits in your style of thinking and you did not expect this. The final meaning is the disbelief that you came up with a seemingly impossible idea. This final meaning of surprise may even result in other ideas that were deemed impossible before.

Defining creativity does not explain the process of forming ideas that are creative. Creativity is not something that can be turned on at will or happens spontaneously but requires cooperation, dialogue and interplay between individuals and is a process of problem solving [39], [40]. A number of techniques have been developed to improve creativity during the elicitation process.

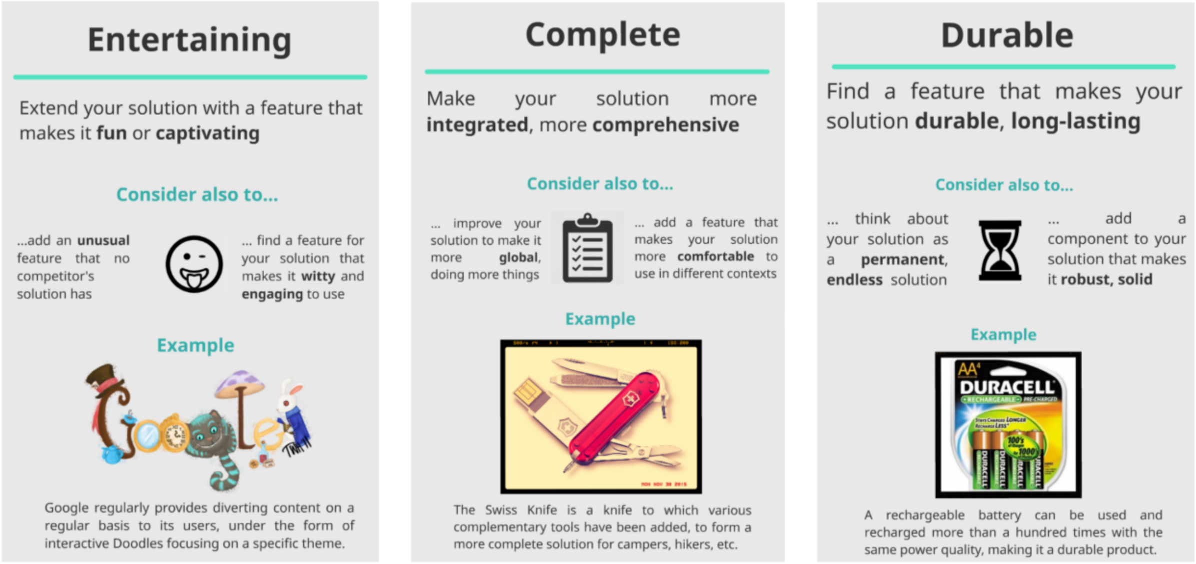
RESCUE is developed by Jones and Maiden [9] to specify requirements in complex systems using human activity modelling, system goal modelling, scenario walkthroughs, and best practice requirements management to determine stakeholder requirements. These are then used to develop creativity workshops. *RESCUE* is one of the main contributions with regard to creativity in RE and has been successfully used in practice during development of a system for air traffic management [7], [9].

ART-SCENE uses similar elements. Developed by Zachos and Maiden [10], it automatically generates scenarios from use cases, generates possible alternative courses, and provides guided walkthroughs through these scenarios. Each walkthrough provides stakeholders with the opportunity to recognise whether requirements are specified, or new requirements are necessary. Zachos and Maiden mention the intention of using the technique on a larger scale at a future project but there are no reports on its result.

iThink [11] takes a completely different approach through the combination of game mechanics with a creative thinking technique, called *The Six Thinking Hats.* Each hat represents a different perspective of looking at the requirements being discussed. For instance, the white hat focuses on facts and numbers in a neutral way. The red hat is concerned with emotions and feelings opposing the neutral information given by the white hat. The black hat relates to negative judgements and explains why an idea may not work. Positive comments about ideas are covered by the yellow hat and the green hat is the perspective of creating new ideas and alternatives. Finally, the blue hat represents global vision and a focus on the problem definition. Whenever players make comments, the used hat (perspective) determines the amount of points they receive. More points are awarded as the perspective taken contributes more to the overall goal of the game. For instance, a new requirement – using the green hat – is worth more points than a positive or negative comment – using the black or yellow hat – and a positive or negative comment is worth more than a star rating – red hat – or concrete comment – white hat. No applications of the *iThink* game used in practice have been found but the technique has been used in two case studies to validate the prototype.

EPMcreate [41] is another technique that focuses on approaching a problem through taking on different roles. Based on the Elementary Pragmatic Model (EPM), a tool developed for relationship therapy to help analyse patterns of interaction, *EPMcreate* uses this tool in reverse. Rather than deducting behavioural patterns through observation of interactions, *EPMcreate* uses the tool to generate interactions between two stakeholders with different positions. These interactions are then used to create new viewpoints that overlap the stakeholders' viewpoints from which creative ideas may flow. This technique can be repeated for each possible pair of stakeholders, but the number of pairs should be carefully chosen by the analyst through the potential of new information.

The *Creativity trigger* technique, created by Burnay, Horkhoff and Maiden [42] is an attempt to create a lightweight creativity technique that is less time and resource consuming to successfully execute. A *creativity trigger* is a guideline for groups of stakeholders that describes a quality of a product associated with innovation. These guidelines are represented on a card and examples are given to stimulate possible combinations and use stakeholders' experiences to generate creative requirements (combinational creativity). A few examples of such cards are shown in Figure 4.



Entertaining

Extend your solution with a feature that makes it **fun** or **captivating**

Consider also to...

...add an **unusual** feature that no competitor's solution has

... find a feature for your solution that makes it **witty** and **engaging** to use

Example

Google regularly provides diverting content on a regular basis to its users, under the form of interactive Doodles focusing on a specific theme.

Complete

Make your solution more **integrated**, more **comprehensive**

Consider also to...

... improve your solution to make it more **global**, doing more things

... add a feature that makes your solution more **comfortable** to use in different contexts

Example

The Swiss Knife is a knife to which various complementary tools have been added, to form a more complete solution for campers, hikers, etc.

Durable

Find a feature that makes your solution **durable, long-lasting**

Consider also to...

... think about your solution as a **permanent, endless** solution

... add a component to your solution that makes it **robust, solid**

Example

A rechargeable battery can be used and recharged more than a hundred times with the same power quality, making it a durable product.

Figure 4 Examples of creativity trigger cards [42]

The mentioned creativity techniques have at least one characteristic in common. They force stakeholders to view a problem or situation from a different perspective to gain new insights from which new requirements can flow. At some unexpected point in time, stakeholders will use the new insights to restructure the problem which causes a sudden understanding and realisation of the required solution, also known as the 'Aha moment' [43], [44].

3.5 Requirements elicitation and creativity in agile environments

Development in agile environments differs significantly from the environment in which traditional RE, and thus requirements elicitation, is practised [13]. Traditional elicitation relies on extensive analysis and information gathering to create an understanding of the to-be system and validating this until it is as complete as possible [4], [45] as described in section 3.1 above. As a result, traditional elicitation may require a lot of time during the project or before actual development starts [13].

Agile methods, however, do not allow for such extensive preliminary analysis and planning. Developed as an answer to increasingly lengthy and frustrating RE processes that produced project plans that got out of date shortly after a project started [45], they embrace the unpredictability and high rates of changes of software development projects [46]. The focus of agile methods lies on the collaboration between people rather than execution of processes [47]. In order to preserve the principles behind agile methods, the *Agile Manifesto* was created in 2001 [13], [48], which describes four core values: *Individuals and interaction over processes and tools*, *working software over comprehensive documentation*, *Customer collaboration over contract negotiation*, and *Responding to change over following a plan*. As a result, these characteristics and core values do not allow for the creativity techniques as described in section 3.4 to be successfully applied in an agile environment, given their complexity, resource, and time requirements. One exception may be the *Creativity triggers* technique [42], which was developed as a lightweight alternative to the resource intensive methods.

Nevertheless, creativity is key in agile methods. As mentioned before, agile methods do not rely on processes to manage a project but rather people, their creativity, and their collaboration [47], [49]. Regardless the reliance on creativity, little work has been published towards enhancing it in agile environments.

Hollis and Maiden [50] have explored the possibility to inject creativity into agile processes. They did this by selecting two places within the agile process at which creative activities could prove useful; The envisioning process in sprint zero in order to discover more novel epics, and at the beginning of certain sprints when epics are developed into more specific requirements that can be developed. The extended process was found to produce more novel requirements, that were viewed as less useful until further specification allowed for subsequent idea incubation.

4. Design cycle: BrainstormPlus

There are a number of things required in order to start working on a technique that is data-driven, lightweight, and suitable for agile environments. First, a further analysis of current techniques is necessary to see if there is a baseline to adapt, as the technique does not need to be built from scratch. This is done through the construction of a taxonomy and subsequent selection of viable techniques from the elicitation techniques outlined Table 1.

4.1 Characteristics

The following characteristics have been defined to determine and select the techniques best suited as a base for a possible feasible technique developed in this research.

C1: Co-presence of participants

This characteristic focuses on whether or not the technique has to be executed with all participants present (e.g., *Brainstorming* or *Group work*) or can be done individually remotely (e.g., *Introspection* or *Questionnaires*). While individual idea generation tends to be more productive and creative compared to group techniques [51], successful software development – and thus requirements elicitation – is a team effort. Ideas should be discussed among stakeholders to ensure their needs are met.

The selected technique does not necessarily have to be a group-focused technique but it should have the ability to be one in order to facilitate discussion and feedback within the group.

C2: Output divergence

The divergence of the output of a technique compared to the input determines whether new requirements are being generated or it is used to gain domain and existing system knowledge. This is important for understanding and starting from an existing system upon which to innovate by using techniques such as *Task Analysis* and *Protocol analysis*.

Given this research's goal, however, it is important to use a technique focused on generating new and creative requirements, not describing what is already available or evaluating previous ideas.

C3: Output granularity

Granularity of the output is focused on the detail of the ideas being generated. Does the technique focus on generating creative ideas that require further discussion and specification (i.e., *Epics*) or does it focus on evaluating pre-existing requirements or ideas and improving their specification?

Both are important for successful requirements engineering but, given this research's focus on creativity, generating ideas that are novel and require further specification is more important for selecting a base technique as these ideas allow for additional idea generation through Boden's categories of creativity [12].

C4: Depth of domain knowledge required

This characteristic evaluates the knowledge required before someone can be an effective participant during the technique sessions. While the required domain knowledge fluctuates between sessions, analysis and evaluation techniques such as *Task Analysis*, *Observation*, and *Interviews* require a deeper understanding about the domain than generation techniques such as *Brainstorming*.

Experts are important for evaluating requirements and ensuring they are attainable and useful for the project, but in order to ensure accessibility and participation among stakeholders a technique that generally requires less domain knowledge would be favourable.

C5: Time required
The time required focuses on how much time is spent to prepare, execute, and evaluate the technique's sessions. While not necessarily crucial in traditional RE activities, as they allow for plenty of planning, time is scarce in agile environments [13]. Furthermore, the constantly changing environment does not allow for long preparation phases. Therefore, a technique that requires little to no preparation is preferable as it can be executed on short notice and fits within the limited time available in agile environments.

C6: Artefacts
Artefacts is concerned with the technique's required input. Does the technique focus on evaluating something tangible or does it use certain documents or other tools to generate new requirements (i.e., *Questionnaires*, *Prototyping*, or *Scenarios)*? Using artefacts has proven to improve participation in requirements elicitation and provides a starting point for further idea generation [15] but is also prone to participants becoming resistant to new ideas once they become attached to a certain solution [15].

Furthermore, it relates with *Time required* as generating an artefact may also take considerable time. As such, whether or not a technique is suitable as a potential base technique relies on the time it takes to create the artefact and whether its goal is to merely evaluate the artefact or use it as a starting point for further idea generation.

C7: Project phase
The project phase characteristic focuses on what phase within the project the technique can be employed or is most effective. For instance, exploration techniques such as *Protocol Analysis, Observation*, and *Interviews* are mostly employed in the starting phases of a project where domain and problem understanding are more important to build or extend a knowledge base rather than finding novel solutions to problems. Ideally, the selected technique would be effectively employable throughout the project.

C8: Existing literature
Albeit not a direct characteristic of the technique, the available work on its goals, execution, benefits, and pitfalls is important. A technique that is popular and often used by practitioners and researched has more literature available, which is useful for understanding and building upon it. Furthermore, it has the added benefit that more professionals will be familiar with the technique, and how to execute it properly.

4.2 Technique selection

Based on the defined characteristics, the list of methods from Table 1 has been reduced to the following candidate techniques. Table 2 presents the excluded techniques with their rated characteristics as to provide insight into why they were excluded from the options.

Prototyping

Although not a group technique, prototyping is useful for improving user involvement and engagement [15], [28] ensuring that all stakeholders stay active and provide feedback as development ensues (C1). Furthermore, the ability to set the detail and focus of the prototypes ensures the granularity of the output can be controlled and new ideas can be generated from general implementation examples (C2 & C3). Additionally, prototypes can be used during any phase in a software project as feedback can be processed continuously (C7).

However, prototypes (C6) are expensive to produce both in time and cost (C5) and some knowledge about the domain is required (C4) to give useful feedback about an application's workflow and business rules other than the basic UI elements.

Scenarios

A walkthrough of scenarios can be done in a group setting, sparking discussion and new idea generation as views about the implementation of the scenario differ (C1). However, it is also possible for individuals to walk through the scenarios in their own time and discuss the results in a later session, allowing for flexibility in planning and execution. Furthermore, scenarios are a useful tool to evaluate requirements while generating new ones at the same time (C2 & C3)) and there are a number of creativity techniques that have incorporated scenario walkthroughs, such as *RESCUE* [9] and *ART-SCENE* [10].

Nonetheless, creating useful scenarios is difficult and time consuming (C5 & C6) and tend to be only useful when the system is described from the user's perspective [28], disregarding other functional system requirements that need to be covered using other techniques.

Brainstorming

Brainstorming can be done both in a group setting and individually (C1). Given that there is virtually no setup required to start a brainstorming session (C6 & C7), relatively little time is required for execution (C5) and its main goal is to think of novel and unique solutions to problems (C2 & C3). Furthermore, the concept of brainstorming is simple and can be executed with participants with all levels of domain knowledge (C4). Additionally, a lot of work on the effects and how to use it effectively has been done and published (C8).

There has been a lot of critique on brainstorming, however. Its claims about increased productivity compared to nominal groups have not held up [35], [51] and there are potential problems regarding personalities within groups that can stifle production and overall quality of the resulting requirements if the session's organization is not done properly [28].

Group work

As implied by the technique's name, the participants have to be present in a group (C1). Its setup lies mostly with the participants to think about their needs so they can articulate them accurately in the group and participate in the discussion (C4). Very useful in eliciting quality requirements efficiently (C5), compared to solo techniques such as *Interviews*, it explores every topic in-depth and records all outcomes [28] (C2). Furthermore, it is very multi-faced and can be employed during any stage in the project as its focus and desired outcome can be adjusted accordingly (C3 & C7).

Problems similar to those affecting *Brainstorming* do occur, such as the effect of strong and weak personalities within a group. Additionally, it takes a lot of time to execute due to the in-depth discussions and subsequent evaluation of all outcomes.

Even though all four techniques could serve as a base for this design cycle's development, *Brainstorming* is deemed the best option. Its light-weight nature and accessibility make it suitable for agile environments in which time is limited. This, combined with the amount of work published on brainstorming, create a good starting point upon which to build from.

Table 2 Excluded elicitation techniques

Method	**Characteristics**
Introspection	**C1:** Is performed by the analyst alone and not a team effort **C2:** Attempts to find new and creative ideas but is limited to the person performing the technique **C3:** Can be used to generate abstract and detailed requirements **C4:** Requires a lot of domain knowledge to be effective **C5:** Time required depends on the analysts' goal **C6:** Can be done without creating anything beforehand **C7:** Typically, only executed during the start of a project **C8:** Is very subjective and not a lot of existing work on how to effectively perform the technique is available
Interviews	**C1:** Although the interview is held with at least two people, the technique itself is not performed as a group **C2:** Is typically used to uncover the user's needs and understand the domain but does not usually result directly in requirements **C3:** Typically, only generates ideas that require further discussion that have the potential to become more specific as the interviewer keeps asking questions. **C4:** Requires some domain knowledge to be able to have an effective conversation with the subject **C5:** Each interview takes a lot of time to execute and analyse to come up with a set of requirements of just a single participant **C6:** Requires the interviewer to generate a set of questions or at least a list of goals to accomplish during the interview **C7:** Can be employed throughout any phase of a project **C8:** A lot of work is available on interviews but an effective interview cannot be retrieved from existing work
Questionnaires	**C1:** Questionnaires are done alone and not in a group **C2:** Results typically only answer the questions asked and do not result in additional data (i.e., requirements) **C3:** Generates ideas that are very specific, depending on the type of questions asked

	C4: The level of domain knowledge is dependent on the type of questions asked and the goal of the questionnaire **C5:** Setup may take a lot of time but total time is reduced as more and more participants fill in the questionnaire, quickly generating results without additional effort required by the analyst **C6:** Requires the analyst to create a set of questions to be asked, which need to be constructed very carefully **C7:** Can be executed throughout a project but has to be planned accordingly as it takes time to gather results **C8:** There isn't a lot of work available on using questionnaires for RE. However, there is plenty of literature to be found on using questionnaires in general
Task analysis	**C1:** Could be performed in a group but is not required for its successful execution **C2:** Is focused on understanding existing systems and extracting their requirements. Does not generate any new ones as it only describes how it is currently done **C3:** Is concerned with uncovering specified requirements in a set of tasks **C4:** A lot of domain knowledge as they have to think critically about how tasks are performed had have to understand why **C5:** Very time consuming as each task has to be understood and dissected **C6:** Doesn't necessarily require any artefacts but is useful to have during analysis **C7:** Typically, only performed at the start of a project, as a means to explore **C8:** Not a lot of work available and requires a lot of experience to be done effectively
Domain analysis	**C1:** Could potentially be performed in a group but not required **C2:** Focuses on extracting requirements from existing documents and does not focus on generating new requirements other than those found in the documents **C3:** Extracts requirements specified in documents that typically do not require additional specification **C4:** Technique primary focus on gaining domain knowledge to elicit requirements **C5:** Analysing all types of documents to find requirements can be very time consuming **C6:** Requires documentation, existing systems and other descriptions to be executed effectively. These do not have to be made by the analyst but do have to be gathered before analysis can start **C7:** Only performed at the start of a project when initial requirements are needed to explore and get the project started **C8:** Plenty of existing work available

Observation	**C1:** Results can be discussed in a group but initial observations are done alone **C2:** Describes the requirements as seen during the observation. Is not concerned with coming up with completely new ones **C3:** Results in generic requirements but can also be specific depending on how the analyst interprets what they see **C4:** Requires a thorough understanding of what is done **C5:** Very time consuming as it requires substantial observation to be able to be confident in the requirements gathered **C6:** Doesn’t require any artefacts to be made beforehand **C7:** Can be done at any point in the project but would become an evaluation technique when employed later in a project **C8:** Little work available
Protocol analysis	**C1:** Is done as a group of users and an analyst but the RE part of the technique is executed alone. **C2:** Generates requirements based on the current tasks performed by users and is not necessarily concerned with new ways of doing these tasks **C3:** Results can be abstract or specific, depending on how the users explain their tasks **C4:** Requires some domain knowledge to understand users’ reasoning **C5:** Thinking aloud is quite time consuming as not everything said is useful for the analyst but users should be able to speak freely **C6:** No artefacts have to be created beforehand **C7:** Useful at any stage of a project **C8:** Used in existing work but not heavily dissected

4.3 Technique design

At this point in the design, the base technique is chosen and the planning for the adaptation is as follows; The technique will consist of three phases: *Preparation*, *Discussion* and *Evaluation*. Its main goal is to, partly, extract the idea generation process to moments when it comes naturally, record these, and use these ideas as input for a brainstorming session. The basic flow of the technique is displayed in Figure 5.

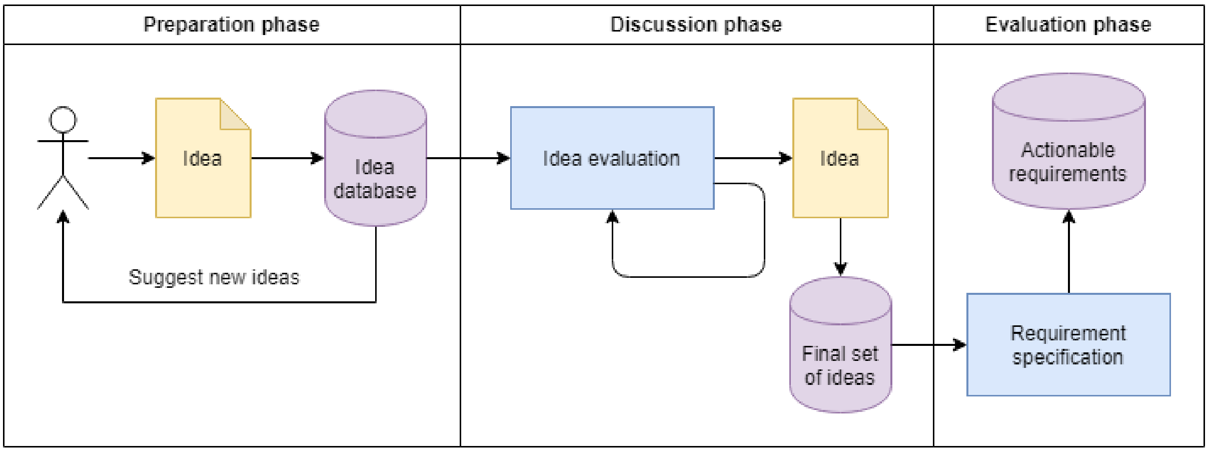


Figure 5 BrainstormPlus basic flow

Preparation phase

This phase focuses on leveraging the higher productivity in generating ideas that occurs when done individually compared to in a group setting [51] and is executed continuously during other, day-to-day activities and tasks. Ideas can occur at any time as you are working through annoyances during current tasks, limited capabilities of current software or other observations during the day. These 'aha' moments need to be captured on the spot and should not be left until an elicitation session, as they can, and probably will be, forgotten about [44], [52].

An application (mobile, desktop, web, etc.) will be built, allowing stakeholders to record ideas as they occur. This has to be done as unobtrusive, quick and easy as possible to ensure that it does not negatively affect their productivity during their day-to-day work. Additionally, users can request a suggestion from the application that serves as a creativity trigger for idea generation. These suggestions will be generated server-side by integrating Wever's work on synthesizing requirements and feature extraction from similar applications through app reviews [16]. Users may also receive push notifications asking for new ideas to stimulate participation but its complications regarding productivity have to be thought out further. Figure 6 shows a mock-up of the envisioned application.

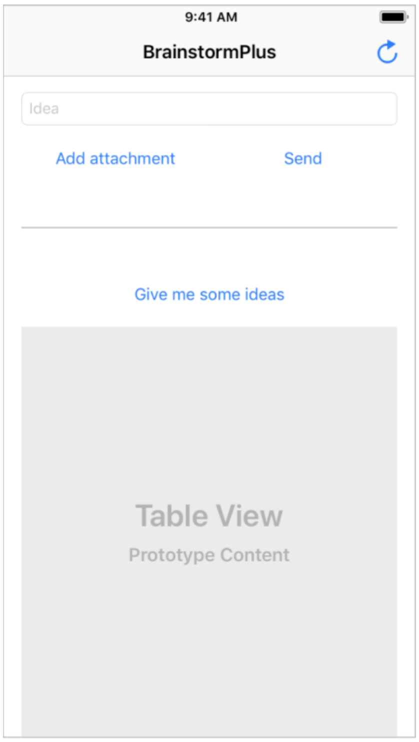


Figure 6 BrainstormPlus mock-up

Discussion phase

The discussion phase uses the ideas generated in the preparation phase as input for a brainstorming session. This session consists of a 'bottom-up' approach where the group divides into smaller groups in which the ideas would be discussed and evaluated. This discussion may result in additional idea generation through combinational creativity [12] and would be recorded and taken to the next step in the brainstorming session. This step will consist of combining groups to create bigger ones, repeating the discussion process and is repeated until the entire group is reformed and the generated ideas can be discussed.

Evaluation phase
The evaluation phase will consist of specifying the generated ideas from the discussion phase onto workable requirements as it is expected that the generated ideas will not necessarily result into a set of actionable requirements such as user stories. This will be done by the requirements analyst(s) that would lead the discussion phase.

4.4 Evaluation

Although the technique currently only has the initial backbone developed, some early evaluation already shows that this implementation might not be feasible for this research. The different phases of the technique make it quite extensive. Each phase requires several steps that need to be evaluated by separating them from the other phases to mitigate their effect on each other. Afterwards, the technique needs to be evaluated as a whole during several projects or iterations to see whether or not it actually works as intended.

Although interesting from a research and method development perspective, within the context of this thesis this would require more time than available to properly evaluate it. As a result, this design is deemed not feasible and a simpler approach will be sought after that fits within this thesis's scope and time limit. This is not to say that the technique could not be feasible, just that it does not fit this research.

5. Design cycle: Feature.Ly

Taking into account the limitations identified in the evaluation of the previous design cycle, this cycle will look into a technique simple enough to be properly evaluated within this research's scope and time span, while still being data-driven, lightweight, easy to implement, and suitable for agile environments.

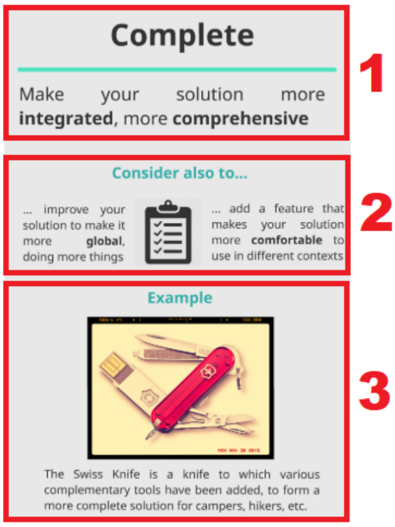
5.1 Technique selection

In order to reduce the number of variables that would have to be accounted for during evaluation, selecting an existing creativity support technique that has been previously evaluated will be most feasible to be adapted. The candidate techniques are those that have been previously mentioned in Chapter 3.4: *RESCUE* [9], *ART-SCENE* [10], *iThink* [11], *EPMcreate* [41], and *Creativity triggers* [42].

The *Creativity trigger* technique by Burnay, Horkoff, and Maiden [42] will fit best as the other techniques are either workshops themselves, requiring a lot of time to be prepared for and executed, or tools to be used during or after a workshop to plan for the next session. This would result in the same problem regarding evaluation as during the previous design cycle in which time constraints would not allow for a proper experiment, and thus evaluation, of the technique. The *Creativity trigger* technique, however, is a tool that can be used at any time and is designed to directly result into new ideas.

5.2 Technique planning

Initial evaluation of the *Creativity Trigger* cards shows that, although very descriptive, they are also static. Figure 7 displays the different elements of such a card. The title and subtitle (element 1) are quite descriptive and provide some keywords that can be used to identify the type of ideas that are sought after when using this card. The guidelines (element 2) provide help in thinking of new ideas. The example (element 3) gives implementations of the previous two elements as a way to relate them to your own situation. However, they are limited to a single type of product so it is less useful in other contexts such as RE. Additionally, the examples do not cover software products. Furthermore, they do not provide the user with additional examples if the one given does not 'trigger' them into thinking of creative ideas. This opens up the opportunity to adapt the trigger cards to make them more dynamic and applicable to a RE team's current focus area.


Complete

Make your solution more **integrated**, more **comprehensive**

Consider also to...

... improve your solution to make it more **global**, doing more things

... add a feature that makes your solution more **comfortable** to use in different contexts

Example

The Swiss Knife is a knife to which various complementary tools have been added, to form a more complete solution for campers, hikers, etc.

Figure 7 Elements of a Creativity Trigger card

Given this research's focus on creating a data-driven approach to aiding creativity in RE during software projects, the choice was made to use app reviews to aid users in thinking of creative ideas by extracting features and presenting these to the user. This stimulates the *Combinational* method of creativity which would be most suitable when using data as the other methods rely on exploring new spaces or concepts that are more abstract, and thus harder to encapsulate in a software tool [12]. App reviews are chosen instead of other means of feature extraction as they often mention features and characteristics of apps that are not included in any other description and keep up with new updates as app users review these updates as soon as they are released.

5.3 Dataset

Before the application could be build, the dataset has to be created so the feasibility of the technique can be assessed before evaluation using an experiment.

Feature extraction

The app reviews are retrieved from the App Store, using a scraper library[1] that allows for the selection of only the 'most useful' applications as filtered by the App Store to exclude reviews that are limited to only a few words or are no longer relevant as they cover an old update. A similar library is available for the Google Play store[2] and was intended to provide a mix of the reviews from the two main application stores. This library became unusable, however, as the Google Play store changed its APIs during the prototype development after which such scraper libraries can no longer retrieve app reviews.

The reviews were taken from 10 of the most popular applications on the App Store: Spotify, Facebook, Instagram, Facebook Messenger, Netflix, Snapchat, Uber, WhatsApp, YouTube, and Gmail. For each of these applications, 50 reviews were retrieved using the App Store scraper.

[1] https://github.com/facundoolano/app-store-scraper

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

A manual analysis was carried out to determine the usefulness of the reviews with a lower rating, as it seemed that reviews with a lower score generally did not mention any features or were incoherent to such an extent that not even a human could decipher the meaning behind the review. This pilot consisted of 125 reviews, all a small subset from the earlier retrieved set. Some examples of useful and less useful app reviews are displayed in Table 3. In this table, features are highlighted in green.

A feature was defined as a sentence that clearly describes functionality of an application from which the expected behaviour of the application could be understood relatively well. In each review, the features mentioned were tagged, counted, and scored based on how clearly it was described. Looking at the number of features and quality of those features resulted in the decision to filter out any reviews that had a rating below 4. This resulted in a set of 187 reviews.

Table 3 App review usefulness examples

App review	**Rating**	**App name**	**# of features**
I have been a loyal premium subscriber for around 7 Years. And I've most the of the time enjoyed the updates. But let me tell you I absolutely hate this one. And it's come to the point if you don't fix this or change something quick I'm going to unsubscribe from premium and delete the app. The biggest concern I have is with recently listened to music. The way you had the UI before was perfect to me. **I loved the way I could see all of my previously listened to artists on one page and scroll down**. I could edit this as well. Now the new way to listen to them is trash. I can click on the artist and listen to their entire library, but I don't wanna do that. **I wanna hear the library I had in the last updates of just the songs I saved from the artist**. So now I have to go the my music tab and search for the artist and listen to them that way. Which is extremely inconvenient and out of the way. It wasn't broken. Why did you change it? Is there any way I could undo this latest update and get the previous one I enjoyed? If not I'm sorry this is such a small thing but it really made a huge difference to me. Like I said if it isn't fixed I'm gonna delete the app, and I'm sure many others feel the same way I do.	5	Spotify	2
All I can say is this app is positively amazing. It is made very nicely, and although it does have an occasional advertisement, that does not possibly amount to the overall amazing quality. Definitely worth getting the premium membership feature (I don't own it, but have used it before with a close friend of mine.) This app is a must-have if you have a busy lifestyle and love listening to music! One of my favorite features is the playlist option. You can make **customized playlists to your music preference**. It's positively amazing! You can also **add certain artists you like**, and it customizes your daily mixes. Another great feature is the daily mixes! The app gives you **daily playlists based on your favorite songs and artists**, and **makes a personal playlist based on the aforementioned favorite songs and artists**. Altogether, it is a spectacular app that has MANY amazing features, and I'm not surprised it is so popular and successful. I hope everyone can notice the amazing quality and effort that was put into the making of this AMAZING app!	5	Spotify	4
I think the idea of the app was spot on. I think the execution was terrible. The app is up to date and on top of these songs but it is a pain to work with. An app should always be user friendly and easy to understand but this isn't it. I don't like too much music and don't know that many singers so when it made me choose three that's a big no, it messed up my recommendations and I'm stuck with sicko mode as a recommendation because I was joking around. The	3	Spotify	0

app doesn’t allow you to play just one song, you have to make a playlist and if it’s not big enough it plays “recommended” music which is extremely irritating. Skipping songs only 6 times per hour is ridiculous and only there to make you pay money. Also the fact that you can’t restart your playlist after you’re done and it starts playing other music is uncalled for. I tried everything from looking it up and restarting my phone. Nothing worked. Ik there are hard core fans who know all the ins and outs of this app but that’s not the point, anyone (within reason) should be able to use and app, if they can’t then there is no point to download this app. Lastly I forgot to mention but not being able to add whole albums is also irritating, adding song by song is meticulous.			
The only reason I still have facebook is because my school posts important updates at times. I get notifications constantly from facebook telling me so and so posted a picture or so and so commented on a post...things like that. These are usually people I don’t even interact with and I do not want these notifications. The only way to stop getting them is to completely unfollow the person, which then takes their posts out of my news feed as well, which I do not want to happen. And even when I unfollow people my phone continues to give me notifications of new people posting things that I don’t care about. The only reason I don’t turn off Facebook notifications all together is, like I said, so I can get important updates from my school’s groups. I’ve turned off every notification setting within the Facebook app that I can and continue to get these notifications on my phone. It is extremely annoying. I will also get badge app notifications saying I have 4 new notifications when I don’t actually have any. Extremely annoying and makes me think I have an important notification to check when I rarely actually do. Will delete my account as soon as I’m done with school if this isn’t fixed.	1	Facebook	0
I love Instagram! It's done a lot to help my career. But whatever new algorithm replaced the old stinks. I am REALLY tired of seeing the same 15 people's posts over and over!!! I follow, like, 400 people! I follow a lot of other artists for inspiration, community and connection, but for some reason, I get a constant stream of my cousin's rabbits and my client's lunches (I can't exactly unfollow my client or my cousin). It's particularly frustrating when a post for a relevant event pops up and I realize it happened 3 days ago. Sure, I may not have seen it anyway, but if I'm gonna see it, I'd like to know I can go! And what does this mean for my art?! Am I not reaching people who are ALSO getting off Instagram 30 seconds after they get on out of frustration that they are STILL seeing their cousin's rabbits?! It would be REALLY cool if we at least had the option to view our feed the old way. Wouldn't that be an interesting experiment? If that's too much, how about a trial run? Give people the option of viewing chronologically or the new way. Majority wins! All I REALLY care about is that Instagram does not go the way of Facebook. I BARELY use FB anymore- it's too much! It feels like high school when the girl next to me in class kept gossiping in my ear while I'm trying to read "Bobby commented on Tara's comment on Devin's photo of Mike and Tina". Don't caaaaaaaare! Anyway, end rant!	3	Instagram	0
Instagram is easy to use and keeps getting better! **Tag people in posts**, great working live streams +filters, all types of posts from all your friends (Challenges, Updates, Memories, etc.), messaging through Instagrams in app direct messages +groups, filters for pictures you want to post, and for the over protective parents if you will, you have the ability to **restrict anyone without permission from user from seeing posts**, and not to mention the filters on post, on ones profile you can **add links to personal websites**, **write some information about yourself**, or just update your followers! A favourite feature of mine is **public commenting directly on ones post**, and if any haters post something offensive, you can **delete the comment as soon as**	5	Instagram	8

it's up, easy as that! As well as most social media apps and websites, you can **block and/or report other users** who break Instagrams community guidelines. Not to mention the fact that no matter how old or how popular the post, no matter who it's been sent to, you can always take it down and removed from anyone ever to see again, there is absolutely no limit on how many posts one can have, and there is always a **function to translate other languages in to the users selected language**			

For each review, the features were manually tagged and extracted. Manual tagging and extraction were chosen to reduce the number of variables that could influence the technique evaluation. For instance, someone deeming the application not useful purely because the examples that were presented were incoherent and hard to understand. Manual extraction would ensure that features are formulated in such a way that they make coherent sentences. After manual extraction, features that occurred more than once per application were removed, resulting in a final set of 197 features.

Feature mapping

Since the features would be included on the creativity trigger cards, they would have to be mapped to one of these cards to ensure that the example fits within the card's guidelines. The guide used for mapping features to one of these trigger cards can be found in Table 4. It has to be noted that this process is subjective; however, this guide provides insight in the reasoning behind the mappings. The triggers were directly taken from the paper written by Burnay, Horkhoff and Maiden [42].

Table 4 Feature mapping guide

Trigger	**Mapping**
Entertaining	This category was used whenever a feature was purely implemented to make it more fun to engage with others, somehow made the app more appealing, or introduce users to content they would normally not interact with. It was only mapped if the feature was not a core feature of the application (Such as sharing photos in Instagram or sending messages in WhatsApp).
Light	The Light category was mapped when a feature existed to either take over a task a user would normally do manually or when it made using the app a lot easier and more efficient, thus reducing the time it takes to accomplish a task in the app. Anything that would 'help' the user by doing things for them that they would want to do anyway.
Adaptable	Adaptable would be mapped if a feature existed so the user could configure something to their liking, make it more personal, or otherwise edit certain settings so they would feel more comfortable using it.
Economical	This would be used whenever the feature allowed for the saving of any costs, usually monetary.
Complete	Complete would be used when a feature described a core feature or other feature that just makes sense given something else or is a feature that supports another (Such as viewing a list of favourite artists, whereas the feature of being able to favourite an artist would be Adaptable).

Durable	Durable would be mapped anytime a feature would ensure the app's reliability, security, privacy and other qualities that ensure its user base will keep using the application without any issues that can be retraced to poor programming, etc.

Not all extracted features were included in the corresponding application. Many reviews mentioned features that the users would like to see in the future. As these features cannot be used to directly relate them to the corresponding application, as there is no relation, a second mapping was assigned, called 'Not in app'. This allowed them to still be used as a secondary example that suggests a new feature instead of one already implemented in an existing application. As the name of the application could not be used, due to the feature not being part of that application, its name was replaced with a type of application such as: *Image sharing app*, *Instant messaging app*, *Music player*, *Ride sharing app*, *Video streaming app*.

Table 5 Number of mappings per trigger

Trigger	**No. of mappings**	**No of mappings with secondary 'Not in app'**
Entertaining	10	10
Light	22	13
Adaptable	35	24
Economical	6	0
Complete	39	21
Durable	9	7

After mapping all the variables, the number of features mapped to each varied widely, as can be seen in Table 5. For instance, *Economical* has only 6 mapped features in total. Consequently, these 6 often mentioned the same feature that was displayed in different applications, such as 'the ability to use Wi-Fi or mobile data to avoid network fees.', offering no variety in the type of examples produced. The *Durable* trigger had the same lack in variety, even though it had more mapped features. As such, *Economical* and *Durable* will be excluded from the dataset and will not be used as trigger options in the application. Arguably, the *Entertaining* trigger also has a small number of mapped features, but these are different to such an extent that they will provide the necessary variety in examples.

The final reduced dataset of the extraction and subsequent mapping can be found in Appendix A.

5.4 Prototype

The prototype application, called *Feature.Ly*, is build using the *react* library, supported with the material-UI react framework[3] to easily build a UI that also supports mobile devices. The code written for the application is placed in a public repository on Bitbucket[4] and, as of writing, the application can be found at *http://featurely.jaspervannoordenburg.nl/*. A screenshot of *Feature.Ly* is displayed in Figure 8.

[3] https://material-ui.com/

[4] https://bitbucket.org/jvnoordenburg/feature.ly/src/master/

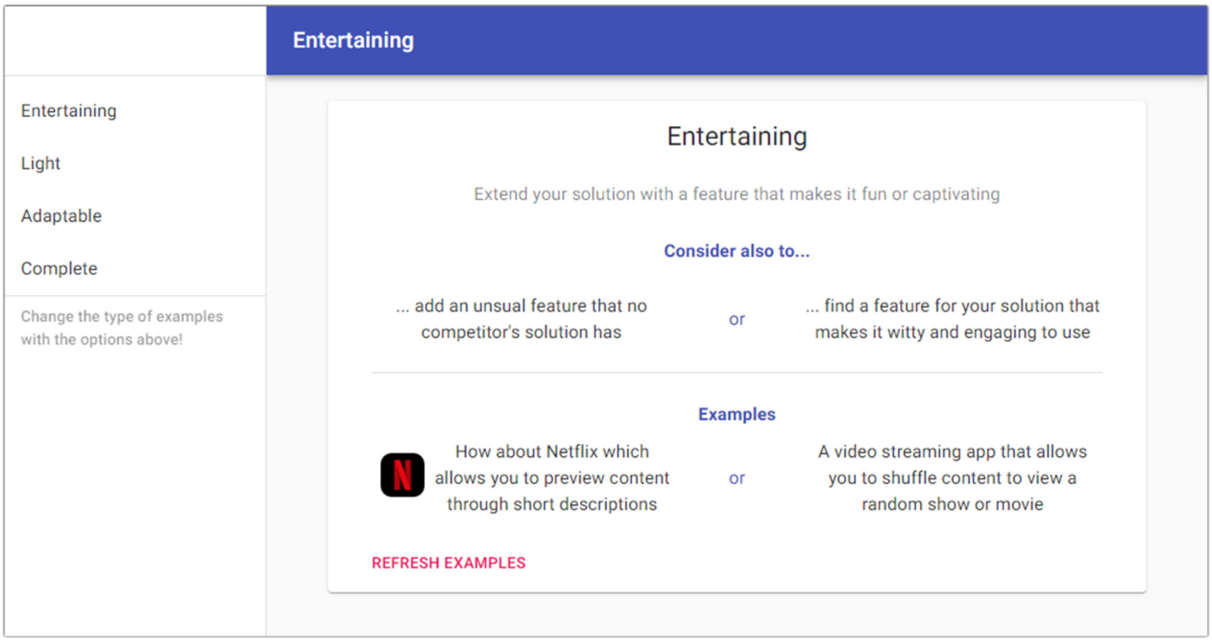
The left navigation bar shows the different types of triggers that are available within the application and can be used to change the trigger card currently being displayed. The top part of the card itself is identical to the trigger cards created by Burnay, Horkoff, and Maiden [42], albeit styled differently. The examples are created by selecting two random features mapped to the current trigger card; one associated with an application and one that had the secondary tag of 'Not in app' as explained previously. The example associated with an app was constructed using the following standardized sentence:

How about <AppName> which allows you to <Feature>

The example not associated with an existing app was constructed using:

A <AppType> that allows you to <Feature>

Clicking the 'Refresh examples' button will randomize the features and create two new examples in case the user does not feel the examples fit their current train of thought or triggers them in any way.



Entertaining

Entertaining

Light

Adaptable

Complete

Change the type of examples with the options above!

Entertaining

Extend your solution with a feature that makes it fun or captivating

Consider also to...

... add an unsual feature that no competitor's solution has

or

... find a feature for your solution that makes it witty and engaging to use

Examples

How about Netflix which allows you to preview content through short descriptions

or

A video streaming app that allows you to shuffle content to view a random show or movie

REFRESH EXAMPLES

Figure 8 Feature.Ly screenshot

5.5 Evaluation

Overall, *Feature.Ly* is a much more accessible and light-weight technique than the entire *BrainstormPlus* approach. It is data driven by using app reviews directly retrieved from the App Store. It is lightweight and can be used at any time and supports different context due to the different types of applications, making it very suitable for agile environments, and keeping in line with this research's goals.

There are some limitations, however. For instance, the features had to be manually extracted to ensure the resulting features are coherent and usable in the trigger cards. Furthermore, the mapping of the features had to be done manually as well, as making this automated would be very difficult as of writing due to the very subjective nature of the feature mapping which cannot be easily copied into Natural Language Processing (NLP) programs. Due to these manual requirements, the number of features available in the application is limited, reducing the application’s potential for fostering divergence in creative thinking.

Ideally, the extraction process would be automated to provide a larger dataset and support more applications to create examples from. Nevertheless, the application as of now is usable and can be tested for its feasibility. The number of variables acting upon the technique has been greatly reduced and testing this prototype will take a lot less time than the previous technique would have taken.

6. Experiment design

To evaluate the technique developed in the previous chapter and determine its effect on users' creativity, an experiment has to be designed. First, the type of experiment will be decided upon, after which the experiment itself will be mapped out.

6.1 Experiment type

Three different types of experiment are considered, each with their own pros and cons; *Expert review*, *Card Creativity Questionnaire*, and a *Scenario Based Questionnaire*.

Expert review

This type of experiment consists of approaching a number of RE experts and have them review the cards to determine their feasibility and possible effect in a development team. The experts will interact with the application by walking through a number of scenarios and then be asked about their opinion about the concept behind the application and its implementation. This is done through a semi-structured interview held with each expert separately.

A benefit of this type of experiment is the access to professionals experienced with different types of techniques Their experience with similar tools can potentially provide insight in the application's feasibility and can provide quick feedback for improvements. Additionally, qualitative evaluation provides a deeper understanding in what does and doesn't work for a technique. Given that the application is a prototype, and probably needs further development.

There are, however, also disadvantages to using expert reviews. For instance, given the time it takes to approach experts and set up an interview, likely only a few experts could be interviewed. This reduces the generalizability of the results and, thus, makes them less reliable. Furthermore, experts only given an estimation. Their opinion, albeit one with a lot of experience behind it, is no evidence that the tool aids in helping people be more creative or not in a real requirements elicitation setting.

Card Creativity Questionnaire

This questionnaire consists of presenting participants with the application and ask them to answer a number of questions for each card they generate. These questions cover their perception of the feasibility and usability of the application, the quality of the examples, and whether or not they perceive these examples to be supporting creativity.

This technique allows for a large number of people that can be reached with various background. It also gives a good indication of user's enthusiasm towards working with the application.

The main disadvantage of this technique is that it gives no evidence regarding the technique's effectiveness. Just the perception of a group of participants.

Scenario Based Questionnaire

The final type of experiment asks participants to generate ideas within a certain category (Entertaining, Adaptable, Light, Complete) through various scenarios to simulate an actual elicitation session. The scenarios describe an application that is looking for ways to improve. It is then up to the participant to come up with new ideas by using the application. After each scenario, they will be asked to answer a small set of questions about the scenario, the quality of their ideas and whether or not the cards helped them.

This type of experiment simulates a real-world example of how the technique could be used and evaluates the entire technique. Furthermore, it can be performed individually so a large number of participants can be reached compared to experiments where interviews are used.

A potential issue with this type of technique is the possibility that participants do not understand the application and thus their ideas and perception of the whole method will be influenced. Another is the quality of the scenarios. These need to be thought out well and be easily understood by almost everyone as participants with different background may not understand the needs of every type of application and thus have trouble generating ideas. Finally, there is no way of determining the quality of the ideas. The participants will rate their own ideas but that is just their perception.

Given it has the closest resemblance to a real-world example of how the technique would be used, the *Scenario-based Questionnaire* will be selected as the type of experiment. The disadvantages can be overcome by carefully constructing the scenarios and provide pointers during the questionnaire. Additionally, an element of expert review will be included to overcome the lack of idea assessment by someone other than the participant. The ideas generated will be assessed by experts to determine their quality.

6.2 Phase one – Scenario-based Questionnaire

The first phase of the experiment will be conducted as follows; The participants will be asked a number of questions regarding their demographic characteristics to be used in the analysis of the results. These can potentially help explain certain discrepancies or general conclusions.

After the demographic characteristics have been filled in, the participants will continue to the first scenario. The first scenario will present them with a problem statement and they are prompted to come up with three ideas to help solve this problem. A questionnaire is filled in afterwards in which the participants are asked to evaluate the ideas they just came up with. These two steps, the generation of ideas and subsequent evaluation, are then repeated for another problem statement within the same scenario.

They then proceed to the second scenario. Again, they will be presented with a problem statement and following evaluation questionnaire two times. This scenario, however, they are able to use *Feature.Ly* to help them in the generation of ideas for both problem statements.

After the second scenario is completed, the participants will be asked a couple of questions regarding their experience with *Feature.Ly* and their perception if it helped them in generating the ideas.

The flow of the first phase of the experiment is visualized in Figure 9. The participants will be randomly assigned to one of two groups, group A and group B respectively. Each group will do a different scenario with and without the use of *Feature.Ly*. To ensure users are not influence by exposure to the application when generating ideas without *Feature.Ly*, the scenario that does not allow for use of the application is always done first. This crossover design is displayed in Figure 10.

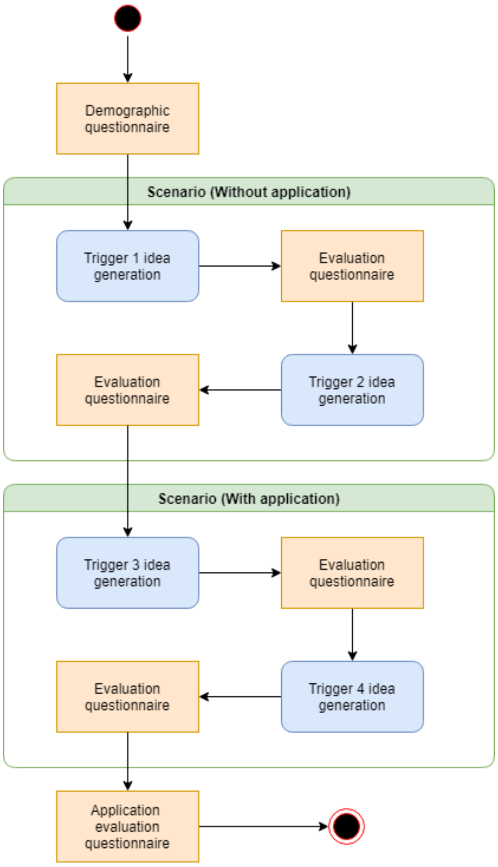

Demographic questionnaire
Scenario (Without application)
Trigger 1 idea generation
Evaluation questionnaire
Trigger 2 idea generation
Evaluation questionnaire
Scenario (With application)
Trigger 3 idea generation
Evaluation questionnaire
Trigger 4 idea generation
Evaluation questionnaire
Application evaluation questionnaire

Figure 9 First phase flow diagram

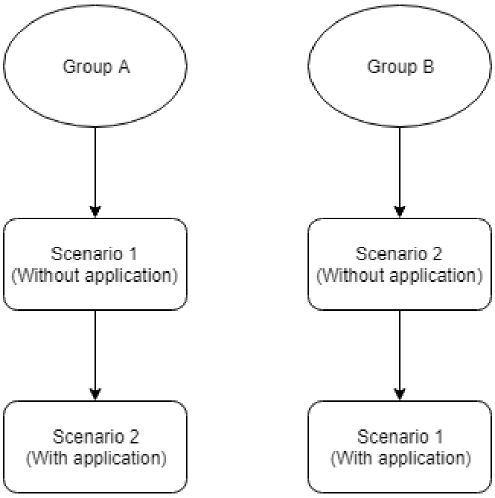

Group A
Group B
Scenario 1 (Without application)
Scenario 2 (Without application)
Scenario 2 (With application)
Scenario 1 (With application)

Figure 10 Experiment crossover design

Demographic questionnaire

The demographic questionnaire will serve only to interpret the retrieved data and relate them to the demographic to explain results that are unexpected or need further clarification. The questions contained in Table 6 will be included.

Table 6 Demographic questionnaire questions

Question	Options	Explanation
What is your age?	18-24, 25-34, 35-44, 45-54, 55-64, 65+	This question allows for the comparison of responses between age groups. For instance, this could determine if the app should be more user friendly for less 'tech-savy' people or the apps used for examples should be broader as they do not know the applications (e.g., SnapChat is primarily used by younger people)
What is the highest degree or level of school you have completed? If currently enrolled, select the highest degree received	High school or equivalent, Bachelor's degree, Master's degree, Doctorate	This question can help give insight in whether or not the sample of participants is an accurate representation of who the application is created for.
What is your experience with brainstorming?	I have never done any brainstorming before, I have participated in a brainstorming session before but not often, I regularly participate in brainstorming sessions	The idea behind this question is experience in coming up with new (creative) ideas. Perhaps, people that often do brainstorming may find the application less useful as they can do it themselves. Or maybe they feel like it is extremely useful as they often revert to what they already know and find the examples refreshing. It can help determine if the application can be used in a requirements elicitation setting (i.e. if the people with lots of experience negatively respond to the effect of the application, it may not be feasible for that purpose)
What is your experience with reading app reviews?	I have never read an app review before, I occasionally read app reviews before downloading an application, I always check the	These questions are meant as a way to see how many people actually use app reviews and possibly use that as a means to explain the usability of reviews in future work. Additionally, it can be expected that users experienced with app reviews are

	app reviews before downloading an application	more eager in coming up with new ideas as they are familiar with suggesting new features.
What is your experience with writing app reviews?	I have never written an app review before, I occasionally write app reviews for apps I use, I regularly write app reviews for the apps I use	

Scenarios

As mentioned before, the scenarios have to be carefully constructed to ensure they are easily understood by the participants. As a result, the scenarios will be based on two existing applications, *Slack* and *Udemy*; A business-oriented communication tool and an online learning platform. The applications' core functionalities can be easily explained as they are relatively simple. As a result, participants can easily relate to them, but they are not that well known that they can just copy feature based on their knowledge of the application. Participants have to actually think of ways to improve the application. As a result, the following two scenarios are constructed:

Scenario 1 - Communify

> *Your organization develops an application, called Communify, in which people can quickly and easily communicate with each other within the company. Current functionality includes the ability to directly message colleagues and create group message conversations for teams to collaborate. It is also possible to post announcements, either within a single team, or for the entire company to see. At the time, the application is limited to these features.*

Scenario 2 - Edusphere

> *Your organization has created an online learning platform, called Edusphere, on which users can follow a wide range of video-based courses created by other users to learn skills and explore their interests. Each course is divided into lectures, each consisting of their own video. The user purchases a course and is then able to watch it on demand. The creator of the course has the ability to add tests at the end of the course or during it to allow the user to test their knowledge throughout the course. It is also possible to communicate with the course's creator in case the user has a problem or question regarding the content.*

Problem statements

For each trigger included in the dataset (see Section 5.3), a problem statement will be created to encourage participants to think of a certain type of feature an use the corresponding trigger card in the *Feature.Ly* application. The participants will not be told which trigger card is assigned to which problem statement as to encourage them to use the entire application. During construction, the descriptions and guidelines of the trigger cards will be used as a starting point for the creation of the problem statements. The following problem statements are thought of:

Scenario 1 – Communify

Complete trigger

> *Your team is tasked with expanding the product's features and provide employees with more support during their work day.*
>
> *Think of (at least) 3 new ideas that could be useful for the application.*

Light trigger

> *The application's users are complaining that the list of conversations is too cluttered and it is hard to find a single message. They also mention that it is difficult to distinguish between conversations and announcements, making the entire application difficult and too time-consuming to use for what service it provides.*
>
> *Think of (at least) 3 ideas that could make the application easier to use.*

Scenario 2 – Edusphere

Adaptable trigger

> *In order to allow users to learn at their own pace, your team is asked to come up with ideas to have a learning experience that users can control and configure to be as flexible as possible, while remaining effective.*
>
> *Think of (at least) 3 ideas that could help them achieve this goal.*

Entertaining trigger

> *After a survey among users, it was concluded that users find it difficult to discover other courses that interest them and usually do not keep using the platform to learn other skills after their initial sign up. Your team is asked to come up with a solution to help retain users.*
>
> *Think of (at least) 3 new features that could help solve this problem.*

Idea evaluation questionnaire

As mentioned before, after each idea generation round, there will be a questionnaire evaluating the ideas generated. The objectives of this questionnaire are to determine the participants' confidence in the feasibility of the ideas, their own perception of the creativity of the ideas, and their perception of how easily the ideas were generated.

All questions used a Likert scale to make comparison between groups and triggers easy, as well as limiting the time it takes to complete the experiment, as there are already a large number of steps involved. Each question uses a 7-point Likert scale instead of the more traditional 5-point scale as to offer the participant more accuracy in rating their ideas and find an answer they are content with [53]. The questions included in Table 7 will be incorporated.

Table 7 Idea evaluation questionnaire questions

Statement	**Scoring**	**Reasoning**
I understood the scenario and what type of ideas were asked of me	*1-7 Likert scale*	Determines the participant's understanding of the scenario and could help explain why they had difficulty creating ideas if that happens
The ideas I generated are addressing the problem stated by the question	*1-7 Likert scale*	Determines the participant's perception of whether or not the ideas they were able to come up with are actually within the boundaries of what was asked of them and are solving, or at least addressing the stated problem
The ideas I generated are clear, i.e. can be easily understood by others	*1-7 Likert scale*	Determines the confidence of the participant in how easily their ideas can improve the application.
The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before	*1-7 Likert scale*	One of the elements of what creative ideas are by Boden [38].
The ideas I generated can increase the "value" of the app	*1-7 Likert scale*	One of the elements of what creative ideas are by Boden [38].
The ideas I generated were easy to come up with	*1-7 Likert scale*	Perhaps Feature.Ly does not result in more creative ideas but helps in how easily ideas are generated and thus can still be of value.
I could have easily provided additional ideas	*1-7 Likert scale*	Relates to the participant's confidence in the generation of more ideas. Perhaps the entered ideas aren't the most surprising or useful, but if they can generate more ideas there is more to discuss and lead to ideas that are of value to the application.

Application evaluation questionnaire

This questionnaire's goal is to retrieve participant's opinion about the application and their perception of how it aided them in generating the ideas. The results will be used to compare them against the result of the between group comparison of the idea evaluation questionnaires to see if the participant's perception matches the objective results. The questions will be similar to the idea evaluation questionnaire but more general and geared towards Feature.Ly. Additionally, an optional field will be added in which participants can give an elaboration of their experience with the application. The questionnaires for group A and B can be found in Appendix B and Appendix C, respectively.

Table 8 Application evaluation questionnaire questions

Statement	**Scoring**
By using Feature.ly, I felt more confident in the ideas I came up with	*1-7 Likert scale*
By using Feature.ly, I felt my ideas were more surprising than without using it	*1-7 Likert scale*
By using Feature.ly, I felt my ideas were more valuable than without using it	*1-7 Likert scale*
By using Feature.ly, I felt it was easier to come up with ideas than without using it	*1-7 Likert scale*

6.3 Phase two – Expert review

The second phase of the experiment will be done through a review of the ideas generated by the participants of the first phase. The goal of this phase is to see if ideas generated with the application are actually more creative than ideas generated without and compare this to the participants own perception of creativity. As such, the questions asked for each idea will be partly similar to the questions asked on the idea evaluation questionnaire mentioned previously. The ideas will be randomized before sending them to the experts to ensure that they do not know which ideas are generated using the application and which are generated without. For each idea, the following questions will be asked.

Table 9 Expert review questions

Statement	Scoring	Reasoning
The idea is addressing the problem stated by the question	*1-7 Likert scale*	Copied from the idea evaluation questionnaire
The idea is surprising, i.e., not seen implemented in this type of app before	*1-7 Likert scale*	Copied from the idea evaluation questionnaire
The idea can increase the "value" of the app	*1-7 Likert scale*	Copied from the idea evaluation questionnaire

7. Results

In this chapter, we present and analyse the results to determine the effect of using *Feature.Ly* when generating ideas for a software application. We first present the demographics of the participants. Afterwards we cover the results of phase one and phase two.

7.1 Participants demographic

In total, 18 respondents participated in the survey, which were equally divided between group A (N=9) and group B (N=9). Of these participants, the majority consisted of young people between the ages 18 and 35 with either a Bachelor's or Master's degree, as can be seen in Figure 11 and Figure 12.

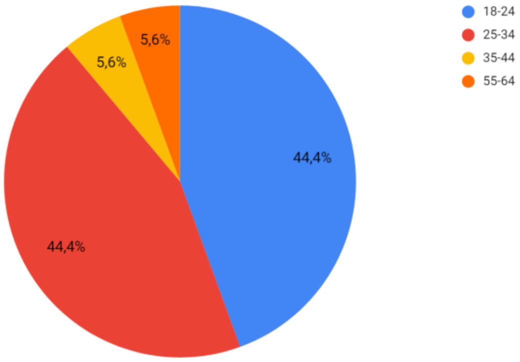


Figure 11 Participants age distribution

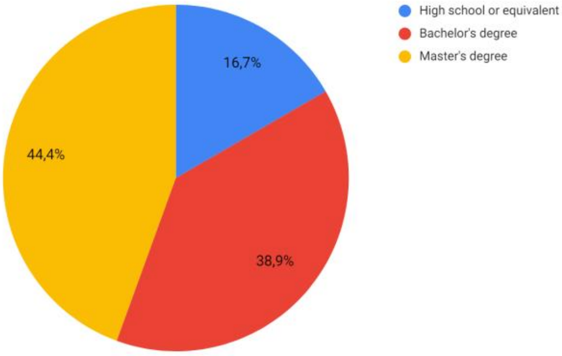


Figure 12 Participants education level distribution

Of the participants, half had never participated in a brainstorming session, wheras the other half was equally divided between occasionally and regularly participating in such sessions. Furthermore, only 3 participants had never read an app review before, whereas the majority of participants, 10 out of 18, occasionally read app reviews. Writing app reviews, however, was done by only 1 participant regularly, wheras 15 out of 18 participants had never written an app review before.

7.2 Idea generation – Summary

The results regarding the idea generation section of the survey did not always yield 9 sets of data per group. Some participants did not fill in any ideas for certain problem statements. As such, these results were removed from the final set of data. Each trigger, and accompanying dataset, was analysed separately as they cannot be combined due to the different nature of each problem statement and the type of ideas they require.

Complete trigger

The first problem statement's results, regarding the *Complete* trigger, are presented in Table 10. The distribution of the scores given by the participants is visualized in Figure 13. In general, the use of *Feature.Ly* resulted in a higher score regarding each question, except for three; The participant's perceived value of the ideas, the ease of coming up with the ideas, and whether or not they understood the scenario, which was equal between groups. Participants scored their understanding of the scenario exactly the same in both groups on average. Furthermore, when participants were asked if their ideas would make *Communify* more valuable to use, the group using *Feature.Ly* (UsingAid = TRUE) rated their ideas less valuable. Similarly, the group using *Feature.Ly* felt the ideas were not as easy to come up with than the group that didn't use the application. Finally, the scores given by the group using *Feature.Ly* were more tightly packed, i.e., had a lower standard deviation, as well as a lower range of scores compared to the group that didn't use the application.

Table 10 Complete trigger results

	Using Aid	**Mean**	**Std. Deviation**	**Min**	**Max**
Understood_Scenario	FALSE	6.11	1.05	4.00	7.00
	TRUE	6.11	0.78	5.00	7.00
Ideas_AdressingProblem	FALSE	4.67	1.66	2.00	7.00
	TRUE	5.78	0.67	5.00	7.00
Ideas_Clear	FALSE	5.22	1.56	2.00	7.00
	TRUE	5.56	0.88	4.00	7.00
Ideas_Surprising	FALSE	2.78	1.56	1.00	5.00
	TRUE	3.78	0.97	3.00	6.00
Ideas_Value	FALSE	5.89	1.27	3.00	7.00
	TRUE	5.56	0.73	4.00	6.00
Ideas_Easy	FALSE	5.11	1.45	3.00	7.00
	TRUE	4.89	0.78	4.00	6.00
Ideas_Additional	FALSE	5.11	1.69	3.00	7.00
	TRUE	5.67	0.71	5.00	7.00

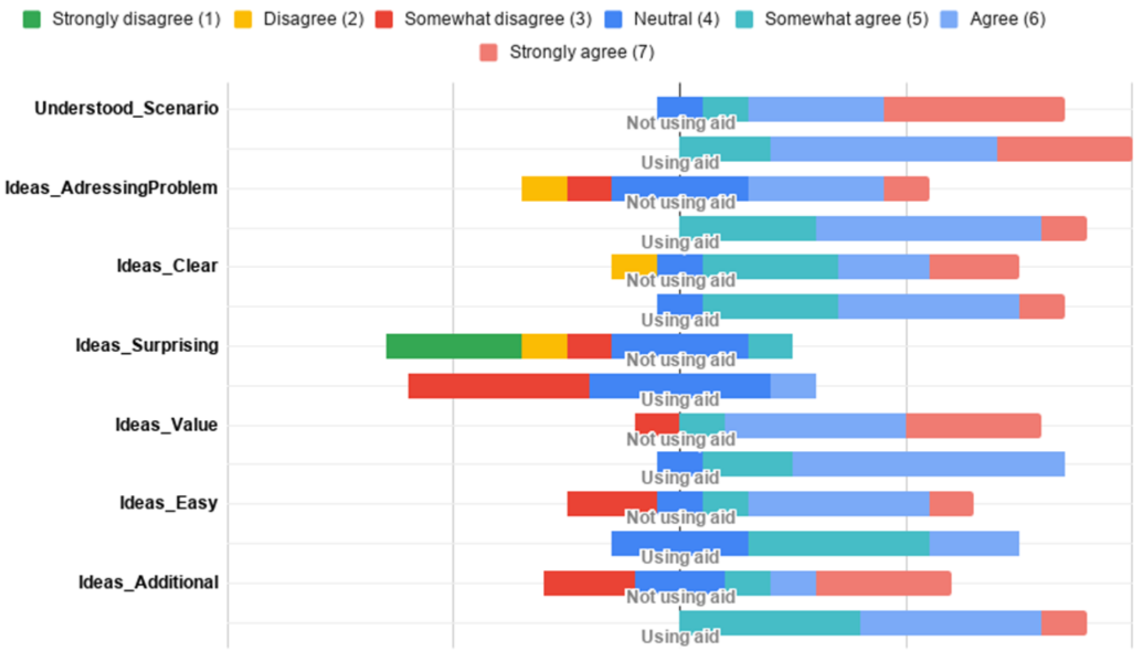


Figure 13 Complete trigger scores distribution

Light trigger
Similar to the first trigger, the results of the *Light* trigger show some questions being scored lower for the group using *Feature.Ly* compared to the group that didn't, as displayed in Table 11 and visualized in Figure 14. Participants using the application scored their understanding of the scenario, their perception of whether or not the ideas were addressing the problem stated in the scenario, and their perception of the surprising factor regarding their ideas lower than the group that didn't. For this trigger, however, participants rated the value of their ideas for *Communify* the same on average across both groups.

Table 11 Light trigger results

	Using Aid	Mean	Std. Deviation	Min	Max
Understood_Scenario	FALSE	6.33	0.87	5.00	7.00
	TRUE	5.78	1.30	3.00	7.00
Ideas_AdressingProblem	FALSE	6.00	1.12	4.00	7.00
	TRUE	5.67	1.12	4.00	7.00
Ideas_Clear	FALSE	4.89	1.27	2.00	6.00
	TRUE	5.00	0.87	3.00	6.00
Ideas_Surprising	FALSE	3.89	1.76	1.00	7.00
	TRUE	3.22	0.83	2.00	5.00
Ideas_Value	FALSE	5.67	0.71	5.00	7.00
	TRUE	5.67	0.50	5.00	6.00
Ideas_Easy	FALSE	4.67	1.12	3.00	6.00

	TRUE	5.44	1.13	3.00	7.00
Ideas_Additional	FALSE	5.00	1.22	3.00	7.00
	TRUE	5.22	1.56	3.00	7.00

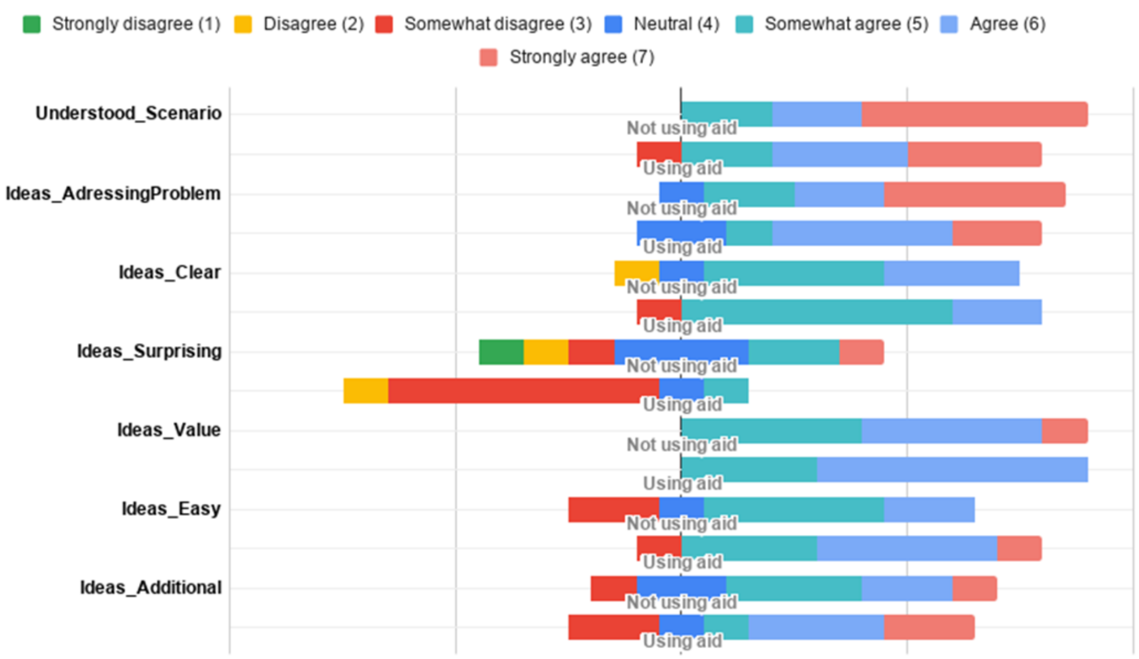


Figure 14 Light trigger scores distribution

Adaptable trigger

Table 12 and Figure 15 show the results for the *Adaptable* trigger, included in the *Edusphere* scenario. Just like the previous two triggers, participants using *Feature.Ly* scored their understanding of the scenario lower than the group of participants that didn't. Another notable score is that over the *surprising* question. The participants not using *Feature.Ly* never scored their ideas as positively surprising (i.e., higher than 4). This is reflected in the increase in the mean score of this question between groups.

Table 12 Adaptable trigger results

	Using Aid	**Mean**	**Std. Deviation**	**Min**	**Max**
Understood_Scenario	FALSE	6.00	0.71	5.00	7.00
	TRUE	5.86	1.35	3.00	7.00
Ideas_AdressingProblem	FALSE	5.00	1.41	2.00	7.00
	TRUE	5.86	1.07	4.00	7.00
Ideas_Clear	FALSE	5.67	1.22	3.00	7.00
	TRUE	6.00	0.82	5.00	7.00
Ideas_Surprising	FALSE	2.67	1.00	1.00	4.00
	TRUE	4.29	1.11	3.00	6.00

Ideas_Value	FALSE	5.78	0.67	5.00	7.00
	TRUE	5.86	1.46	3.00	7.00
Ideas_Easy	FALSE	4.89	1.05	3.00	6.00
	TRUE	5.29	1.60	2.00	7.00
Ideas_Additional	FALSE	4.22	1.56	2.00	6.00
	TRUE	5.71	1.38	3.00	7.00

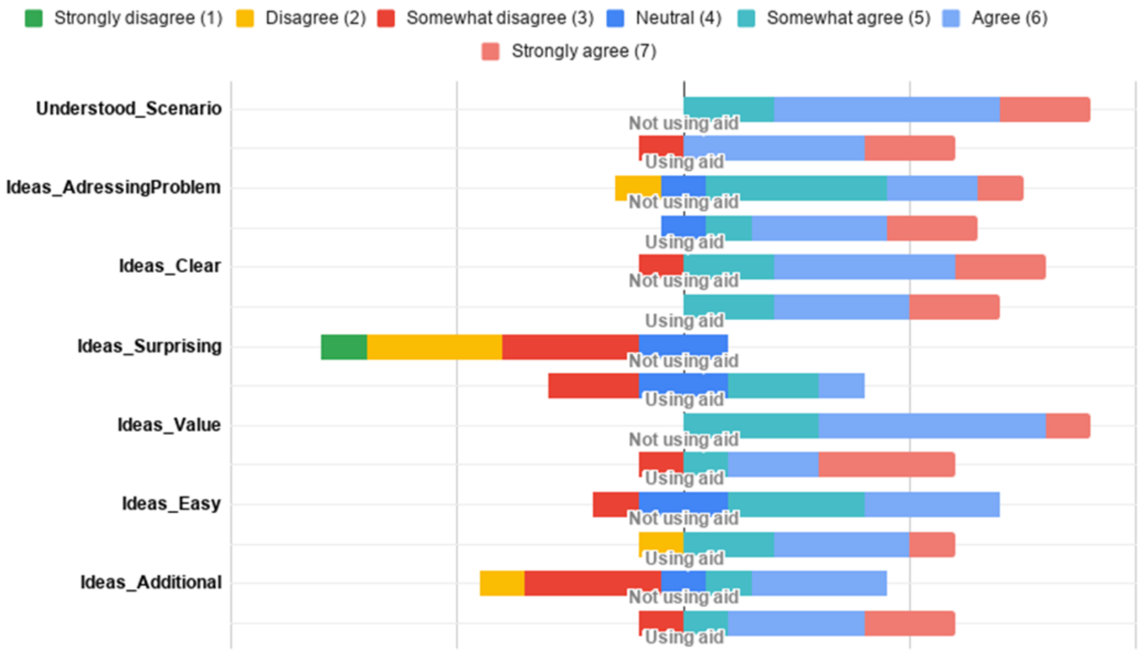


Figure 15 Adaptable trigger scores distribution

Entertaining trigger

The results for the second trigger within the *Edusphere* scenario are presented and visualized in Table 13 and Figure 16 respectively. Unlike the other triggers, the group of participants using *Feature.Ly* scored each question higher than the group that didn't use the application.

Table 13 Entertaining trigger results

	Using Aid	**Mean**	**Std. Deviation**	**Min**	**Max**
Understood_Scenario	FALSE	6.11	0.93	4.00	7.00
	TRUE	6.38	0.92	5.00	7.00
Ideas_AdressingProblem	FALSE	5.33	0.87	4.00	6.00
	TRUE	6.13	1.13	4.00	7.00
Ideas_Clear	FALSE	5.44	0.73	5.00	7.00
	TRUE	6.00	0.76	5.00	7.00
Ideas_Surprising	FALSE	3.00	1.58	1.00	6.00
	TRUE	4.13	1.12	2.00	6.00

Ideas_Value	FALSE	5.56	0.88	4.00	7.00
	TRUE	5.75	1.04	4.00	7.00
Ideas_Easy	FALSE	4.44	1.67	2.00	7.00
	TRUE	5.88	1.36	3.00	7.00
Ideas_Additional	FALSE	4.33	1.58	2.00	7.00
	TRUE	5.25	1.58	3.00	7.00

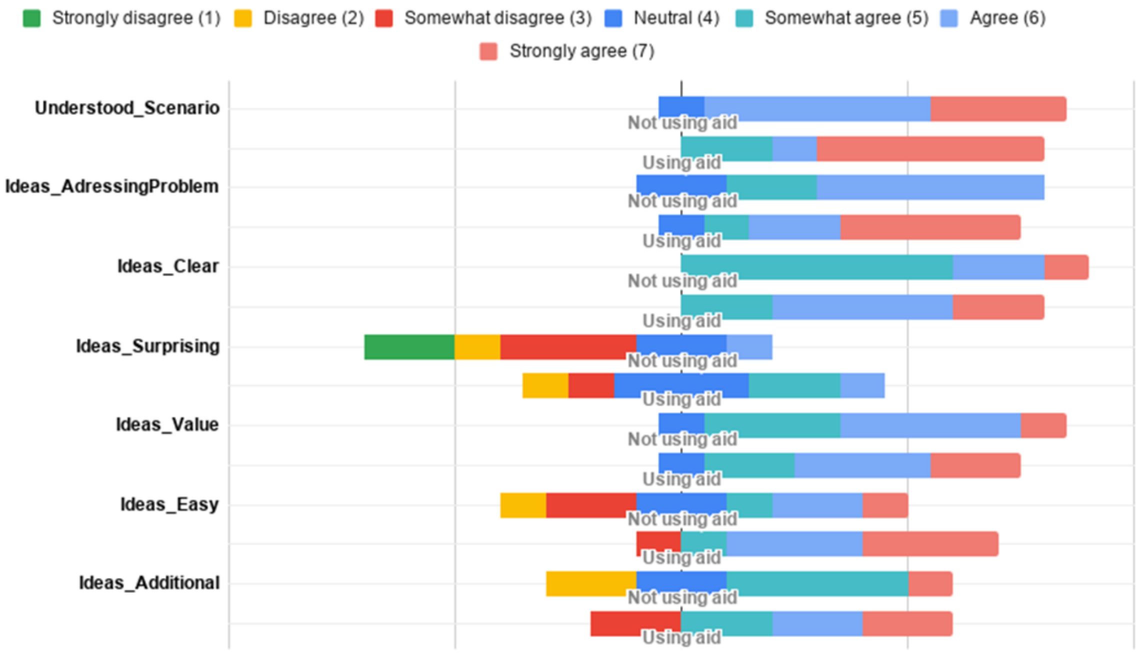


Figure 16 Entertaining trigger scores distribution

7.3 Idea generation – Analysis

Overall, with the exceptions of the questions noted in the previous section, participants scored their confidence in their ideas, the quality of these ideas, and the ease of coming up with them higher when using *Feature.Ly* as a support tool than participants that didn't. However, whether or not these increases in mean scores are due to the introduction of *Feature.Ly* will be analysed in this section.

Before choosing between either parametric or non-parametric analysis methods, a normality test has to be performed. Due to the low sample size (n<50), in addition to having only one independent variable, the use of *Feature.Ly*, the Shapiro-Wilk Test of Normality is best suited [54]. The results of this test are presented in Table 14. Any results indicating that the data is normally distributed are highlighted with bold text. The table shows some datasets to be distributed normally but not consistently among questions and groups.

Table 14 Shapiro-Wilk Test of Normality results

	Using Aid	**Complete trigger**		**Light trigger**		**Adaptable trigger**		**Entertaining trigger**	
		df	*Sig.*	*df*	*Sig.*	*df*	*Sig.*	*df*	*Sig.*

Understood_Scenario	FALSE	9	0.039	9	0.005	9	0.049	9	0.007
	TRUE	9	**0.055**	9	**0.080**	7	0.006	8	0.002
Ideas_AdressingProblem	FALSE	9	**0.405**	9	**0.065**	9	**0.239**	9	0.005
	TRUE	9	0.028	9	**0.102**	7	**0.294**	8	0.036
Ideas_Clear	FALSE	9	**0.290**	9	0.018	9	**0.083**	9	0.001
	TRUE	9	**0.338**	9	0.003	7	**0.144**	8	**0.093**
Ideas_Surprising	FALSE	9	**0.108**	9	**0.850**	9	**0.364**	9	**0.480**
	TRUE	9	0.006	9	0.014	7	**0.482**	8	**0.792**
Ideas_Value	FALSE	9	0.018	9	0.024	9	0.028	9	**0.338**
	TRUE	9	0.001	9	0.000	7	**0.059**	8	**0.408**
Ideas_Easy	FALSE	9	**0.105**	9	**0.102**	9	**0.194**	9	**0.740**
	TRUE	9	**0.055**	9	**0.106**	7	**0.066**	8	0.036
Ideas_Additional	FALSE	9	**0.096**	9	**0.830**	9	**0.108**	9	**0.181**
	TRUE	9	0.024	9	**0.149**	7	**0.099**	8	**0.175**

Even though some data sets appear to be normally distributed, all tests performed will be non-parametric, using the Mann Whitney U test [55]. Given the small sample size per group, even for the Shapiro-Wilk Test, provides relatively little statistical power and confidence in the actual normal distribution of the data, as it might be the result of noise in the data [56].

Furthermore, since the data is comprised of solely ordinal data, i.e., Likert scales, retrieved from two independent samples, the only suitable parametric test would be a *t* test [55]. As this test requires both groups to be normally distributed, this would leave only a select few data sets eligible for the test. The other data sets would require its non-parametric counterpart, the Mann Whitney U Test [55]. In order to be consistent in the tests, in addition to the earlier mentioned low confidence in the Shapiro-Wilk test due to the small sample size the datasets that appear to be normally distributed might as well be included in these non-parametric tests. The results of these tests are presented in Table 15.

Table 15 Mann Whitney U Test (α = 0.05)

	Complete trigger	**Light trigger**	**Adaptable trigger**	**Entertaining trigger**
Null Hypothesis	*Sig. value*	*Sig. value*	*Sig. value*	*Sig. value*
The distribution of **Understood_Scenario** is the same across categories of *UsingAid*	0.863	0.387	0.837	0.541
The distribution of **Ideas_AdressingProblem** is the same across categories of *UsingAid*	0.190	0.546	0.210	0.114
The distribution of **Ideas_Clear** is the same across categories of *UsingAid*	0.796	1.000	0.758	0.167
The distribution of **Ideas_Surprising** is the same across categories of *UsingAid*	0.297	0.297	**0.016**	0.114
The distribution of **Ideas_Value** is the same across categories of *UsingAid*	0.297	0.931	0.536	0.673
The distribution of **Ideas_Easy** is the same across categories of *UsingAid*	0.546	0.161	0.351	0.093

The distribution of **Ideas_Additional** is the same across categories of *UsingAid*	0.546	0.666	0.071	0.236

Complete trigger

The Mann Whitney U test for the complete trigger questions show no significant difference (sig. value < 0.05) between the group using *Feature.Ly* and the group that didn't. Thus, all null hypotheses for the complete trigger questions will be retained.

Light trigger

Like the previous trigger, the test results for the light trigger show no significant differences between the group using a support tool, i.e. *Feature.Ly*, compared to the group not using one. Again, all null hypotheses for this trigger will be retained.

Adaptable trigger

The adaptable trigger, does show some significant difference between groups, where the perception of the surprising factor of participant's ideas was ranked significantly higher when using the application compared to not using it. For this question, the null hypothesis will be rejected. The other questions show no significant difference and for these, the null hypothesis will be retained.

Entertaining trigger

Following the first two triggers, the entertaining trigger test results also show no significant differences between using *Feature.Ly* and not using it. As such, all null hypotheses will be retained.

Overall, the Mann Whitney U test results show no evidence of *Feature.Ly* significantly supporting participants in coming up with new ideas for the factors that were tested. Comparison between triggers, however, does show some large differences in their significance scores. The implications and possible reasoning for these abnormalities will be discussed in Chapter 8.

7.4 Application evaluation

The result regarding the evaluation of the *Feature.Ly* application can be found in Table 16. Although no tests will be performed on this data, it shows the participant's overall impression of the effect the application had on during the survey. Considering that, for the 7-point Likert-scale, a 4 would be considered neutral, overall, the application had a positive effect, especially for the ease of coming up with ideas.

No statistical testing can be performed on this dataset as this would require a lot of assumptions to be made regarding the difficulty and equality of the scenarios due to each group using the application for a different scenario (see Figure 10). The descriptive statistics in Table 16 are added for sake of completeness and consistency.

Table 16 Application evaluation results

	Mean	Std. Deviation	Min	Max
Ideas_Confidence_Total	4.89	1.28	3.00	7.00
Ideas_Surprising_Total	4.17	1.38	2.00	7.00
Ideas_Valuable_Total	4.94	0.80	4.00	7.00
Ideas_Easy_Total	5.61	1.20	4.00	7.00

As mentioned in the design, the final set of questions also provided the option to fill in any comments the participants had on the use of the application. As this question was optional, not all participants have commented but the comments of those that did are shown in Table 17.

Table 17 Application evaluation participants' comments

Comments
Especially the examples are very good triggers to come up with new ideas. I think it would be cool to also have a 'random' category. Usually the ideas I came up with using the app didn't fall it the category the example belonged to.
While feature.ly is a nice way to help with ideas, I feel like it would mostly benefit the older generations who didn't grew up with today’s technology. During the questions I noticed I directly compared features I've seen in the past or even on well-known platforms as it acts as a benchmark what the general public actually likes to have or what I personally would like to see. By growing up with technology and actively working with all types of techniques you more or less know what's possible to do and add.
Looking at existing features may help me coming up with 'obvious' features, but I didn't come up with any 'new' features because of it
The types of examples listed in the application did not really make sense to me. All examples seemed similar and I did not really see a difference
Most example I could relate to work, e.g. a community platform is used in every company I suppose, therefore you can relate and easily come up with additional functionalities. Whilst I was clicking through the examples, I did get some extra inspiration which actually made me add another idea in the direction I wasn't thinking about before. I would recommend changing some of the texts, "doing more things" is bit too easy imo, which makes the tool look less sophisticated, I guess. Adding more examples, triggering creativity would definitely add more value but with this simple example I did already experience the added value.
I didn't use it that much because it is a bit confusing with the refreshing of the examples. In my opinion, the examples did not always match the description above.
Seems like a good idea! Only remark is the small number of examples in some cases

A few comments explain that the different categories (i.e. triggers) do not make sense. They see no real difference between the type of examples and did not help in finding the right type of idea. One of these participants mentioned that they would rather have all ideas in one card so they could sample all of them at once. Additionally, some of the participants had enough experience with software applications that the features suggested in the examples did not provide them with any new ideas as they could relate to the scenarios quite easily. Finally, another comment, also later reaffirmed by talking to participants after completing the survey, is the relatively low number of examples in some trigger cards. This problem was also mentioned in Section 5.3 when constructing the feature set to be used in the examples.

7.5 Expert review

This section analyses the results from the review two experts did on the generated ideas to obtain an outsider's perspective regarding the idea's creativity. Ideas that were unclear to the experts were removed from the dataset before analysis.

The data from the two experts was not reconciled due to the inherent subjective nature of creativity, which differs per person. The expert review would represent this as accurately as possible by analysing the sets of data separately. Reconciliation of the data would result in an average of the two, representing neither of the experts.

Analysis was done using the Mann Whitney U test. Like the analysis of the idea generation results, there are still two independent samples, one group of ideas generated without *Feature.Ly* and one group of ideas generated using the application. There was no need for testing for normally distributed data as each data point came from the same expert. Thus, the data could never be normally distributed.

Complete trigger

The results regarding the review of the complete trigger are presented in Table 18 and Table 19 for Expert 1 and Expert 2, respectively. Expert 1 rated most ideas as addressing the problem, with only a few exceptions. The value of the ideas was more divided, with a decrease in value in the ideas that were generated using the application compared to the ideas that didn't. Most notable is the overall low scores Expert 1 gave the ideas regarding the surprising factor, with averages well below a neutral score of 4 on the 1-7 Likert scale.

Table 18 Expert 1 - Complete trigger results

	Using Aid	Mean	Std. Deviation	Min	Max
Ideas_AdressingProblem	FALSE	7.00	0.00	7.00	7.00
	TRUE	6.89	0.58	4.00	7.00
Ideas_Surprising	FALSE	2.33	1.39	1.00	5.00
	TRUE	2.81	1.86	1.00	7.00
Ideas_Value	FALSE	6.38	1.12	4.00	7.00
	TRUE	5.96	1.16	3.00	7.00

Similar to Expert 1, Expert 2 rated most ideas as addressing the problem. However, in this case the ideas generated with *Feature.Ly* are scored lower on average than those without it. The value of the ideas, however, is scored very similar on average between groups. Again, the ideas are scored relatively low on the surprising factor even though there is an increase in the group using the application compared to the one that is not.

Table 19 Expert 2 - Complete trigger results

	Using Aid	Mean	Std. Deviation	Min	Max
Ideas_AdressingProblem	FALSE	6.57	1.03	3.00	7.00
	TRUE	6.48	0.89	5.00	7.00
Ideas_Surprising	FALSE	3.90	1.51	1.00	6.00
	TRUE	4.67	1.18	2.00	6.00
Ideas_Value	FALSE	5.52	1.03	3.00	7.00
	TRUE	5.56	0.80	3.00	7.00

Light trigger
Table 20 and Table 21 show the results regarding the light trigger. Similar to the complete trigger, Expert 1 rated the generated ideas consistently high on the addressing problem question. Additionally, the ideas generated with the application scored lower than those generated with it regarding the value of the ideas. Again, Expert 1 scored the surprising factor of the ideas low.

Table 20 Expert 1 - Light trigger results

	Using Aid	Mean	Std. Deviation	Min	Max
Ideas_AdressingProblem	FALSE	6.93	0.38	5.00	7.00
	TRUE	7.00	0.00	7.00	7.00
Ideas_Surprising	FALSE	1.70	1.23	1.00	4.00
	TRUE	2.07	1.47	1.00	7.00
Ideas_Value	FALSE	5.89	0.93	4.00	7.00
	TRUE	5.70	0.78	5.00	7.00

Expert 2 scored all ideas for the light trigger consistently for the addressing problems and value questions between groups, with both only increasing slightly. Notable, however, is the decrease in the average score of the surprising question for the group using *Feature.Ly*.

Table 21 Expert 2 - Light trigger results

	Using Aid	Mean	Std. Deviation	Min	Max
Ideas_AdressingProblem	FALSE	6.85	0.36	6.00	7.00
	TRUE	6.88	0.32	6.00	7.00
Ideas_Surprising	FALSE	3.19	1.24	2.00	6.00

	TRUE	2.85	1.56	1.00	7.00
Ideas_Value	FALSE	5.70	0.99	3.00	7.00
	TRUE	5.77	0.70	4.00	7.00

Adaptable trigger

Not unlike the previous triggers, Expert 1 (Table 22) rated the ideas similar regarding whether or not they are addressing the problem. Again, the ideas were scored lower on the value question for those that were generated with the application used compared to those without. Also

Table 22 Expert 1 - Adaptable trigger results

	Using Aid	**Mean**	**Std. Deviation**	**Min**	**Max**
Ideas_AdressingProblem	FALSE	6.76	0.96	3.00	7.00
	TRUE	6.60	1.23	3.00	7.00
Ideas_Surprising	FALSE	2.18	1.62	1.00	7.00
	TRUE	2.20	1.51	1.00	5.00
Ideas_Value	FALSE	5.21	1.30	1.00	7.00
	TRUE	4.85	1.73	1.00	7.00

Expert 2 (Table 23) scored the ideas with support from *Feature.Ly* lower than those without for the addressing problems and value questions. There was, however, an increase in the surprising factor of the ideas for the ideas generated with support.

Table 23 Expert 2 - Adaptable trigger results

	Using Aid	**Mean**	**Std. Deviation**	**Min**	**Max**
Ideas_AdressingProblem	FALSE	6.76	0.74	4.00	7.00
	TRUE	6.40	1.10	3.00	7.00
Ideas_Surprising	FALSE	3.59	1.96	1.00	7.00
	TRUE	4.25	1.59	2.00	6.00
Ideas_Value	FALSE	6.41	0.86	3.00	7.00
	TRUE	6.10	0.97	4.00	7.00

Entertaining trigger

The results for the final trigger in the expert review are presented in Table 24 and Table 25. Consistent among the previous triggers, Expert 1 scored the questions regarding the ideas' value and addressing the problem similar between groups, albeit with a decrease in average score for the value question for the ideas generated with *Feature.Ly*. Again, the surprising question is scored quite below the neutral score of the 1-7 Likert scale.

Table 24 Expert 1 - Entertaining trigger results

	Using Aid	Mean	Std. Deviation	Min	Max
Ideas_AdressingProblem	FALSE	6.42	1.43	1.00	7.00
	TRUE	6.61	1.03	4.00	7.00
Ideas_Surprising	FALSE	2.74	1.12	1.00	4.00
	TRUE	3.61	1.56	1.00	7.00
Ideas_Value	FALSE	4.71	1.10	3.00	7.00
	TRUE	4.39	1.50	1.00	7.00

Expert 2, however, does have some notable results. Scoring the ideas generated with the application lower on each question compared to the ones without, albeit slightly. Also notable is the relatively high score on the surprising factor, which was quite low for the other triggers.

Table 25 Expert 2 - Entertaining trigger results

	Using Aid	Mean	Std. Deviation	Min	Max
Ideas_AdressingProblem	FALSE	6.90	0.40	5.00	7.00
	TRUE	6.74	0.75	4.00	7.00
Ideas_Surprising	FALSE	5.06	1.53	1.00	7.00
	TRUE	5.00	1.48	1.00	7.00
Ideas_Value	FALSE	6.39	0.50	6.00	7.00
	TRUE	6.04	0.82	3.00	7.00

As mentioned before, the results of the expert review were tested using the Mann Whitney U Test. Expert 1’s results can be found in Table 26. The results show one factor being significantly different between ideas generated with *Feature.Ly* compared to those without, which is the surprising factor in the entertaining trigger, as highlighted in Table 26. As such, this null hypothesis will be rejected, while the others will be retained.

Table 26 Expert 1 - Mann Whitney U Test results (α = 0.05)

	Complete trigger	**Light trigger**	**Adaptable trigger**	**Entertaining trigger**
Null Hypothesis	*Sig. value*	*Sig. value*	*Sig. value*	*Sig. value*
The distribution of **Ideas_AdressingProblem** is the same across categories of *UsingAid*	0.378	0.317	0.580	0.723

The distribution of **Ideas_Surprising** is the same across categories of *UsingAid*	0.471	0.221	0.892	**0.007**
The distribution of **Ideas_Value** is the same across categories of *UsingAid*	0.120	0.422	0.546	0.580

Expert 2’s results are presented in Table 27. Some values are nearly significant, such as the surprising factor for the complete trigger, the addressing problem scores for the adaptable trigger, and the value score for the entertaining trigger. However, none of these scores represent a significant change in scores between groups and thus all hypotheses will be retained.

Table 27 Expert 2 - Mann Whitney U Test results (α = 0.05)

	Complete trigger	**Light trigger**	**Adaptable trigger**	**Entertaining trigger**
Null Hypothesis	*Sig. value*	*Sig. value*	*Sig. value*	*Sig. value*
The distribution of **Ideas_AdressingProblem** is the same across categories of *UsingAid*	0.580	0.688	0.049	0.394
The distribution of **Ideas_Surprising** is the same across categories of *UsingAid*	0.087	0.135	0.152	0.883
The distribution of **Ideas_Value** is the same across categories of *UsingAid*	0.592	0.898	0.193	0.096

8. Discussion

This chapter will consist of the results' interpretation, the limitations of the experiment, and conclusions to be drawn from this analysis, if any.

8.1 Statistical findings

The results indicate that the current version of the *Feature.Ly* application, within the context of the four problem statements tested in the experiment, does not seem to provide users with extra confidence regarding ideas they come up with. Nor does it seem to help users to generate ideas that they perceive as more valuable or applicable for the problem at hand. Furthermore, the application does not seem to help users in coming up more different ideas. The application, however, appears to help users in generating surprising ideas, i.e. ideas not seen before in similar applications, for one of the four problem statements; the adaptable trigger.

The subsequent expert review also did not provide any significant evidence supporting the intended effect of the application on the ideas generated, except for the surprising factor in the entertaining trigger problem statement. This effect, however, was only found for one of the experts reviewing the generated ideas.

Nevertheless, users did rate their experience with the *Feature.Ly* application as positive and perceived the application to help them during the survey, albeit not very conclusive given the slightly higher than neutral scores given.

8.2 General findings

Some effects can be deduced from the scores when comparing them between problem statements. It should be noted that these effects are not significant, nor is this implied. These findings are purely based on the qualitative analysis of the results presented in Chapter 7.

Surprising factor

As mentioned above, the surprising factor was the only factor in which the use of the application generated significantly more surprising ideas, i.e. not seen before in this type of applications. This effect was only present for the problem statement accompanying the adaptable trigger in the idea evaluation survey and for the entertaining trigger in the review of expert 1.

The reason for this significance may lie in the goal of the problem statement. The adaptable trigger asks for ideas to make the scenario's application controllable and more configurable. These ideas may be highly subjective and thus ideas generated for this problem statement may be surprising due to the participants generating ideas that would be useful for themselves specifically.

Biggest potential

Similar to the section above, Table 15 shows the surprising factor to have the lowest significance scores across all triggers. Although three of four of these scores do not indicate an effect that is significant, they do show the most potential as they are affected the most out of all scores when using the application, compared to not using it. This opens up new opportunities for future work to explore the effect of the application on the surprising factor of generated ideas.

Trigger difficulty

When analysing the Mann Whitney U Test results in Table 15, an interesting relationship can be deduced. When looking at the *Understood_Scenario* and *Ideas_Easy* questions, the significance scores for the light and entertaining triggers are both lower than the complete and adaptable triggers. There is no significant effect, but there is a bigger effect for the light and entertaining triggers compared to the other two. This difference could be explained due to the light and entertaining triggers to be more difficult to understand and thus generate ideas for. Whereas 'making the app do more things' and 'make the app more configurable' are relatively straightforward tasks, 'making the app easier to use' and 'make the app retain users' require the participant to think differently. In these situations, *Feature.Ly* could provide additional guidance helping users understand and interpret the type of problems that need to be tackled.

Tendencies towards significant effects

Although the results show only one factor to have a significant effect, there are some factors with a low enough significance score to make them stand out. For instance, the ease of generating ideas (Ideas_Easy) for the entertaining trigger is close to a significant effect. As is the possibility of generating additional ideas for the adaptable trigger. Similarly, expert 2's review shows the scores regarding the value of ideas for the entertaining trigger and the surprising factor for the complete trigger close to a significant effect. These scores show the potential of the application and highlight areas that could be focused on to further develop the application to make it more effective in its goal.

Expert experience

The results regarding the different triggers for the reviews done by the expert show some interesting differences. While both experts were generally agreeing, based on their scores, about whether or not ideas were addressing the problem and the ideas' value, they scored the surprising factor very differently. Expert 1 was much harsher in their perception of ideas being surprising compared to expert 2. This may be due to expert 1 having significantly more experience with various different software applications and thus having seen more different implementations of features. This shows the effect of experience on the overall perception of ideas and further shows the quality of ideas to be very subjective.

8.3 Limitations

There are number of factors that could have negatively influence the experiment and its analysis of the results.

Sample size

The first of these potential limitations is the sample size. With 18 participants in total, 9 for each group, it is hard to argue the sample size is representative of the population, which it isn't. As such, the result of the application having a significant positive effect on the surprising factor could very well be, and likely is, just a coincidence. In order for the experiment to be more representative of the population, more participants should be involved.

Survey duration

The duration of participating in the survey is also a potential limitation. As a low number of participants was expected from the start, the survey had to include all different types of problem statements and still expose the participant to the *Feature.Ly* application. This resulted in the survey taking up to 40 minutes to complete, which is a long time for a survey and could have further reduced the number of willing participants. Furthermore, this could have resulted in participants being impatient and less focused at the end of the survey, reducing their capabilities regarding idea generation.

Treatment distribution

The use of the application had to be done after not using it, as it could otherwise have an effect on ideas generated without it when participant had already seen some examples. Nevertheless, this, combined with the potential reduced focus and willingness of participants, could have resulted in the ideas generated to be of lesser quality, as well as participants' perception of the ideas when having to work through 4 different problem statements.

Feature data set

As some comments in the application evaluation suggested, the number of features included in the *Feature.Ly* application were limited for certain triggers and apparently separating them into categories did not add anything to their relatability. It has to be taken into account these comments are from only a few of the participants, but it might be valid. Perhaps a more extensive set of features, such as features that aren't included in some of the most popular apps on the app store, would have created other results.

App Store reviews

As mentioned in section 5.3, the reviews used for creating the dataset were extracted only from the *App Store* and not from the *Google Play Store* due to an outdated scraper library. Perhaps the quality of reviews from users of the *Google Play Store* is better or users mention different features as some functionality may be differently integrated which users like or dislike more.

Fictive experiment setting

The experiment used in the thesis was a fictive one. The scenarios were constructed with a specific goal in mind and may not have appealed to some of the participants or were just not an accurate representation of how such problem statements occur in software development projects. Perhaps using the method in a real setting may result in an overall different conclusion.

9. Conclusion & Future work

This research aimed to devise a technique that uses data to improve creativity during requirement elicitation in agile environments. This was done by first exploring the core activities of requirements elicitation, techniques used, and how the activities change in agile environments. Furthermore, it explored current techniques and method available to aid creativity during requirements elicitation sessions.

Following the literature review, a full-fledged technique, balled *BrainstormPlus*, was envisioned based on the traditional brainstorming technique. The brainstorm technique was chosen after creating a taxonomy highlighting the characteristics of a technique suitable for integrating in a creativity focused elicitation session. Due to the scope and time constraint of this research, however, this technique was not viable for evaluation.

As a result, a section of *BrainstormPlus* was taken as a starting point to be separately evaluated. This resulted in the development of *Feature.Ly*, a technique developed to utilize app reviews from the *App Store* to supply features to be used during brainstorming sessions as a means to trigger creativity on demand and without extensive preparation.

This technique was then tested through a two-part evaluation in which participants were asked to come up with ideas following a set of problem statements in context of a scenario directing them what kind of ideas to generate, after which they were asked to rate the ideas they just came up with. After collecting, two experts evaluated the ideas in terms of them addressing the problem, being surprising, and their value to the scenario's application.

Results proved to be inconclusive during both parts of the evaluation. Nevertheless, during the final application evaluation survey, participants rated their experience with the application as positive and saw potential in the application's intended use.

So, even though the technique in its current state does not have the initially intended effect, it does show potential and steps are made towards creating a technique that can aid in creativity during requirements elicitation and can be deployed ad hoc to suit agile environments.

9.1 Future work

Recommendations of future work are plentiful, given the inconclusive results of the evaluation. One of the most evident actions to be taken is to re-evaluate the technique with a larger sample size and rethink the method of evaluation to make it less taxing on the participants. This also includes evaluating the technique in a real world setting instead of a fictive one as done in this research.

Additionally, the current evaluation saw some tendencies towards a significant effect. Successive work could focus solely on these effects, such as the ease of coming up with new ideas as to help the quantity of ideas being produced, some of which are likely to be useful. Another is the surprising factor, which showed the most potential toward having a significant effect.

Furthermore, future work could focus on the effect of *Feature.Ly* when the problem statements are more difficult to grasp may require more granulated ideas instead of general improvements.

Finally, future work could focus on introducing different types of features as the dataset used for the application's examples. By using less generic and popular applications of which people do not already know the features, they may be stimulated in different ways. This also includes the quality of the features, which can be improved by further analysis and perhaps retrieval from the *Google Play Store* as well.

Bibliography

[1] H. van den Broeck, E. Cools, and T. Maenhout, “A case study of arteconomy - Building a bridge between art and enterprise: Belgian businesses stimulate creativity and innovation through art,” *J. Manag. Organ.*, vol. 14, no. 5, pp. 573–587, 2008.

[2] J. S. McMullen and D. A. Shepherd, “Entrepreneurial Action and the Role of Uncertainty in the Theory of the Entrepreneur,” *Acad. Manag. Rev.*, vol. 31, no. 1, pp. 3–23, 2006.

[3] M. Mumford, “Where Have We Been, Where Are We Going? Taking Stock in Creativity Research,” *Creat. Res. J.*, vol. 15, no. 2, pp. 107–120, 2003.

[4] B. Nuseibeh and S. Easterbrook, “Requirements engineering: a roadmap,” in *Proceedings of the Conference on the Future of Software Engineering*, 2000, vol. 1, pp. 35–46.

[5] N. Maiden and A. Gizikis, “Where Do Requirements Come From?,” *IEEE Softw.*, vol. 18, no. 5, pp. 10–12, 2001.

[6] N. Maiden, S. Manning, S. Robertson, and J. Greenwood, “Integrating Creativity Workshops into Structured Requirements Processes,” in *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, 2004, vol. 6, pp. 113–122.

[7] J. Lemos, C. Alves, L. Duboc, and G. N. Rodrigues, “A systematic mapping study on creativity in requirements engineering,” in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, 2012, no. March, pp. 1083–1088.

[8] I. K. Karlsen, N. Maiden, and A. Kerne, “Inventing requirements with creativity support tools,” in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2009, vol. 5512 LNCS, pp. 162–174.

[9] S. Jones and N. Maiden, “RESCUE: An Integrated Method for Specifying Requirements for Complex Sociotechnical Systems,” *Requir. Eng. Sociotechnical Syst.*, pp. 245–265, 2005.

[10] K. Zachos and N. Maiden, “ART-SCENE: Enhancing scenario walkthroughs with multi-media scenarios,” in *Proceedings of the 12th IEEE International Conference on Requirements Engineering*, 2004, vol. 2, pp. 360–361.

[11] J. Fernandes, D. Duarte, C. Ribeiro, C. Farinha, J. M. Pereira, and M. M. da Silva, “iThink: A Game-Based Approach Towards Improving Collaboration and Participation in Requirement Elicitation,” *Procedia Comput. Sci.*, vol. 15, pp. 66–77, 2012.

[12] M. A. Boden, “Computer models of creativity,” *Psychologist*, vol. 13, no. 2, pp. 72–76, 2000.

[13] F. Paetsch, A. Eberlein, and F. Maurer, “Requirements engineering and agile software development,” in *Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2003, pp. 308–313.

[14] A. van Lamsweerde, “Requirements engineering in the year 00: a research perspective.,” in *Proceedings of the 22nd international conference on software engineering*, 2000, pp. 5–19.

[15] D. Zowghi and C. Coulin, *Requirements elicitation: A survey of techniques, approaches, and tools*. Berlin, Heidelberg: Springer, 2005.

[16] N. Wever, “Synthesizing Creative Requirements with Natural Language Processing,” Utrecht University, 2018.

[17] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems

research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.

[18] R. Wieringa, "Design science as nested problem solving," in *Proceedings of the 4th international conference on design science research in information systems and technology*, 2009, pp. 8–20.

[19] A. M. Hickey and A. M. Davis, "A unified model of requirements elicitation," *J. Manag. Inf. Syst.*, vol. 20, no. 4, pp. 65–84, 2004.

[20] C. J. Davis, R. M. Fuller, M. C. Tremblay, and D. J. Berndt, "Communication Challenges in Requirements Elicitation and the Use of the Repertory Grid Technique," *J. Comput. Inf. Syst.*, vol. 46, no. 5, pp. 78–86, 2006.

[21] M. G. Christel and K. C. Kang, "Issues in Requirements Eflictation," 1992.

[22] T. Tuunanen, "A new perspective on requirements elicitation methods," *J. Inf. Technol. Theory Appl.*, vol. 5, no. 3, pp. 45–62, 2003.

[23] J. A. Goguen and C. Linde, "Techniques for Requirements Elicitation," in *Proceedings of the IEEE International Symposium on Requirements Engineering*, 1993, pp. 152–164.

[24] K. Holtzblatt and H. Beyer, "Making customer-centered design work for teams," *Commun. ACM*, vol. 36, no. 10, pp. 92–103, 1993.

[25] A. Van Lamsweerde, R. Darimont, and E. Letier, "Managing conflicts in goal-directed requirements engineering," *IEEE Trans. Softw. Eng.*, vol. 24, no. 11, pp. 908–925, 1998.

[26] N. Maiden, S. Minocha, A. Sutcliffe, D. Manuel, and M. Ryan, "Co-operative scenario based approach to acquisition and validation of system requirements: how exceptions can help!," *Interact. Comput.*, vol. 11, no. 6, pp. 645–664, 1999.

[27] Z. Zhang, "Effective Requirements Development - A Comparison of Requirements Elicitation techniques," *Softw. Qual. Knowl. Soc.*, no. May, pp. 225–240, 2007.

[28] M. Yousuf and M. A. M.Asger, "Comparison of Various Requirements Elicitation Techniques," *Int. J. Comput. Appl.*, vol. 116, no. 4, pp. 8–15, 2015.

[29] R. Agarwal and M. R. Tanniru, "Knowledge acquisition using structured interviewing: an empirical investigation," *J. Manag. Inf. Syst.*, vol. 7, no. 1, pp. 123–140, 1990.

[30] J. R. Olson and H. H. Rueter, "Extracting expertise from experts: Methods for knowledge acquisition," *Expert Syst.*, vol. 4, no. 3, pp. 152–168, 1987.

[31] G. Lindgaard *et al.*, "User Needs Analysis and requirements engineering: Theory and practice," *Interact. Comput.*, vol. 18, no. 1, pp. 47–70, 2006.

[32] J. Richardson, T. C. Ormerod, and A. Shepherdb, "The role of task analysis in capturing requirements for interface design," *Interact. Comput.*, vol. 9, no. 4, pp. 367–384, 1998.

[33] A. Sutcliffe and N. Maiden, "The domain theory for requirements engineering," *IEEE Trans. Softw. Eng.*, vol. 24, no. 3, pp. 174–196, 1998.

[34] R. Prieto-Díaz, "Domain analysis: An introduction," *ACM SIGSOFT Softw. Eng. Notes*, vol. 15, no. 2, pp. 47–54, 1990.

[35] B. Mullen, C. Johnson, and E. Salas, "Productivity Loss in Brainstorming Groups : A Meta-Analytic Integration," *Basic Appl. Soc. Psych.*, vol. 12, no. 1, pp. 3–23, 1991.

[36] A. Sutcliffe, "Scenario-based requirements engineering," in *11th IEEE International Requirements Engineering Conference*, 2003, pp. 320–329.

[37] D. Duarte, C. Farinha, M. M. Da Silva, and A. R. Da Silva, "Collaborative requirements elicitation with visualization techniques," in *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2012, pp. 343–

348.

[38] M. A. Boden, *The creative mind: Myths and mechanisms*. Routledge, 2004.

[39] D. Leonard and S. Sensiper, “The Role of Tacit Knowledge in Group Innovation,” *Calif. Manage. Rev.*, vol. 40, no. 3, pp. 112–132, 1998.

[40] N. Maiden, S. Jones, K. Karlsen, R. Neill, K. Zachos, and A. Milne, *Requirements engingeering as creative problem solving: A research agenda for idea finding*. IEEE, 2010.

[41] L. Mich, C. Anesi, and D. M. Berry, “Applying a pragmatics-based creativity-fostering technique to requirements elicitation,” *Requir. Eng.*, vol. 10, no. 4, pp. 262–275, 2005.

[42] C. Burnay, J. Horkoff, and N. Maiden, “Stimulating Stakeholders’ Imagination : New Creativity Triggers for Eliciting Novel Requirements,” in *2016 IEEE 24th International Requiremenst Engineering Conference (RE)*, 2016, pp. 36–45.

[43] O. Hoffmann, D. Cropley, A. Cropley, L. Nguyen, and P. Swatman, “Creativity, Requirements and Perspectives,” *Australas. J. Inf. Syst.*, vol. 13, no. 1, pp. 159–175, 2005.

[44] L. Aziz-Zadeh, J. T. Kaplan, and M. Iacoboni, “‘Aha!’: The neural correlates of verbal insight solutions,” *Hum. Brain Mapp.*, vol. 30, no. 3, pp. 908–916, 2009.

[45] L. Williams and A. Cockburn, “Agile software development: it’s about feedback and change,” *IEEE Comput.*, vol. 36, no. 6, pp. 39–43, 2003.

[46] T. Dybå and T. Dingsøyr, “Empirical studies of agile software development: A systematic review,” *Inf. Softw. Technol.*, vol. 50, no. 9–10, pp. 833–859, 2008.

[47] A. Cockburn and J. Highsmith, “Agile software development: The people factor,” *Computer (Long. Beach. Calif).*, vol. 34, no. 11, pp. 131–133, 2001.

[48] K. Beck *et al.*, “Manifesto for Agile Software Development,” 2001.

[49] J. Highsmith and A. Cockburn, “Agile software development: The business of innovation,” *Computer (Long. Beach. Calif).*, vol. 34, no. 9, pp. 120–122, 2001.

[50] B. Hollis and N. Maiden, “Extending agile processes with creativity techniques,” *IEEE Softw.*, vol. 30, no. 5, pp. 78–84, 2013.

[51] C. J. Nemeth, B. Personnaz, M. Personnaz, and J. A. Goncalo, “The Liberating Role of Conflict in Group Creativity: A study in two countries,” *Eur. J. Soc. Psychol.*, vol. 34, no. 4, pp. 365–374, 2004.

[52] H. F. Hofmann and F. Lehner, “Requirements Engineering as a Success Factor in Software Projects,” *IEEE Softw.*, pp. 55–66, 2001.

[53] A. Joshi, S. Kale, S. Chandel, and D. Pal, “Likert Scale: Explored and Explained,” *Br. J. Appl. Sci. Technol.*, vol. 7, no. 4, pp. 396–403, 2015.

[54] A. Ghasemi and S. Zahediasl, “Normality tests for statistical analysis: A guide for non-statisticians,” *Int. J. Endocrinol. Metab.*, vol. 10, no. 2, pp. 486–489, 2012.

[55] J. C. F. de Winter and D. Dodou, “Five-point likert items: T test versus Mann-Whitney-Wilcoxon,” *Pract. Assessment, Res. Eval.*, vol. 15, no. 11, 2010.

[56] N. Mohd Razali and Y. Bee Wah, “Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests,” *J. Stat. Model. Anal.*, vol. 2, no. 1, pp. 21–33, 2011.

Appendix A – Final feature dataset

App	Features	Creativity trigger	Secondary	Explanation
Spotify	List previous listened to artists	Complete		It helps you find artists that you listened to without saving them
Spotify	List just the songs saved from the artist	Light		Removes clutter from songs the user didn't save
Spotify	Shuffle songs in a playlist	Entertaining		Provides the user with their own music but in a different order every time. (Review mentions that this does not happen)
Spotify	Save favourite artists	Adaptable		User can save their favourite artists so they can configure the app as they'd like
Spotify	Create daily playlists based on favourite songs and artists	Light		An automatically created playlist helps users create playlists without them having to do it, saving time and effort
Spotify	Create personal playlists based on favourite songs and artists	Light		An automatically created playlist helps users create playlists without them having to do it, saving time and effort
Spotify	Personalize playlists based on listening history	Light		A personalized playlist reduces the effort by the user to create the playlist themselves and provides them the option to listen to all their music at once
Spotify	Create playlists	Adaptable		User can create their own music collection
Spotify	Find recommended songs	Entertaining		Recommended songs allow the user to discover new music
Spotify	Navigate intuitively between views	Light		Easy UI navigation reduces the time it takes to learn the app and make using it more efficient
Music player	Scroll through list quickly with A-Z navigation bar	Not in app	Light	Allows the user to quickly search their list of music, reducing the time spent searching
Music player	Move an album to front of recently played albums	Not in app	Adaptable	Allows the user to configure their albums in a way that provides easy access to recently played music

Spotify	Promptness in answering user questions and problems	Software quality		
Music player	Add songs to multiple playlists through checkboxes	Not in app	Light	Allows the user to quickly add music to multiple playlists without having to do it manually for each playlist
Spotify	Choose personal songs	Adaptable		User can create their own music collection
Spotify	Listen to entire albums	Light		Provides the entire album without having to save every song separately
Spotify	Always use the application when necessary	Durable		System availability shows its robustness
Music player	Combine playlists	Not in app	Adaptable	Allows the user to create new playlists based on existing ones
Music player	Add a song to a playlist through the playlist view	Not in app	Light	Provides the user with an existing option but makes it less cumbersome to do so
Music player	Change the order of songs individually	Not in app	Adaptable	Allows the user to configure their order or music exactly how they want to
Spotify	Navigate through the app fluidly	Light		Good UI flow reduces the time it takes to execute certain actions and thus make it less time consuming
Spotify	Edit, remove and move around songs within a playlist	Adaptable		User can edit their music collection to suit their whishes
Spotify	Interact with a beautiful layout	Light		A good UI reduces the time it takes to learn the application and find the view a user is looking for
Music player	Listen to all saved music	Not in app	Complete	Being able to listen to the music you saved makes sense given the option to save music
Music player	Listen to songs the user saved by a certain artist	Not in app	Adaptable	Allows the user to only listen to the music that they saved for the artist, excluding the rest of their saved music
Spotify	Listen to a large number of songs and albums	Complete		More content is a more complete product
Music player	See who follows a playlist	Not in app	Entertaining	Allows the user to see what kind of people like the same music as they do and perhaps find other music this way
Music player	See who follows you	Not in app	Complete	Being able to see who follow you so you can ensure your own privacy

Music player	Get informed about followed artists' recent activity	Not in app	Entertaining	Allows the user to somewhat engage with their favourite artists and being kept up-to-date with new music
Spotify	Save songs	Adaptable		User can create their own music collection
Spotify	Have folders of music	Adaptable		User can create and archive their own music collection
Spotify	Listen to free music	Economical		Doesn't cost the user any money
Music player	Delete a song from a playlist directly while song is playing	Not in app	Light	Provides the user with a feature that already exists but makes it easier to use
Music player	Resume a playlist at previous point	Not in app	Light	Reduces the effort required to restart the point the playlist left off last time
Instagram	Edit photos through filters and tools	Adaptable		Provides the user with the option to tailer photos exactly how they want
Image sharing app	Edit photos with a Bokeh option	Not in app	Adaptable	Allows the user to further edit photos to their liking
Instagram	Tag people in posts	Entertaining		Helps engage other users in the app through tagging
Instagram	Livestream video	Complete		Provides the user with more features that allow them to do more things
Instagram	Add different types of posts	Adaptable		Allows the user to set up their posts exactly how they want
Instagram	Direct message others	Complete		Makes the app more complete and perhaps a total package
Instagram	Set restrictions for who views your posts	Adaptable		Allows the user to configure the app exactly how they want in terms of privacy settings
Instagram	Add links to your profile	Adaptable		Allows the user to configure their profile as they want
Instagram	Add information about yourself to your profile	Adaptable		Allows the user to configure their profile as they want
Instagram	Comment on posts	Complete		Provides the user with a way to communicate with others on posts
Instagram	Delete comments	Adaptable		Allows the user to delete unwanted content
Instagram	Block and report other users	Durable		Provides a manner of self-policing within the application and make it a more enjoyable and safe experience for everyone
Instagram	Translate posts to other languages	Complete		Provides the user with an option to view posts that aren't in their native language
Instagram	Save images	Complete		Provides the user with an option to save images that wouldn't be available otherwise

Image sharing app	Favourite specific people so they show up at the top of your feed	Not in app	Adaptable	Allows the user to configure their timeline to their liking
Image sharing app	View original sized photos	Not in app	Complete	Provides the user with the option to further use photos in other contexts and work with the orignial sizes
Image sharing app	Search people that viewed your story	Not in app	Entertaining	Provides the user with the option to further engage with the people that view their story
Image sharing app	View comments in chronological order	Not in app	Complete	Displaying comments in chronological order seems like something very basic
Image sharing app	Post videos with a bigger time frame	Not in app	Complete	Allows the user to post longer videos
Image sharing app	Turn off read confirmations	Not in app	Adaptable	Allows the user to configure their messaging experience
Image sharing app	Receive a notification of how many people viewed your story	Not in app	Entertaining	Seeing a story's reach might engage the user in creating more stories
Image sharing app	View posts in chronological order	Not in app	Complete	Displaying posts in chronological order seems like something very basic
Image sharing app	Delete a single picture from post without deleting entire post	Not in app	Adaptable	Allows the user to edit their posts to their liking, even after it has been posted
Image sharing app	Preview selected pictures from private accounts	Not in app	Complete	Allows the user to determine if they want to request a follow for a private account
Instagram	Create posts to share with others	Complete		Core feature of the application
Instagram	Follow other users	Adaptable		Allows the user to configure what and who they see on their timeline
Instagram	Create a story	Complete		Provides the user with the option to share pictures without having to create a post
Instagram	Like posts	Complete		Provides the user with the option to like a post and share their appreciation
Image sharing app	Change your timeline from suggested order to chronological	Not in app	Adaptable	Allows the user to set their timeline preference to their liking
Instagram	View other people's pictures	Complete		Core feature of the application
Image sharing app	Edit posts	Not in app	Adaptable	Allows the user to edit their posts to their liking
Image sharing app	Edit picture captions	Not in app	Complete	Allows the user to edit their pictures when modifying them

Video streaming app	Setup a teen account that blocks certain shows	Not in app	Durable	Allows parents to protect their children, making the app more user friendly for families
Netflix	Download videos for watching without WiFi	Economical		Allows for the use of the app without using mobile data
Netflix	User multiple user profiles	Adaptable		Allows multiple users to use the application without interfering with each other and still having their own content and recommendations
Netflix	Easily navigate the application	Light		An easy to navigate application makes it easier to learn and more efficient to use
Video streaming app	Show alert when a show or movie will be removed soon	Not in app	Entertaining	Engages the user with the app if they wish to see the show before it is removed
Netflix	Get notified when new movies are available	Complete		More content is a more complete application
Netflix	Setup the application to only show child appropriate content	Adaptable		Allows children to use the app without supervision and protects them against inappropriate content
Netflix	Setup the application to only show adult specific content	Adaptable		Allows for filtering out content for children
Video streaming app	Rate content with stars	Not in app	Entertaining	Allows the user to influence the type of shows that will be recommended to them
Video streaming app	Remove a show from the continue watching list	Not in app	Adaptable	Allows the user to modify their watch list when they are no longer interested in a certain show
Video streaming app	Change the default language	Not in app	Adaptable	Allows the user to configure the language to their liking
Netflix	Preview content through short descriptions	Entertaining		Helps users understand if they would like new content and draws them in
Video streaming app	Skip video forward for a set number of seconds	Not in app	Adaptable	Allows the user to scroll through the video to the point from which they want to view the episode
Video streaming app	Fill in a personal recommendation questionnaire	Not in app	Entertaining	Allows the app to instantly suggest shows the user might find interesting and wants to watch
Video streaming app	Preview content through short descriptions	Not in app	Entertaining	The short descriptions provide the user with the ability to quickly decide if they would like to watch the show and perhaps expose them to shows they would normally not watch
Video streaming app	Set a timer for a continuous watching period	Not in app	Adaptable	Allows the user to configure their own watch time before having to confirm they are still watching
Netflix	View a large amount of shows	Complete		More content is a more complete application

Video streaming app	Shuffle content to view a random show or movie	Not in app	Entertaining	Exposes the user to new shows or movies
Netflix	Use the application on tablets and phones	Complete		Multiple platforms improve overall availability and use in different contexts
Video streaming app	Show an alert when saved shows have new content available	Not in app	Light	Removes the tasks of checking for new content repeatedly
YouTube	Use the application when necessary	Durable		An app that is always available to be used is more likely to remain being used
YouTube	Sort the watch later playlist	Light		Sorting the playlist makes it easier for the user to review their saved content
YouTube	Rewind or fast forward by double tapping on either side of the video player	Adaptable		Allows the user to navigate a video precisely
Video streaming app	Filter out clips of movies to prevent spoilers	Not in app	Durable	Preventing spoilers allows the user to view more content they would otherwise no longer watch as they have already seen the end, so the app would be used more and longer
Video streaming app	Go back to previously watched video	Not in app	Complete	Simple feature that most apps with some sort of navigation have
Instant messaging app	Remove contacts from recently contacted list while sharing media	Not in app	Adaptable	Allows the user to configure their recent contacted list as they prefer
Instant messaging app	Mute group chat notifications indefinitely	Not in app	Complete	Feature makes sense if muting group chat notifications are already available for shorter time periods
Instant messaging app	Revoke admin rights in a group chat	Not in app	Complete	Feature makes sense if giving someone admin rights is already a feature
Instant messaging app	Draw a message	Not in app	Complete	Another type of message creation
WhatsApp	Back up conversations to the cloud	Durable		Backups makes the app more robust as it allows you to restore your data easily
Instant messaging app	Set individual chat wallpapers	Not in app	Adaptable	Allows the user to configure each chat as they want to
WhatsApp	Use the application without blocking other communication options	Adaptable		Allows the user to keep using other phone functions while using the application
Instant messaging app	Fast forward or rewind voice messages	Not in app	Light	Provides the user to listen to only parts of the voice message, reducing the time required to listen to it
WhatsApp	Make payments to contacts	Complete		More features
Instant messaging app	Lock the app with a password	Not in app	Durable	Makes the app more secure and protects the user's privacy
Instant messaging app	Disable the camera sound in settings	Not in app	Adaptable	Setting so the user can configure their app to their liking
WhatsApp	See when texts are delivered and read	Complete		More features

WhatsApp	Replay voice messages	Complete		More features
WhatsApp	Create stories to share with contacts	Entertaining		Allows the user to create interesting content to share with their contacts
WhatsApp	Text, call and send pictures for free	Economical		Saves phone plan costs
WhatsApp	Send a large number of pictures at once	Light		Allows the user to send all the pictures they want without having to break them up
WhatsApp	Interact with a simple and intuitive user interface	Light		A simple UI makes the application easier to learn and more efficient to use
WhatsApp	Encrypt messages	Durable		Reduces the possibility of private messages leaking and thus making the whole application more solid and safe
Instant messaging app	Use the application on iPad	Not in app	Complete	Availability on multiple platforms makes the app completer and more usable in different contexts
Instant messaging app	Follow other people and keep up with their status updates	Not in app	Entertaining	Following other people engages the user with others and makes the app more fun to use
Instant messaging app	Separate group chats in a dedicated tab	Not in app	Light	Makes it easier and quicker to find group chats
WhatsApp	Send video messages	Complete		More features
WhatsApp	Send stickers	Complete		More features
WhatsApp	Create group video chats	Complete		More features
WhatsApp	Make calls	Complete		More features
WhatsApp	Make video calls	Complete		More features

Instant messaging app	Deliver messages at a scheduled time	Not in app	Adaptable	Allows the user to configure the app to their liking
WhatsApp	Send texts	Complete		More features
WhatsApp	Make calls	Complete		More features
WhatsApp	Make video calls	Complete		More features
Instant messaging app	Open message threads on the phone's lock screen and reply to them	Not in app	Light	Reduces the time it takes to reply to messages as you don't have to unlock your screen and navigate to the conversation
WhatsApp	Use cellular data to bypass network fees	Economical		Saves phone plan costs as cellular data is cheaper than other network fees
WhatsApp	Send voice messages	Complete		More features
Instant messaging app	Create events and invite people in a group chat	Not in app	Complete	Makes the app more complete
Instant messaging app	Link account to email so it can be retrieved when your number changes	Not in app	Durable	Makes your account retrievable and thus the whole experience more durable as the user's situation changes
Instant messaging app	Create subgroups from group chat	Not in app	Complete	Makes the app more complete
Image sharing app	Setup the app to improve accessibility for the elderly	Not in app	Light	Makes the app easier to use for the elderly, reducing the time it takes to learn the app
SnapChat	Share photos and videos with friends	Complete		Base functionality of the app
SnapChat	Use filters to change photos and videos	Adaptable		Allows the user to modify their photos and videos to their liking
SnapChat	Put text on photos and videos	Adaptable		Allows the user to modify their photos and videos to their liking
SnapChat	Add stickers to photos and videos	Adaptable		Allows the user to modify their photos and videos to their liking
SnapChat	Save messages to be read later	Adaptable		Allows the user to configure the messaging to their liking (The review mentions this feature as if it isn't in the application but it is)

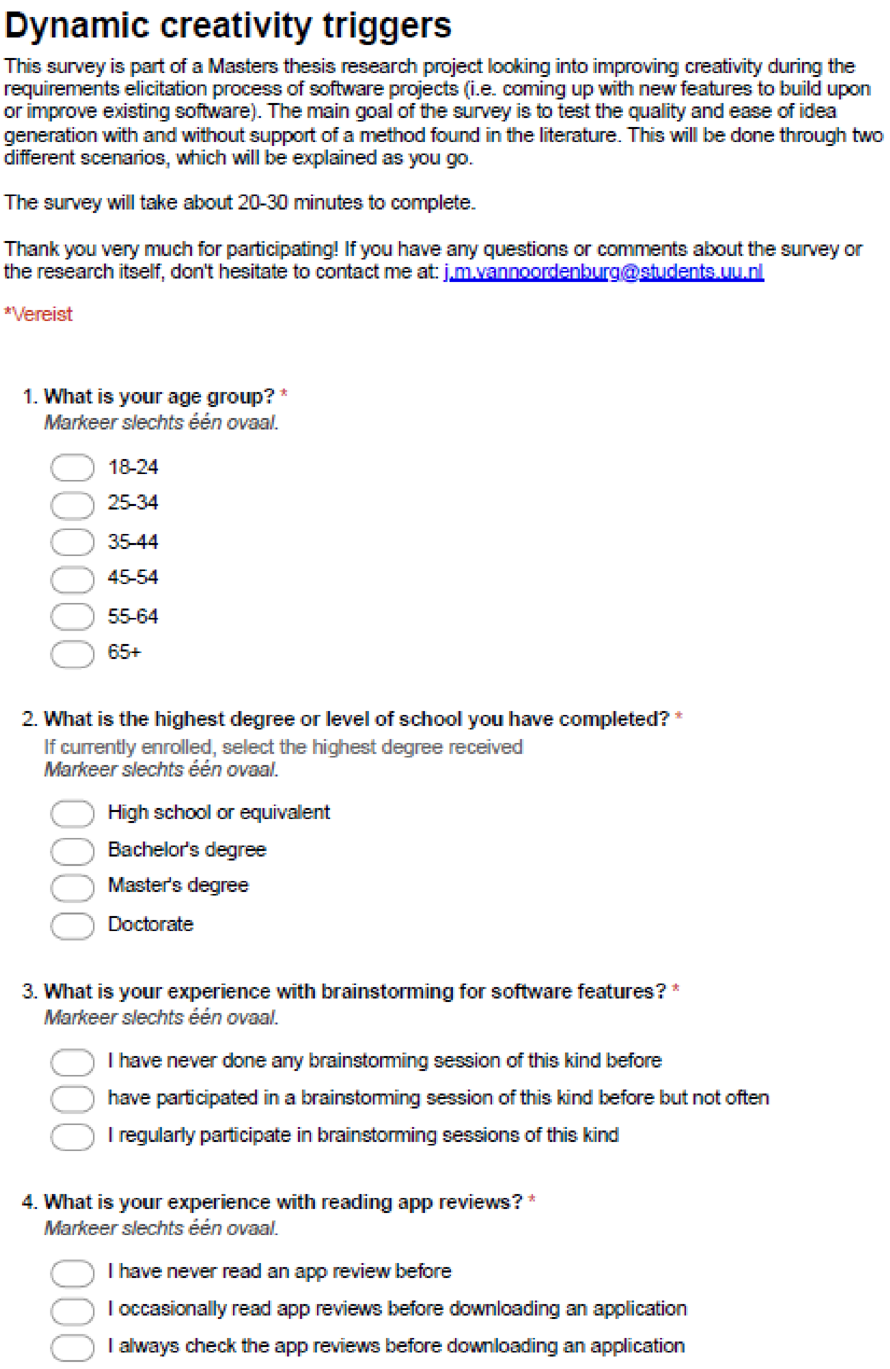
SnapChat	Interact with a easy to use and appealing user interface	Light		A UI that is easy to use and appealing helps users understand the application more quickly and be more efficient in using it
Image sharing app	See history of viewed posts	Not in app	Complete	Makes the app more complete
SnapChat	Videocall and text people using WiFi	Economical		Saves phone plan costs by utilising other available connections
SnapChat	Block people	Adaptable		Allows the user to only communicate with people they want to
Image sharing app	Remove people from a group chat	Not in app	Complete	Seems a logical and complete if adding a user to a group chat is also a feature
Image sharing app	Block all communications when blocking someone	Not in app	Durable	Helps users to protect themselves and their privacy
SnapChat	See saved memories	Complete		Provides the user with an overview of the memories they have saved, which is a logical feature after being able to save a memory
Image sharing app	Mute notifications from lock screen	Not in app	Light	Makes a feature that is already available in the app easier to access and use, without having to direct all attention to it
SnapChat	Create personal filters	Adaptable		Allows the user to create filters to customize their photos and videos
SnapChat	Save images	Adaptable		Allows the user to save images they'd like to keep
SnapChat	Call other people	Complete		Application base feature
SnapChat	Text other people	Complete		Application base feature
SnapChat	Edit your avatar to your personal preference	Adaptable		Allows the user to customize their avatar to their liking
SnapChat	Search filters from local artists	Entertaining		Introduces the user to local art and encourages them to try new filters
SnapChat	Make group chat calls	Complete		Application base feature
SnapChat	Add a timer to photos and videos for how long they will be seen	Adaptable		Allows the user to configure how long their photos or videos are seen
Image sharing app	Automatically select users you have a streak with when sending a message	Not in app	Light	Reduces the time it takes to send a message as the user no longer has to select all of them seperately
Facebook Messenger	Use the application for free	Economical		No costs associated with using the application
Facebook Messenger	Show logged in devices information	Durable		Allows the user to supervise and control what devices have access to their account
Facebook Messenger	Use colorful messages	Entertaining		Makes the UI experience more fun

Facebook Messenger	Message Facebook friends	Complete		Application base feature
Facebook Messenger	Customize individual chats	Adaptable		Allows the user to configure each chat as they want to
Facebook Messenger	Add custom sound effects	Entertaining		Engages the user and makes the app unique
Facebook Messenger	Message people without sharing your phone number	Complete		Application base feature
Facebook Messenger	Video call with good quality	Entertaining		Good quality video calls helps engage the user and make the app more fun to use
Instant messaging app	Unsend a message	Not in app	Complete	Makes the app more complete and perhaps a total package
Instant messaging app	Choose the notification sound	Not in app	Adaptable	Allows the user to configure the app to their liking
Instant messaging app	Add a background picture	Not in app	Adaptable	Allows the user to configure the app to their liking
Facebook Messenger	Attach articles and pictures	Complete		Application base feature
Facebook Messenger	Use the application quickly and reliably	Durable		A reliable application should be more likely to keep being used
Uber	See information about ride costs and arrival time	Complete		Provides the user with the necessary information to review and confirm a ride
Uber	See driver name, vehicle and previous ratings	Complete		Provides the user with the necessary information to review and confirm a ride
Uber	See driver live location	Complete		Provides the user with a means to determine the driver and spot them
Uber	Call driver without sharing phone number	Light		Makes communicating with driver more efficient and less time consuming
Uber	Schedule a pickup beforehand on a specific time or date	Adaptable		Allows the user to set up a ride ahead of time and plan their day
Uber	Use the application with a straightforward and easy to use user interface	Light		A UI that is straightforward and easy to use allows the user to learn the app more quickly and use it more efficiently
Ride sharing app	Be sure of a ride due to the inability to cancel a ride last minute when close to the pickup location	Not in app	Durable	Protects the users from unnecessary driving and reduces abuse of the app
Ride sharing app	Choose who your driver will be	Not in app	Durable	Allows users to protect themselves by choosing a driver they would feel most comfortable with

Ride sharing app	Change distance to kilometer or miles regardless of current location	Not in app	Adaptable	Allows the user to configure the app to their liking
Ride sharing app	Change currency to preference regardless of current location	Not in app	Adaptable	Allows the user to configure the app to their liking
Ride sharing app	Indicate to a driver when a you are in a rush	Not in app	Complete	Makes the app more complete
Gmail	Sort your inbox	Light		Sorting the inbox allows for a less time-consuming experience
Gmail	Send emails	Complete		Application base feature
Gmail	Receive emails	Complete		Application base feature
Gmail	Undo any action	Durable		Allows for the user to make mistakes without actually deleting or changing anything significant
Gmail	Archive emails without deleting them	Adaptable		Allows users to clean up their inbox without removing emails they might like to keep
Gmail	Switch between conversation thread and invidual emails	Adaptable		Allows users to configure the email view as they prefer
Gmail	Mark emails as unread	Adaptable		Allows users to mark emails so they can read them later

Mail app	Bundle emails by type automatically	Not in app	Light	Makes the app less time consuming to use as the user can now more easily scan their inbox and filter out priority messages
Gmail	Use a streamlined and efficient user interface	Light		A streamlined and efficient UI is easier to learn and less time consuming to use
Gmail	Switch between accounts easily	Adaptable		Allows the user to use multiple accounts on the same device
Gmail	Delete email from lock screen	Light		Makes deleting unwanted emails more efficient and less timeconsuming
Mail app	Mute all notifications except from certain people	Not in app	Adaptable	Allows the user to configure the app to their liking
Gmail	Use an application that is fast and reliable	Durable		An application that is reliable and fast ensures the sustainability of its userbase
Gmail	View all unread mail through a unified inbox with multiple accounts	Light		Allows the user to view all emails of all accounts at the same time without having to spend time or effort switching accounts (Review says it's not included in the app, but it is)
Gmail	Snooze messages	Adaptable		Allows the user to revisit messages later on without forgetting them (Review says it's not included in the app, but it is)
Mail app	Integrate dropbox for file attachments	Not in app	Complete	Makes the app more complete
Gmail	Disable conversation view in emails	Adaptable		Allows users to configure the email view to their liking
Gmail	Swipe to delete or achrive message	Light		Quick options lower the amount of time it takes to get through all emails
Gmail	Use a colorful user interface	Entertaining		Color usually is more fun to look at than single color applications
Mail app	Configure preview shown on lock screen	Not in app	Adaptable	Allows the user to configure the app to their liking
Gmail	Use the application with quick loading times	Light		Quick loading times make the app less time consuming to use
Gmail	Automatically empty spam and trash folders after 30 days	Light		Removes the tasks of emptying trash and spam manually
Mail app	Select all email through a button	Not in app	Complete	Makes the app more complete

Appendix B – Group A questionnaire



Dynamic creativity triggers

This survey is part of a Masters thesis research project looking into improving creativity during the requirements elicitation process of software projects (i.e. coming up with new features to build upon or improve existing software). The main goal of the survey is to test the quality and ease of idea generation with and without support of a method found in the literature. This will be done through two different scenarios, which will be explained as you go.

The survey will take about 20-30 minutes to complete.

Thank you very much for participating! If you have any questions or comments about the survey or the research itself, don't hesitate to contact me at: j.m.vannoordenburg@students.uu.nl

*Vereist

1. **What is your age group?** *

Markeer slechts één ovaal.

○ 18-24
○ 25-34
○ 35-44
○ 45-54
○ 55-64
○ 65+

2. **What is the highest degree or level of school you have completed?** *

If currently enrolled, select the highest degree received

Markeer slechts één ovaal.

○ High school or equivalent
○ Bachelor's degree
○ Master's degree
○ Doctorate

3. **What is your experience with brainstorming for software features?** *

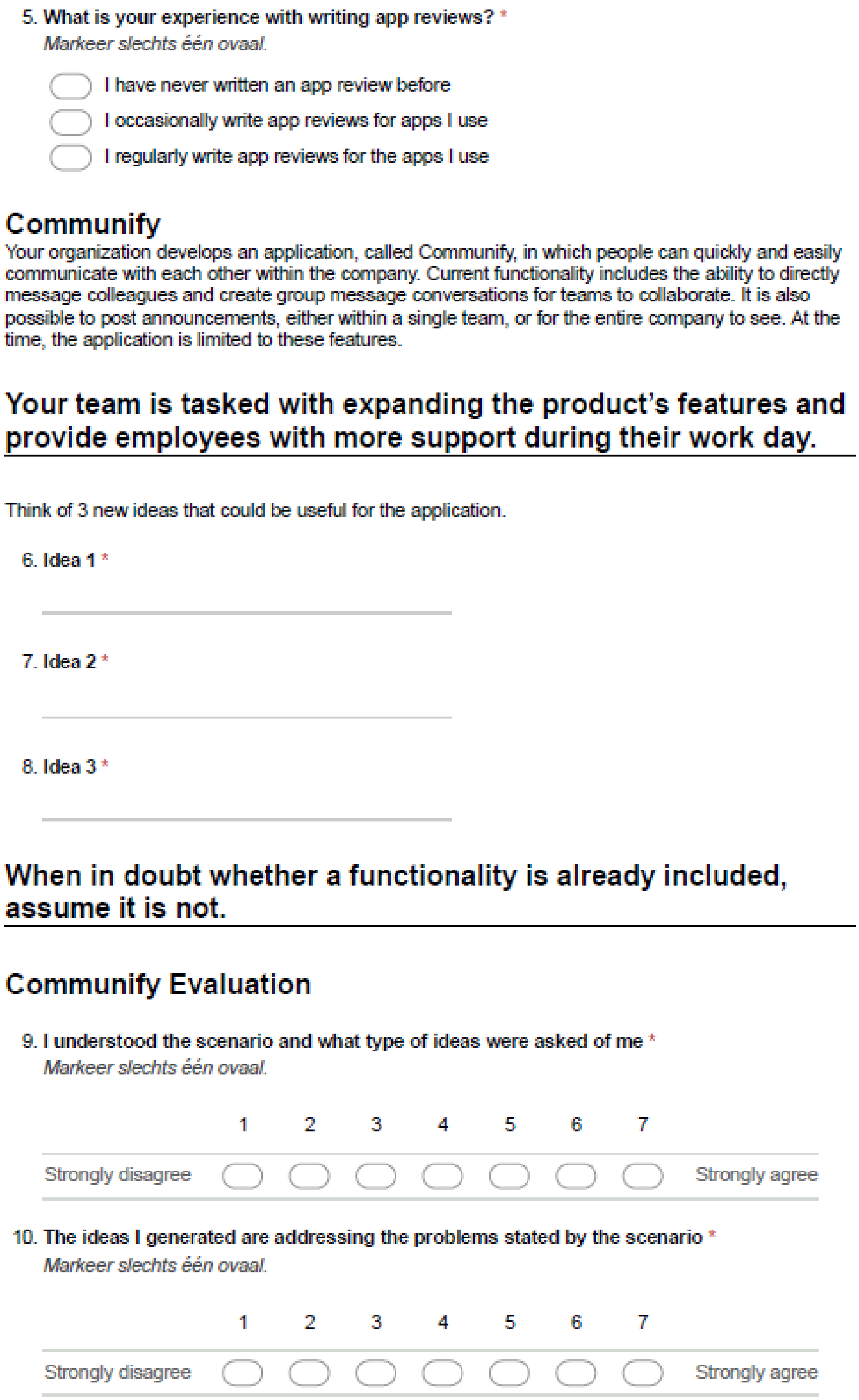
Markeer slechts één ovaal.

○ I have never done any brainstorming session of this kind before
○ have participated in a brainstorming session of this kind before but not often
○ I regularly participate in brainstorming sessions of this kind

4. **What is your experience with reading app reviews?** *

Markeer slechts één ovaal.

○ I have never read an app review before
○ I occasionally read app reviews before downloading an application
○ I always check the app reviews before downloading an application



5. **What is your experience with writing app reviews?** *

Markeer slechts één ovaal.

○ I have never written an app review before
○ I occasionally write app reviews for apps I use
○ I regularly write app reviews for the apps I use

Communify

Your organization develops an application, called Communify, in which people can quickly and easily communicate with each other within the company. Current functionality includes the ability to directly message colleagues and create group message conversations for teams to collaborate. It is also possible to post announcements, either within a single team, or for the entire company to see. At the time, the application is limited to these features.

Your team is tasked with expanding the product's features and provide employees with more support during their work day.

Think of 3 new ideas that could be useful for the application.

6. **Idea 1** *

____

7. **Idea 2** *

____

8. **Idea 3** *

____

When in doubt whether a functionality is already included, assume it is not.

Communify Evaluation

9. **I understood the scenario and what type of ideas were asked of me** *

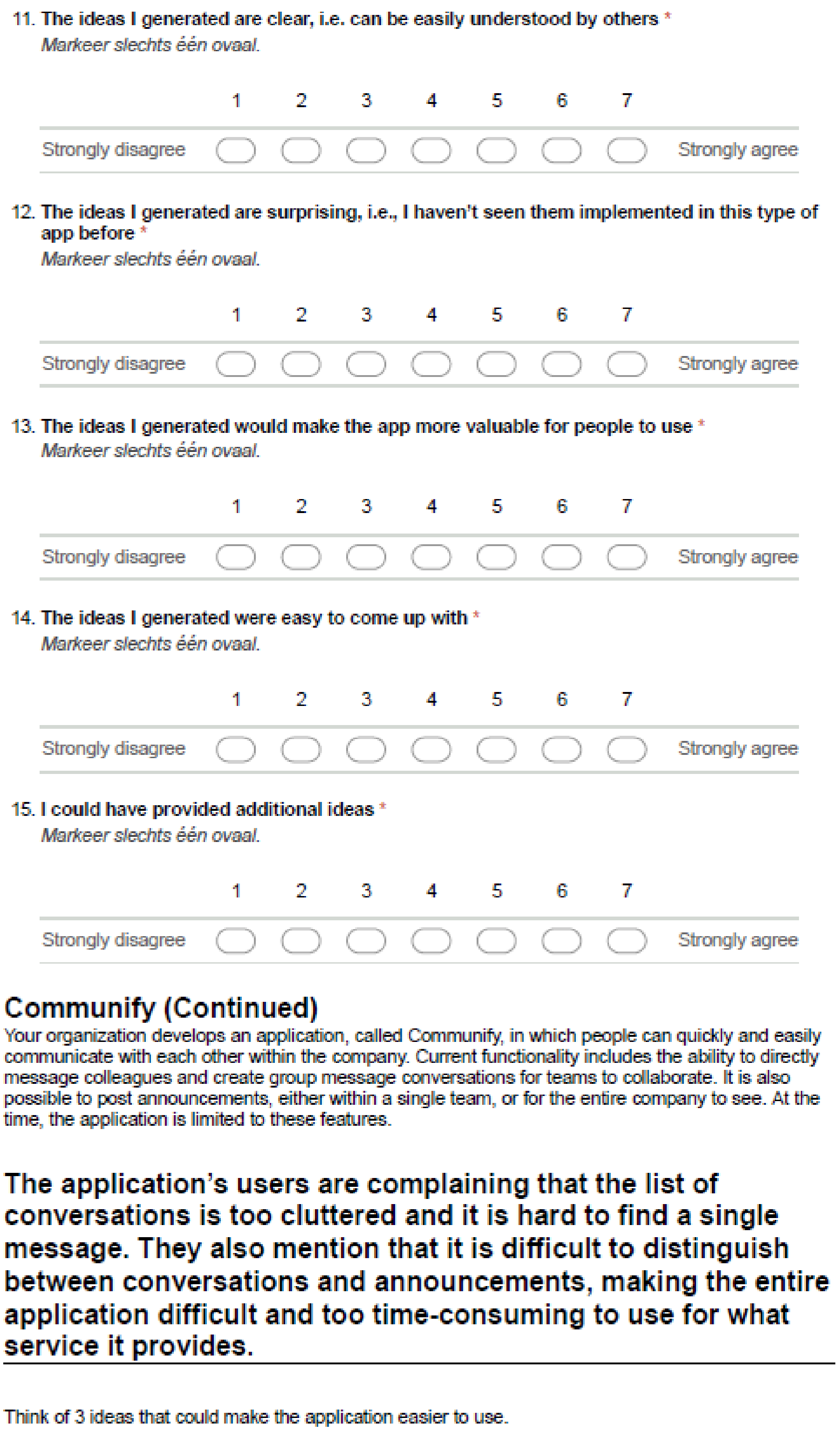
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

10. **The ideas I generated are addressing the problems stated by the scenario** *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree



11. **The ideas I generated are clear, i.e. can be easily understood by others** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

12. **The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

13. **The ideas I generated would make the app more valuable for people to use** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

14. **The ideas I generated were easy to come up with** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

15. **I could have provided additional ideas** *
Markeer slechts één ovaal.

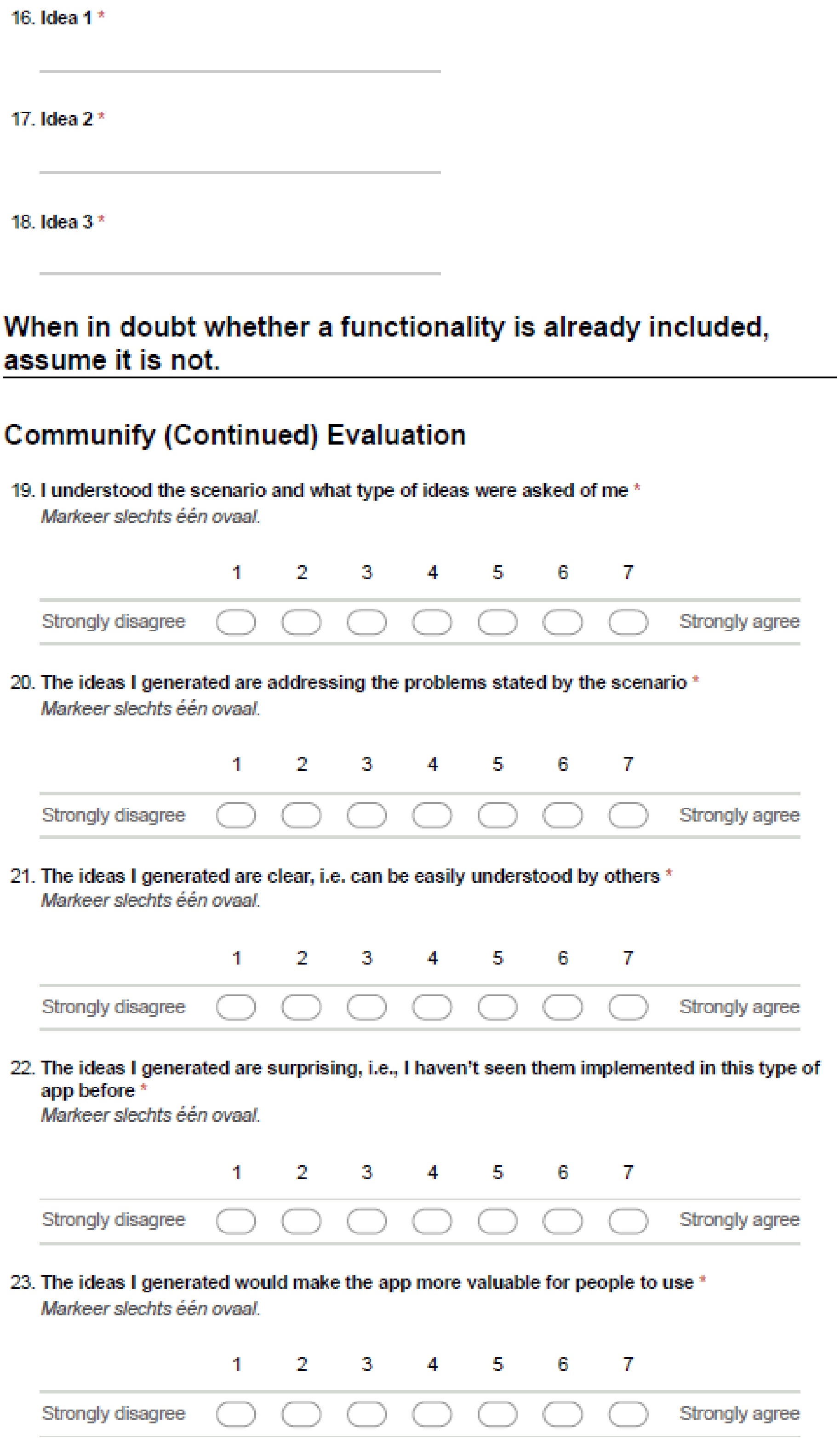
	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

Communify (Continued)

Your organization develops an application, called Communify, in which people can quickly and easily communicate with each other within the company. Current functionality includes the ability to directly message colleagues and create group message conversations for teams to collaborate. It is also possible to post announcements, either within a single team, or for the entire company to see. At the time, the application is limited to these features.

The application's users are complaining that the list of conversations is too cluttered and it is hard to find a single message. They also mention that it is difficult to distinguish between conversations and announcements, making the entire application difficult and too time-consuming to use for what service it provides.

Think of 3 ideas that could make the application easier to use.



16. Idea 1 *

17. Idea 2 *

18. Idea 3 *

When in doubt whether a functionality is already included, assume it is not.

Communify (Continued) Evaluation

19. I understood the scenario and what type of ideas were asked of me *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

20. The ideas I generated are addressing the problems stated by the scenario *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

21. The ideas I generated are clear, i.e. can be easily understood by others *
Markeer slechts één ovaal.

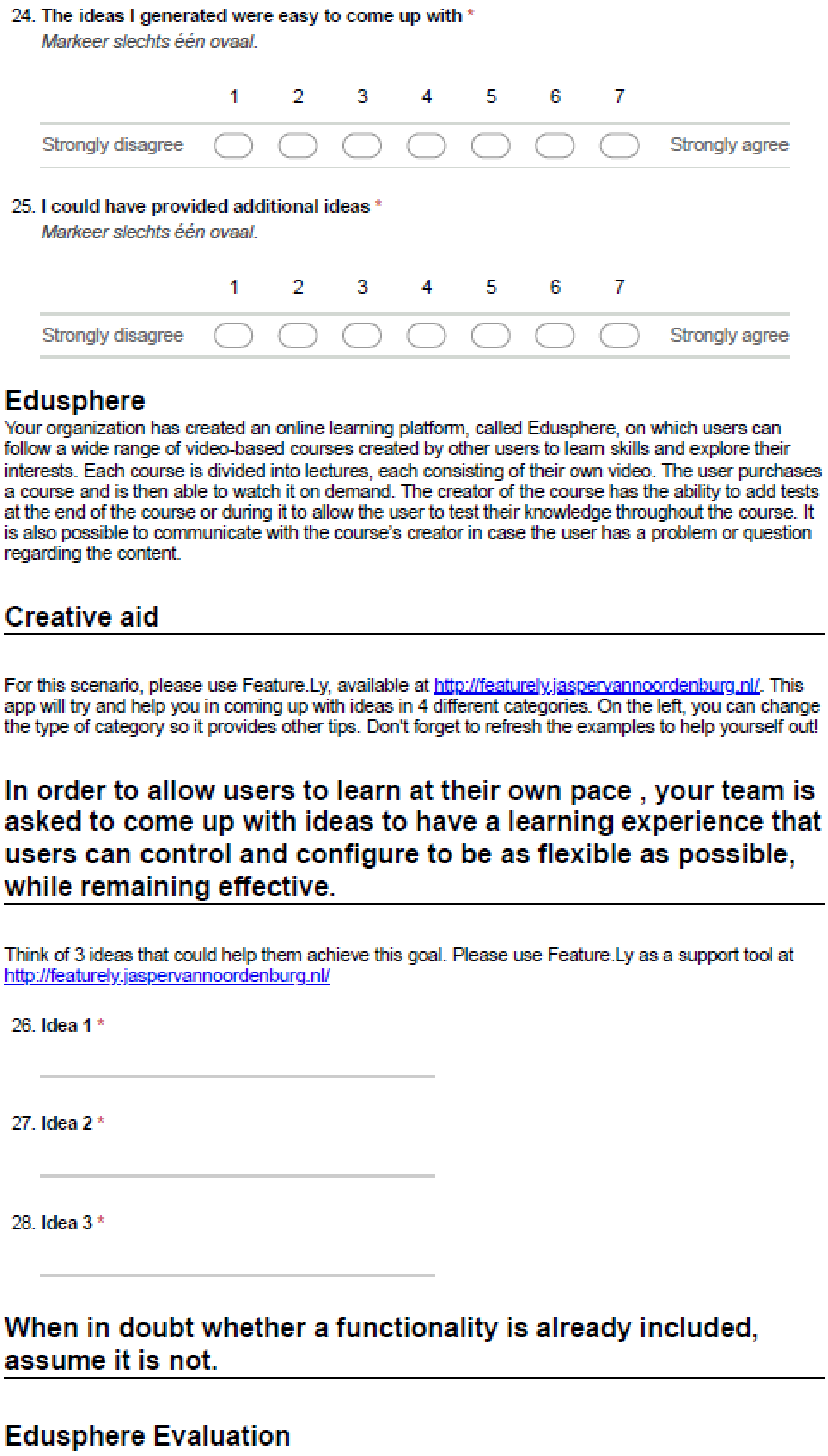
	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

22. The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

23. The ideas I generated would make the app more valuable for people to use *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree



24. **The ideas I generated were easy to come up with** *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

25. **I could have provided additional ideas** *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

Edusphere

Your organization has created an online learning platform, called Edusphere, on which users can follow a wide range of video-based courses created by other users to learn skills and explore their interests. Each course is divided into lectures, each consisting of their own video. The user purchases a course and is then able to watch it on demand. The creator of the course has the ability to add tests at the end of the course or during it to allow the user to test their knowledge throughout the course. It is also possible to communicate with the course's creator in case the user has a problem or question regarding the content.

Creative aid

For this scenario, please use Feature.Ly, available at http://featurely.jaspervannoordenburg.nl/. This app will try and help you in coming up with ideas in 4 different categories. On the left, you can change the type of category so it provides other tips. Don't forget to refresh the examples to help yourself out!

In order to allow users to learn at their own pace , your team is asked to come up with ideas to have a learning experience that users can control and configure to be as flexible as possible, while remaining effective.

Think of 3 ideas that could help them achieve this goal. Please use Feature.Ly as a support tool at http://featurely.jaspervannoordenburg.nl/

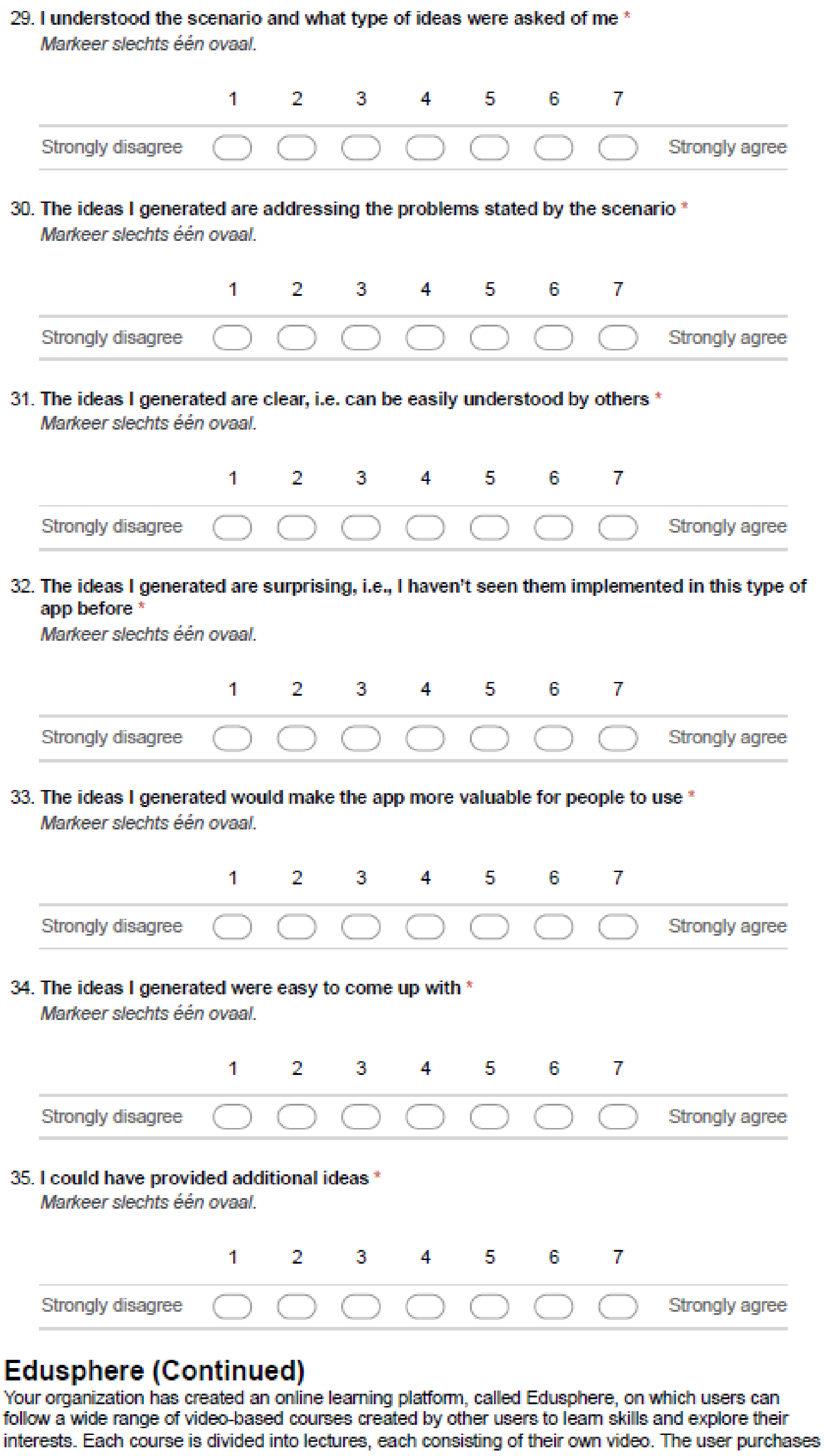
26. **Idea 1** *

27. **Idea 2** *

28. **Idea 3** *

When in doubt whether a functionality is already included, assume it is not.

Edusphere Evaluation



29. **I understood the scenario and what type of ideas were asked of me** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

30. **The ideas I generated are addressing the problems stated by the scenario** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

31. **The ideas I generated are clear, i.e. can be easily understood by others** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

32. **The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

33. **The ideas I generated would make the app more valuable for people to use** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

34. **The ideas I generated were easy to come up with** *
Markeer slechts één ovaal.

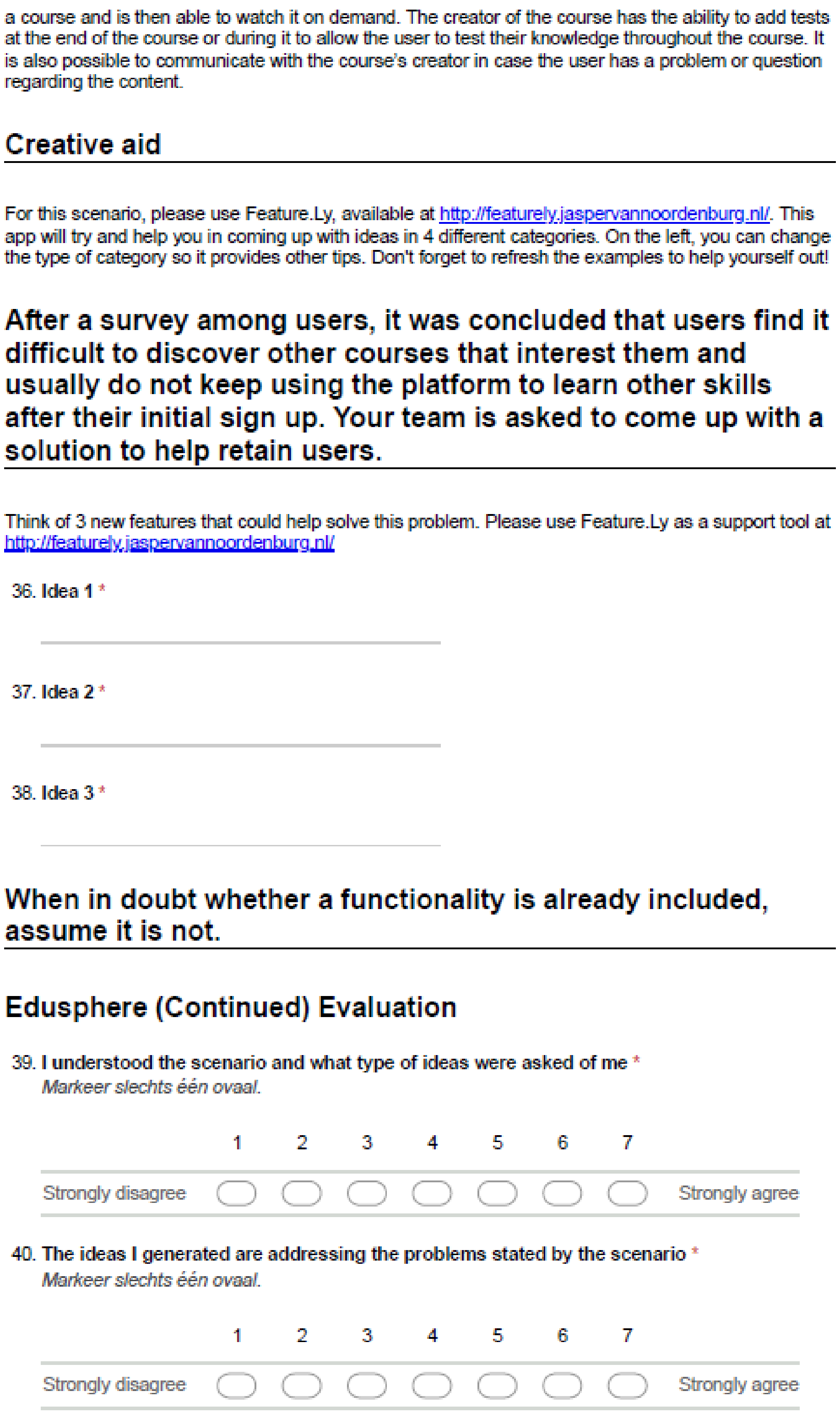
	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

35. **I could have provided additional ideas** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

Edusphere (Continued)

Your organization has created an online learning platform, called Edusphere, on which users can follow a wide range of video-based courses created by other users to learn skills and explore their interests. Each course is divided into lectures, each consisting of their own video. The user purchases



a course and is then able to watch it on demand. The creator of the course has the ability to add tests at the end of the course or during it to allow the user to test their knowledge throughout the course. It is also possible to communicate with the course's creator in case the user has a problem or question regarding the content.

Creative aid

For this scenario, please use Feature.Ly, available at http://featurely.jaspervannoordenburg.nl/. This app will try and help you in coming up with ideas in 4 different categories. On the left, you can change the type of category so it provides other tips. Don't forget to refresh the examples to help yourself out!

After a survey among users, it was concluded that users find it difficult to discover other courses that interest them and usually do not keep using the platform to learn other skills after their initial sign up. Your team is asked to come up with a solution to help retain users.

Think of 3 new features that could help solve this problem. Please use Feature.Ly as a support tool at http://featurely.jaspervannoordenburg.nl/

36. **Idea 1** *

37. **Idea 2** *

38. **Idea 3** *

When in doubt whether a functionality is already included, assume it is not.

Edusphere (Continued) Evaluation

39. **I understood the scenario and what type of ideas were asked of me** *

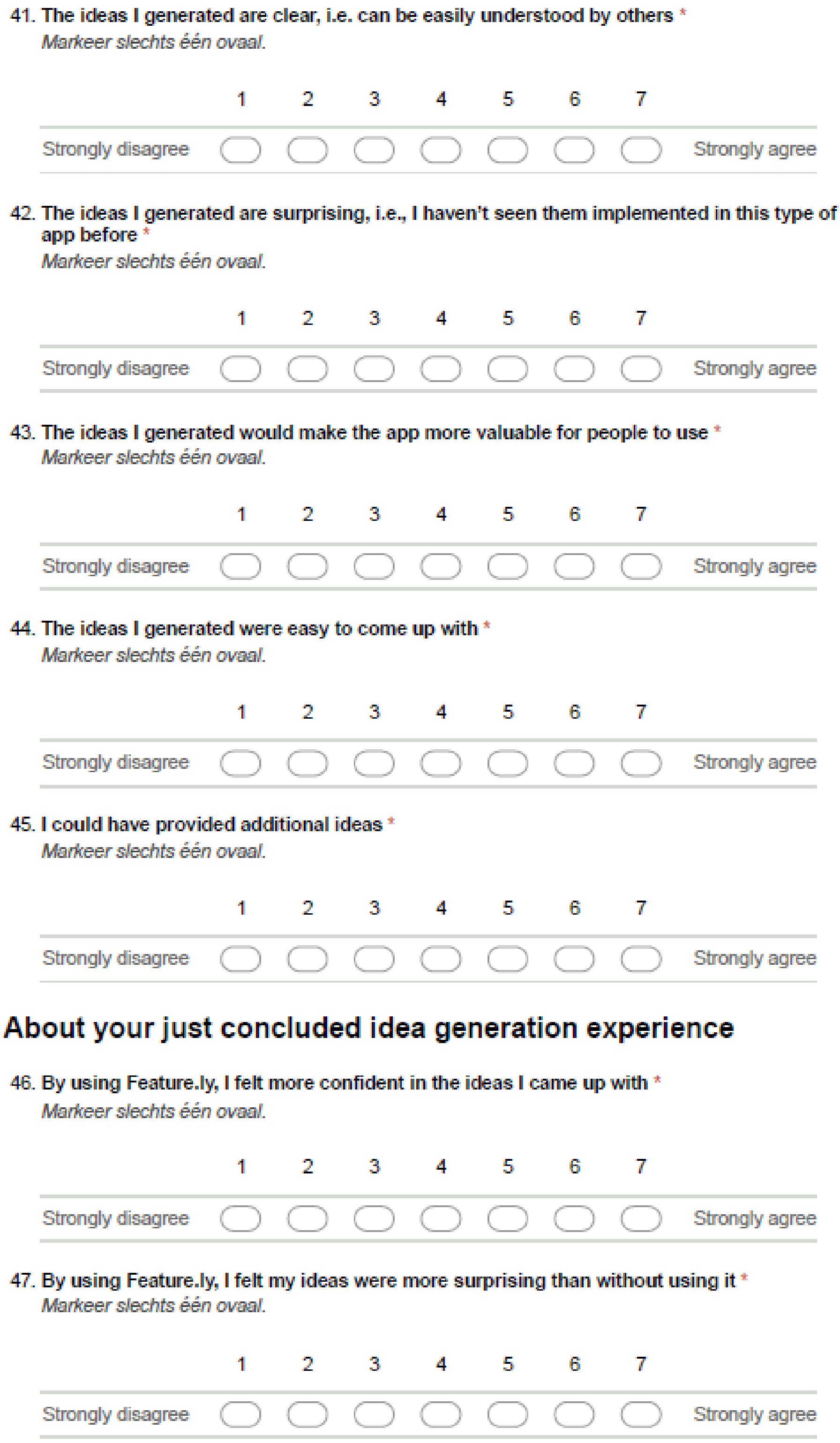
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

40. **The ideas I generated are addressing the problems stated by the scenario** *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree



41. The ideas I generated are clear, i.e. can be easily understood by others *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

42. The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

43. The ideas I generated would make the app more valuable for people to use *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

44. The ideas I generated were easy to come up with *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

45. I could have provided additional ideas *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

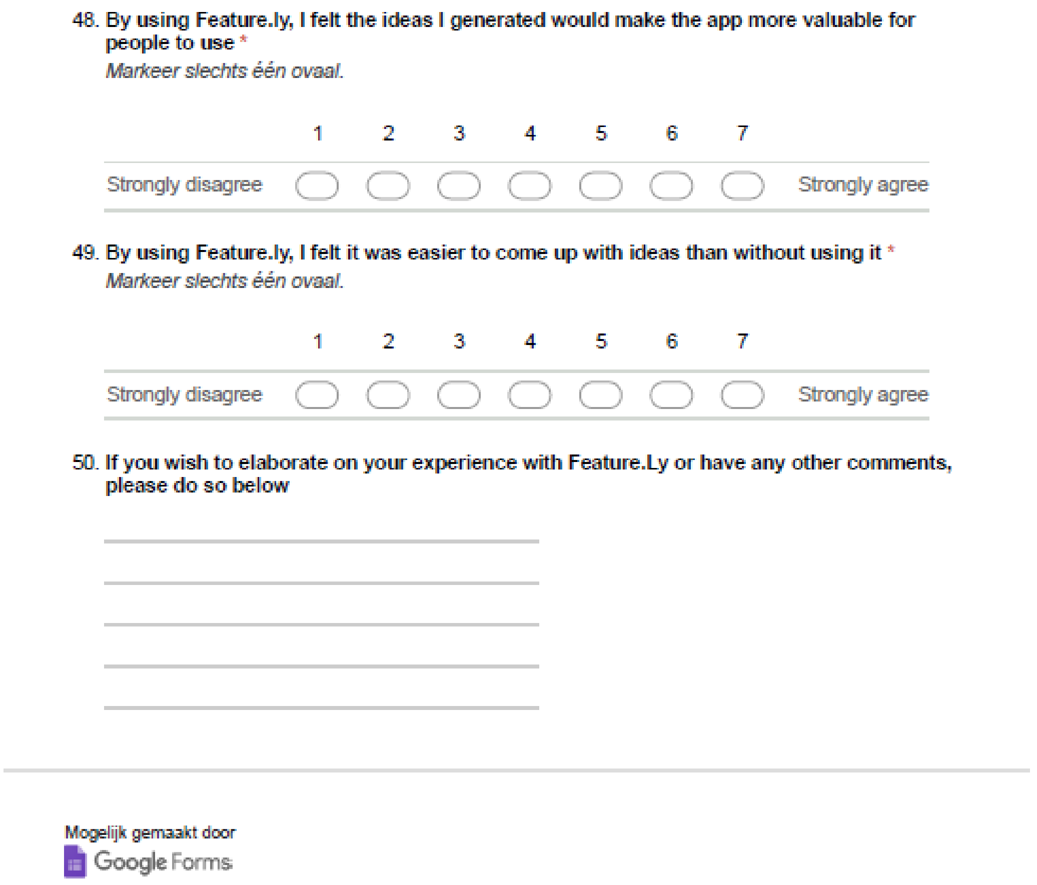
About your just concluded idea generation experience

46. By using Feature.ly, I felt more confident in the ideas I came up with *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

47. By using Feature.ly, I felt my ideas were more surprising than without using it *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree



48. **By using Feature.ly, I felt the ideas I generated would make the app more valuable for people to use** *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

49. **By using Feature.ly, I felt it was easier to come up with ideas than without using it** *

Markeer slechts één ovaal.

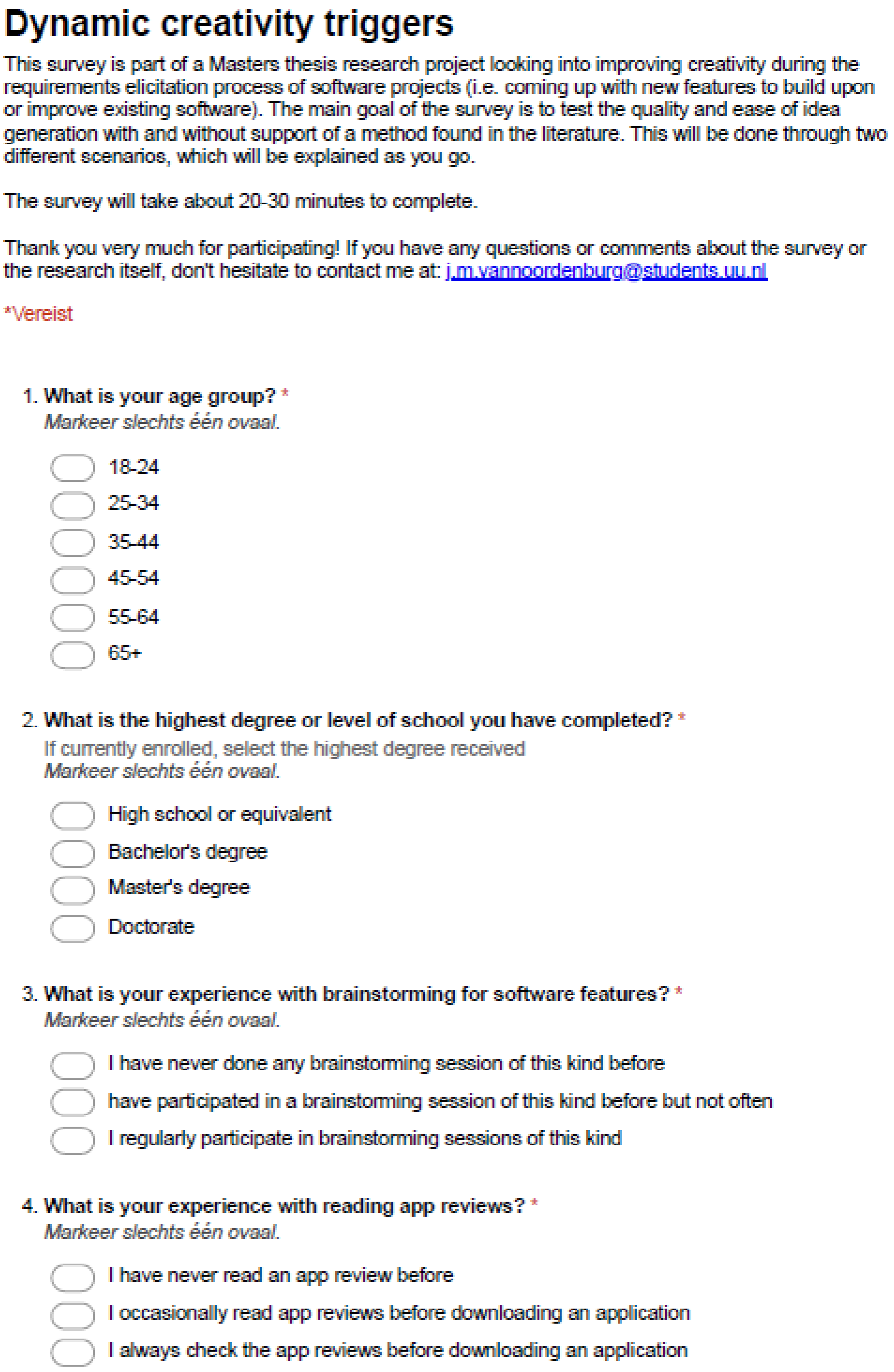
	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

50. **If you wish to elaborate on your experience with Feature.Ly or have any other comments, please do so below**

Mogelijk gemaakt door

Google Forms

Appendix C – Group B questionnaire



Dynamic creativity triggers

This survey is part of a Masters thesis research project looking into improving creativity during the requirements elicitation process of software projects (i.e. coming up with new features to build upon or improve existing software). The main goal of the survey is to test the quality and ease of idea generation with and without support of a method found in the literature. This will be done through two different scenarios, which will be explained as you go.

The survey will take about 20-30 minutes to complete.

Thank you very much for participating! If you have any questions or comments about the survey or the research itself, don't hesitate to contact me at: j.m.vannoordenburg@students.uu.nl

*Vereist

1. **What is your age group?** *
Markeer slechts één ovaal.

○ 18-24
○ 25-34
○ 35-44
○ 45-54
○ 55-64
○ 65+

2. **What is the highest degree or level of school you have completed?** *
If currently enrolled, select the highest degree received
Markeer slechts één ovaal.

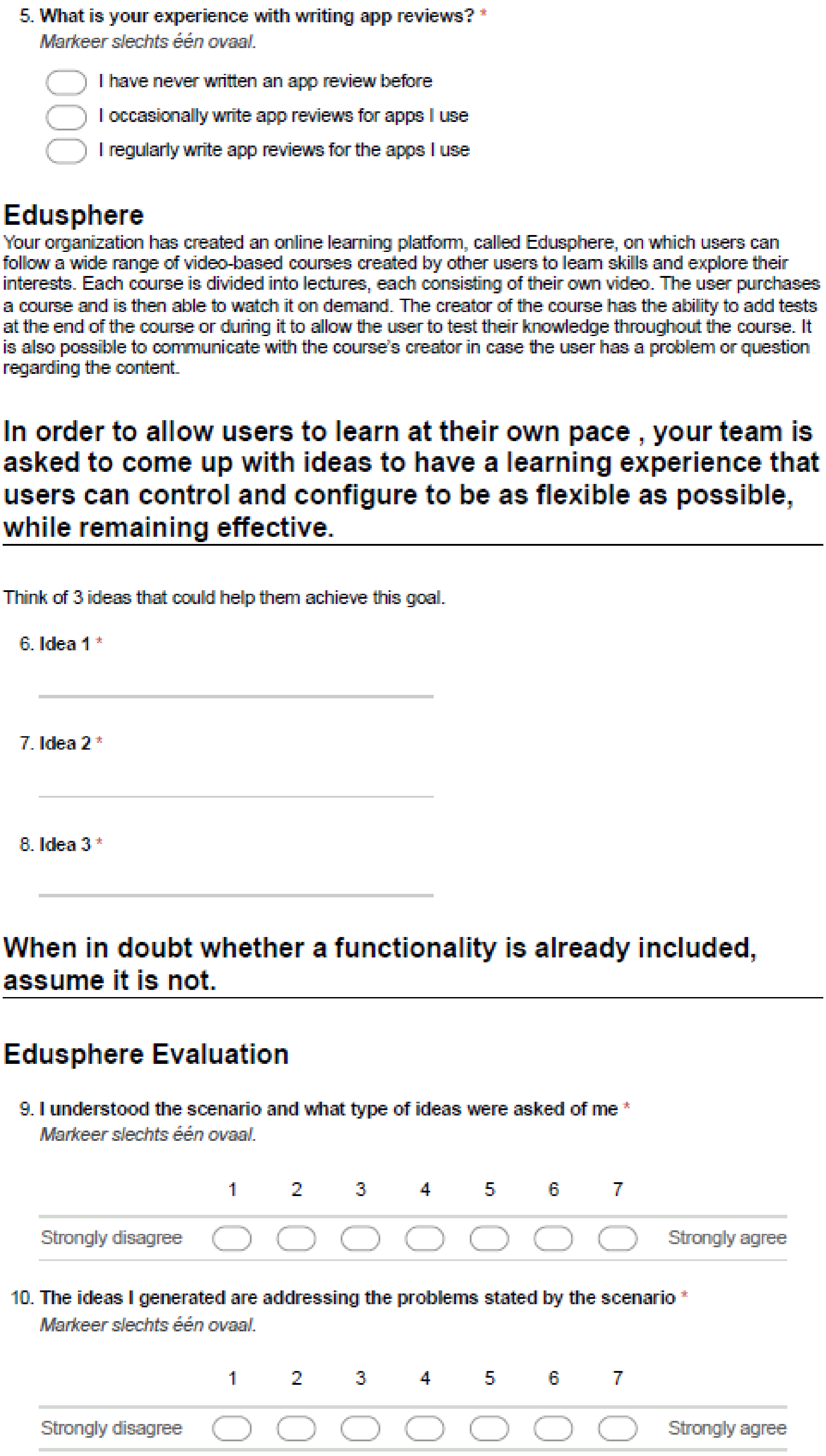
○ High school or equivalent
○ Bachelor's degree
○ Master's degree
○ Doctorate

3. **What is your experience with brainstorming for software features?** *
Markeer slechts één ovaal.

○ I have never done any brainstorming session of this kind before
○ have participated in a brainstorming session of this kind before but not often
○ I regularly participate in brainstorming sessions of this kind

4. **What is your experience with reading app reviews?** *
Markeer slechts één ovaal.

○ I have never read an app review before
○ I occasionally read app reviews before downloading an application
○ I always check the app reviews before downloading an application



5. **What is your experience with writing app reviews?** *

Markeer slechts één ovaal.

○ I have never written an app review before

○ I occasionally write app reviews for apps I use

○ I regularly write app reviews for the apps I use

Edusphere

Your organization has created an online learning platform, called Edusphere, on which users can follow a wide range of video-based courses created by other users to learn skills and explore their interests. Each course is divided into lectures, each consisting of their own video. The user purchases a course and is then able to watch it on demand. The creator of the course has the ability to add tests at the end of the course or during it to allow the user to test their knowledge throughout the course. It is also possible to communicate with the course's creator in case the user has a problem or question regarding the content.

In order to allow users to learn at their own pace , your team is asked to come up with ideas to have a learning experience that users can control and configure to be as flexible as possible, while remaining effective.

Think of 3 ideas that could help them achieve this goal.

6. **Idea 1** *

____

7. **Idea 2** *

____

8. **Idea 3** *

____

When in doubt whether a functionality is already included, assume it is not.

Edusphere Evaluation

9. **I understood the scenario and what type of ideas were asked of me** *

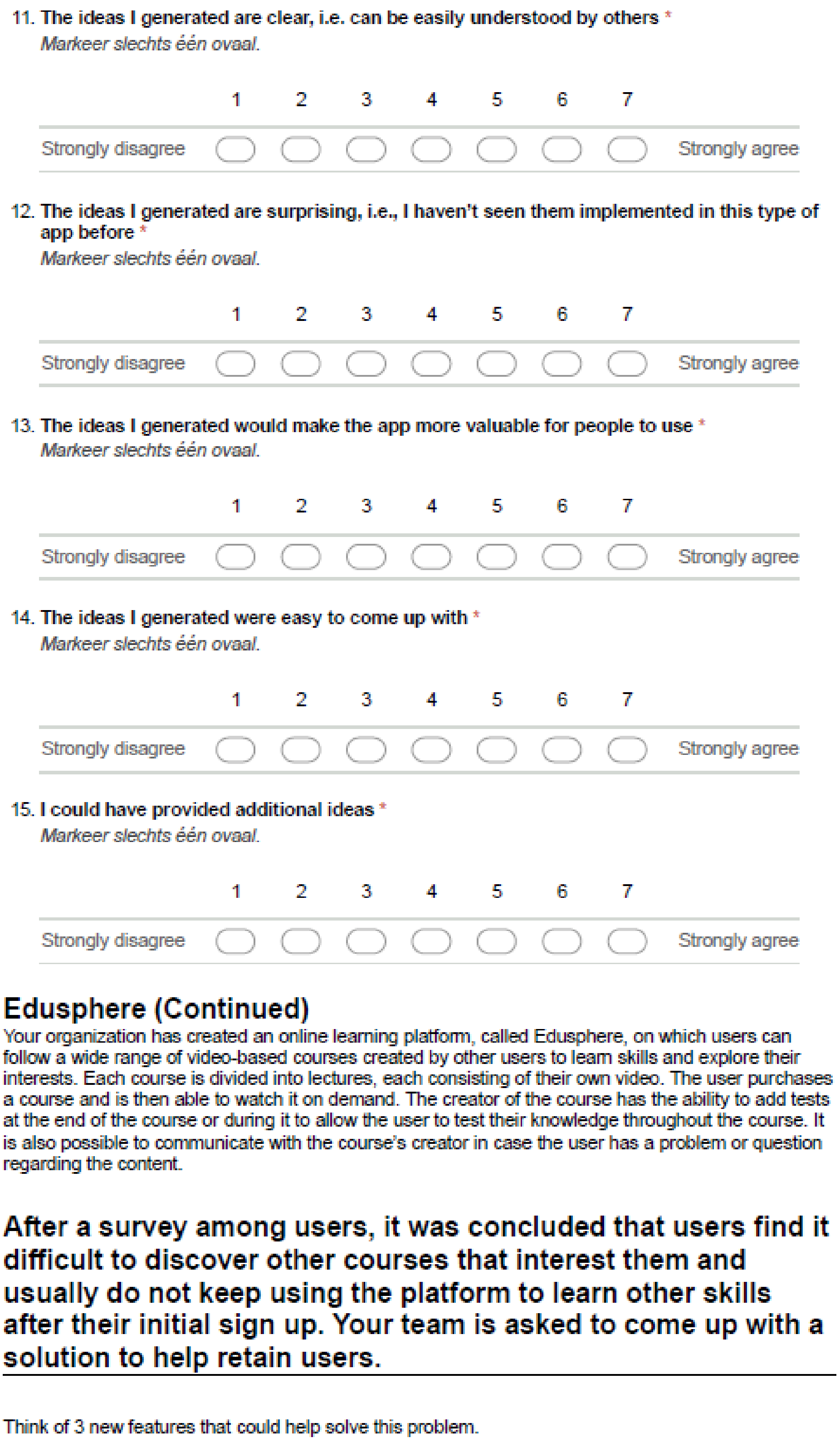
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

10. **The ideas I generated are addressing the problems stated by the scenario** *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree



11. **The ideas I generated are clear, i.e. can be easily understood by others** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

12. **The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

13. **The ideas I generated would make the app more valuable for people to use** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

14. **The ideas I generated were easy to come up with** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

15. **I could have provided additional ideas** *
Markeer slechts één ovaal.

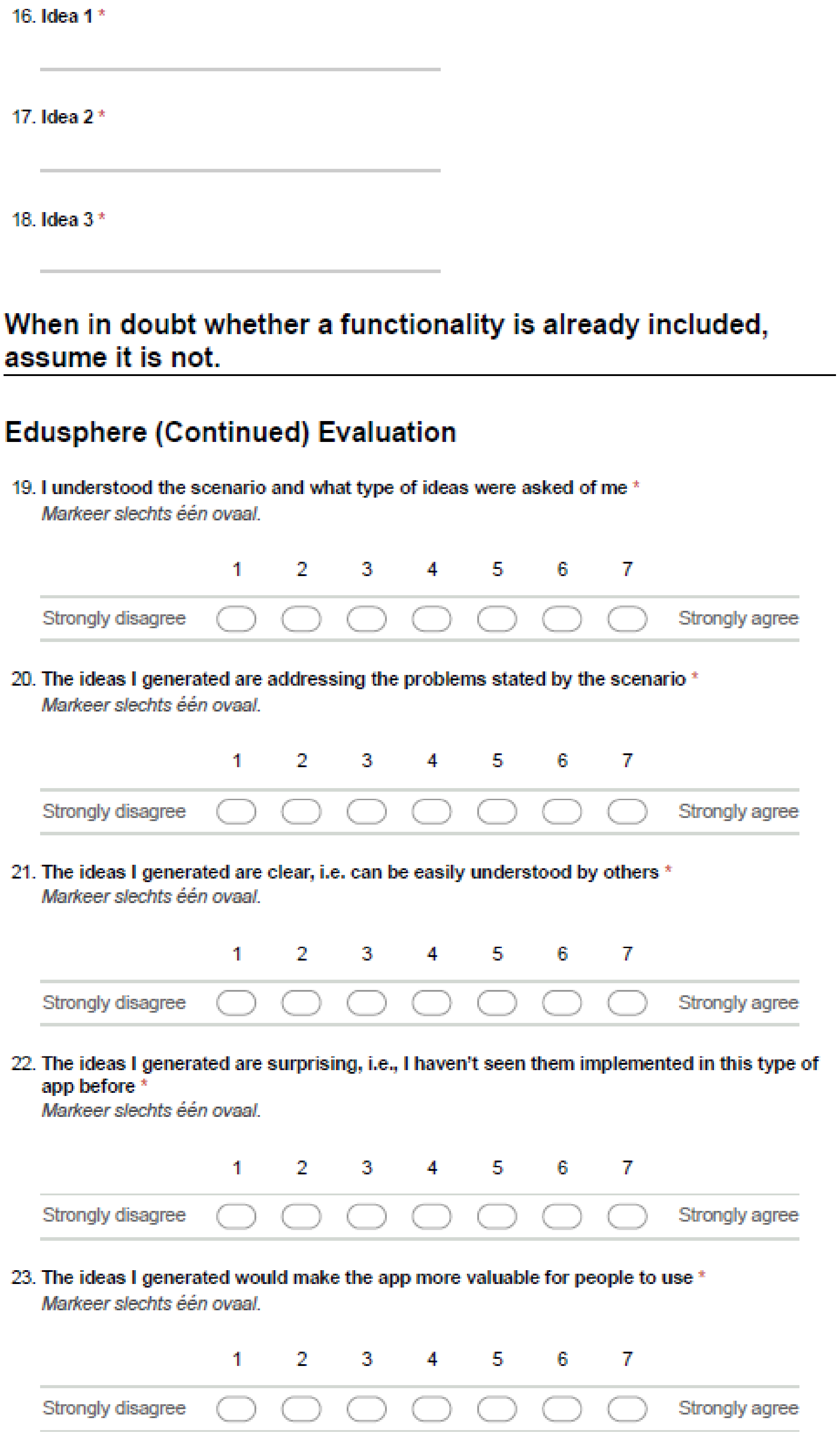
	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

Edusphere (Continued)

Your organization has created an online learning platform, called Edusphere, on which users can follow a wide range of video-based courses created by other users to learn skills and explore their interests. Each course is divided into lectures, each consisting of their own video. The user purchases a course and is then able to watch it on demand. The creator of the course has the ability to add tests at the end of the course or during it to allow the user to test their knowledge throughout the course. It is also possible to communicate with the course's creator in case the user has a problem or question regarding the content.

After a survey among users, it was concluded that users find it difficult to discover other courses that interest them and usually do not keep using the platform to learn other skills after their initial sign up. Your team is asked to come up with a solution to help retain users.

Think of 3 new features that could help solve this problem.



16. Idea 1 *

17. Idea 2 *

18. Idea 3 *

When in doubt whether a functionality is already included, assume it is not.

Edusphere (Continued) Evaluation

19. I understood the scenario and what type of ideas were asked of me *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

20. The ideas I generated are addressing the problems stated by the scenario *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

21. The ideas I generated are clear, i.e. can be easily understood by others *
Markeer slechts één ovaal.

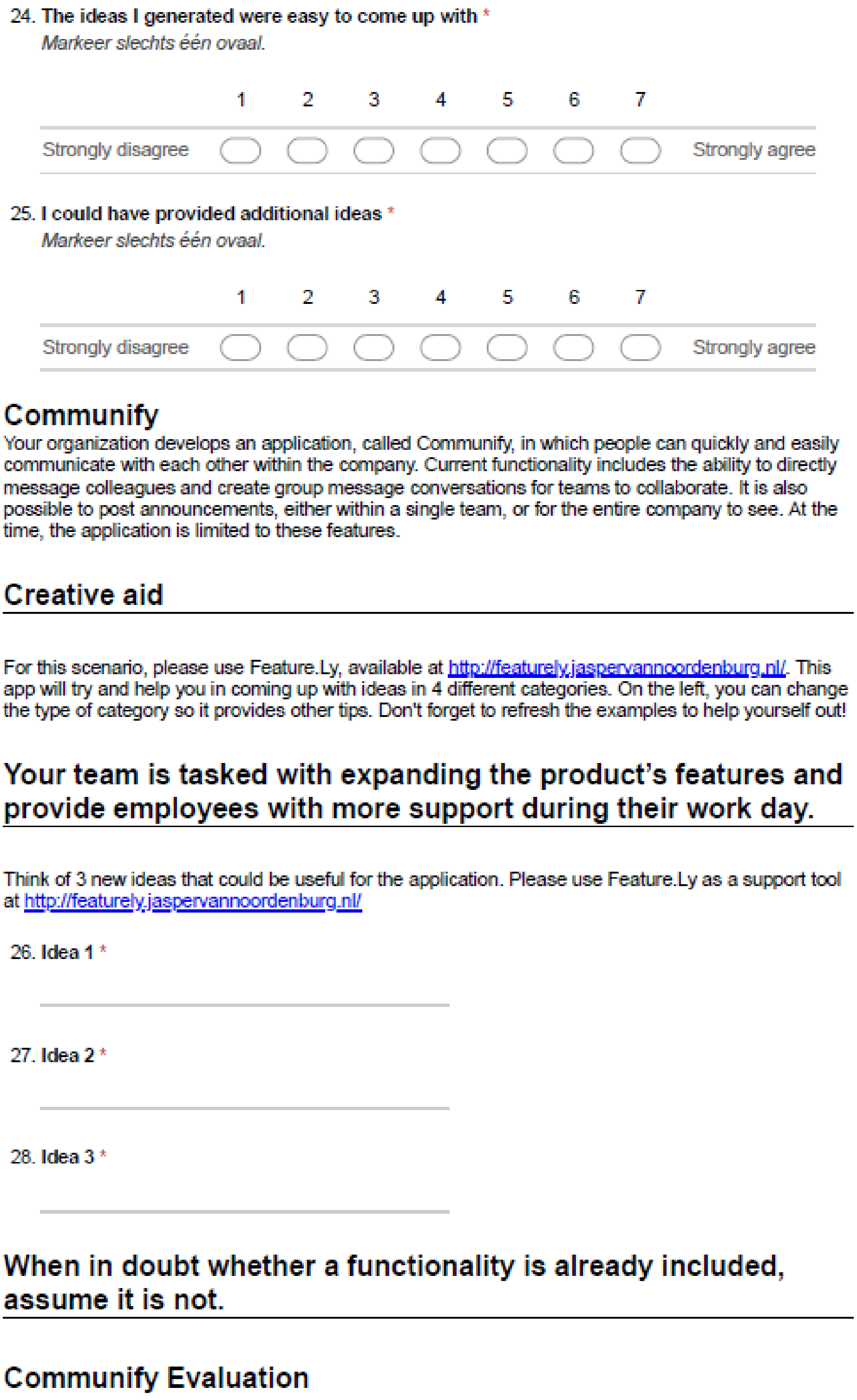
	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

22. The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

23. The ideas I generated would make the app more valuable for people to use *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree



24. **The ideas I generated were easy to come up with** *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

25. **I could have provided additional ideas** *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

Communify

Your organization develops an application, called Communify, in which people can quickly and easily communicate with each other within the company. Current functionality includes the ability to directly message colleagues and create group message conversations for teams to collaborate. It is also possible to post announcements, either within a single team, or for the entire company to see. At the time, the application is limited to these features.

Creative aid

For this scenario, please use Feature.Ly, available at http://featurely.jaspervannoordenburg.nl/. This app will try and help you in coming up with ideas in 4 different categories. On the left, you can change the type of category so it provides other tips. Don't forget to refresh the examples to help yourself out!

Your team is tasked with expanding the product's features and provide employees with more support during their work day.

Think of 3 new ideas that could be useful for the application. Please use Feature.Ly as a support tool at http://featurely.jaspervannoordenburg.nl/

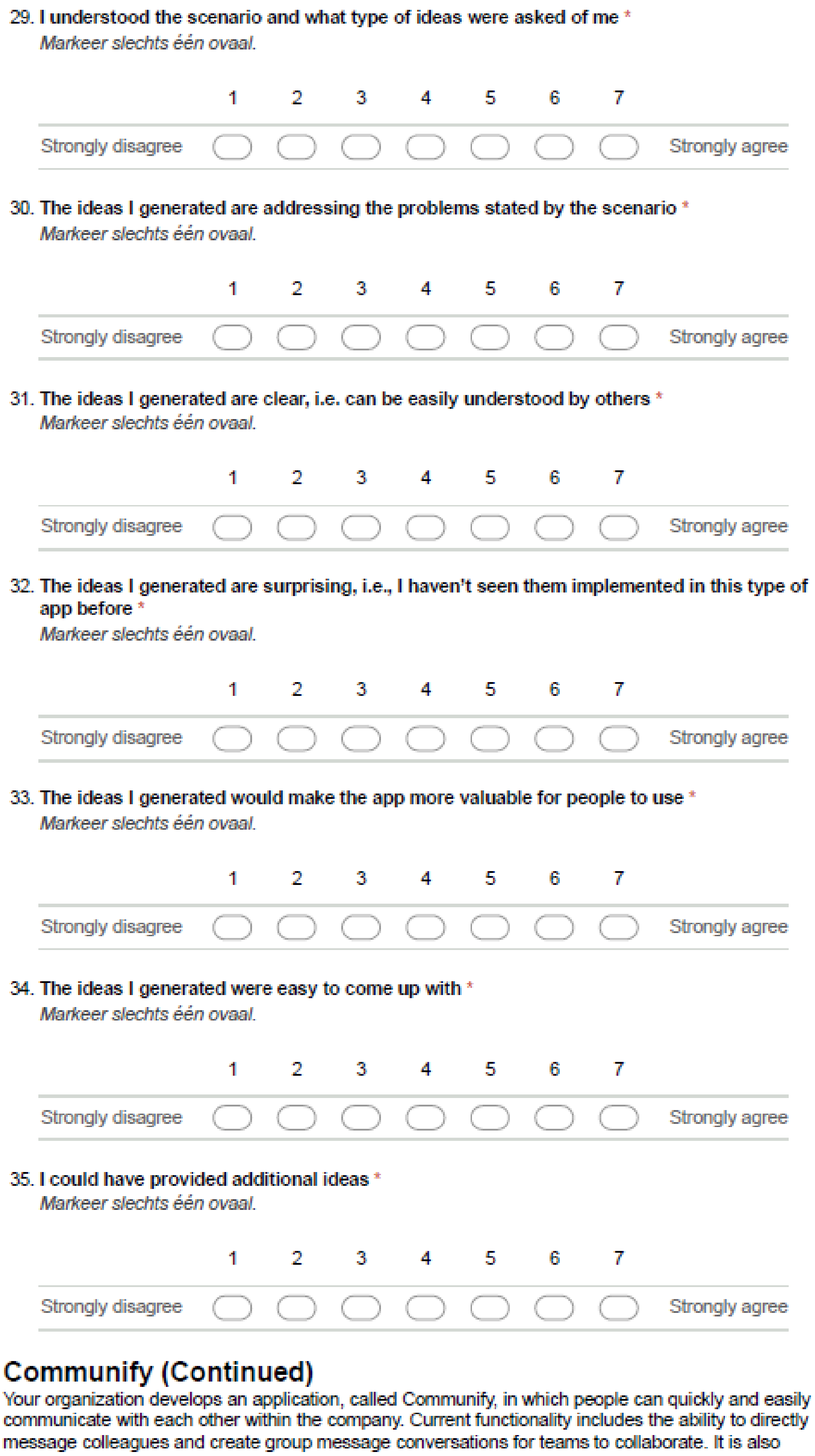
26. **Idea 1** *

27. **Idea 2** *

28. **Idea 3** *

When in doubt whether a functionality is already included, assume it is not.

Communify Evaluation



29. I understood the scenario and what type of ideas were asked of me *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

30. The ideas I generated are addressing the problems stated by the scenario *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

31. The ideas I generated are clear, i.e. can be easily understood by others *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

32. The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

33. The ideas I generated would make the app more valuable for people to use *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

34. The ideas I generated were easy to come up with *
Markeer slechts één ovaal.

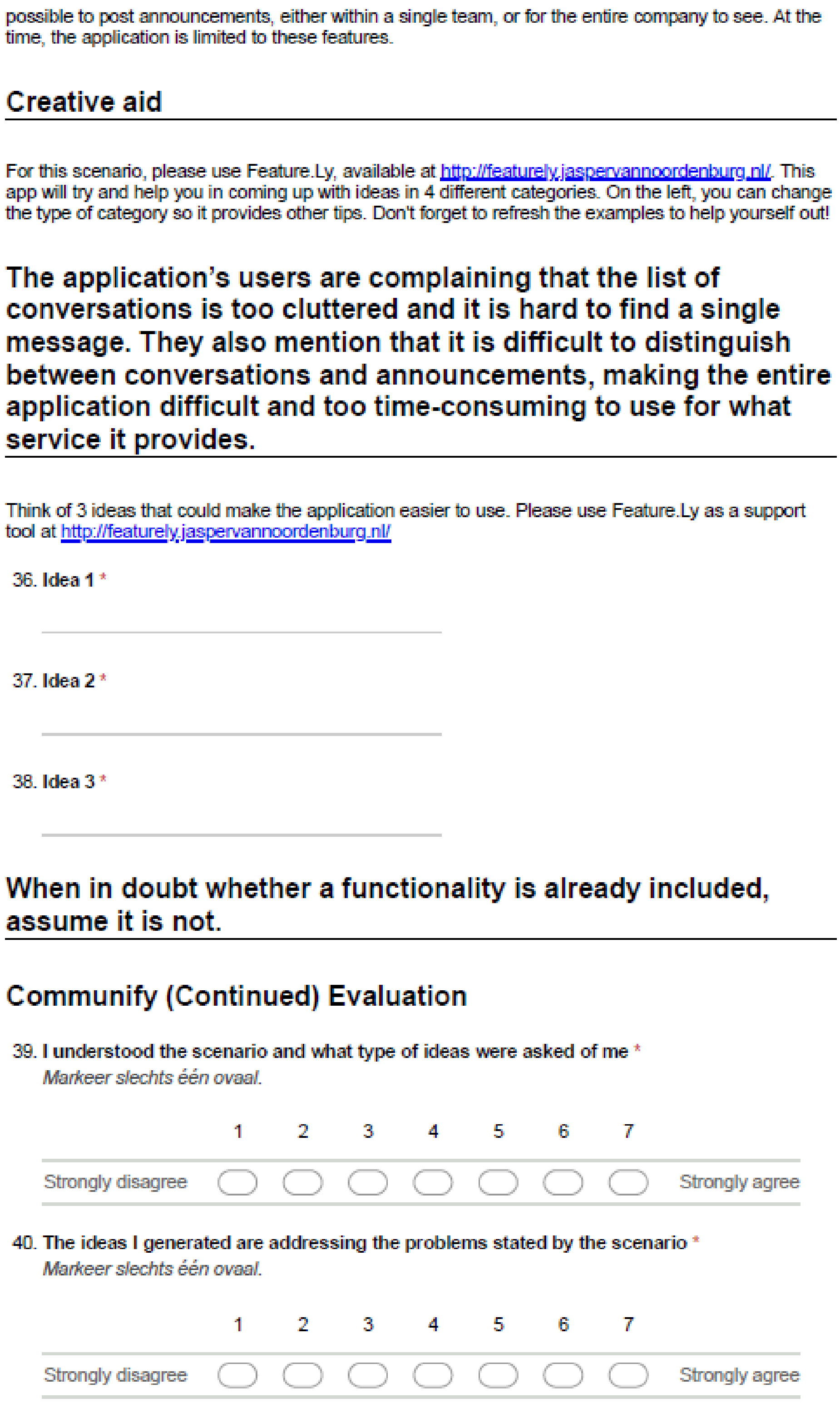
	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

35. I could have provided additional ideas *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

Communify (Continued)

Your organization develops an application, called Communify, in which people can quickly and easily communicate with each other within the company. Current functionality includes the ability to directly message colleagues and create group message conversations for teams to collaborate. It is also



possible to post announcements, either within a single team, or for the entire company to see. At the time, the application is limited to these features.

Creative aid

For this scenario, please use Feature.Ly, available at http://featurely.jaspervannoordenburg.nl/. This app will try and help you in coming up with ideas in 4 different categories. On the left, you can change the type of category so it provides other tips. Don't forget to refresh the examples to help yourself out!

The application's users are complaining that the list of conversations is too cluttered and it is hard to find a single message. They also mention that it is difficult to distinguish between conversations and announcements, making the entire application difficult and too time-consuming to use for what service it provides.

Think of 3 ideas that could make the application easier to use. Please use Feature.Ly as a support tool at http://featurely.jaspervannoordenburg.nl/

36. **Idea 1** *

37. **Idea 2** *

38. **Idea 3** *

When in doubt whether a functionality is already included, assume it is not.

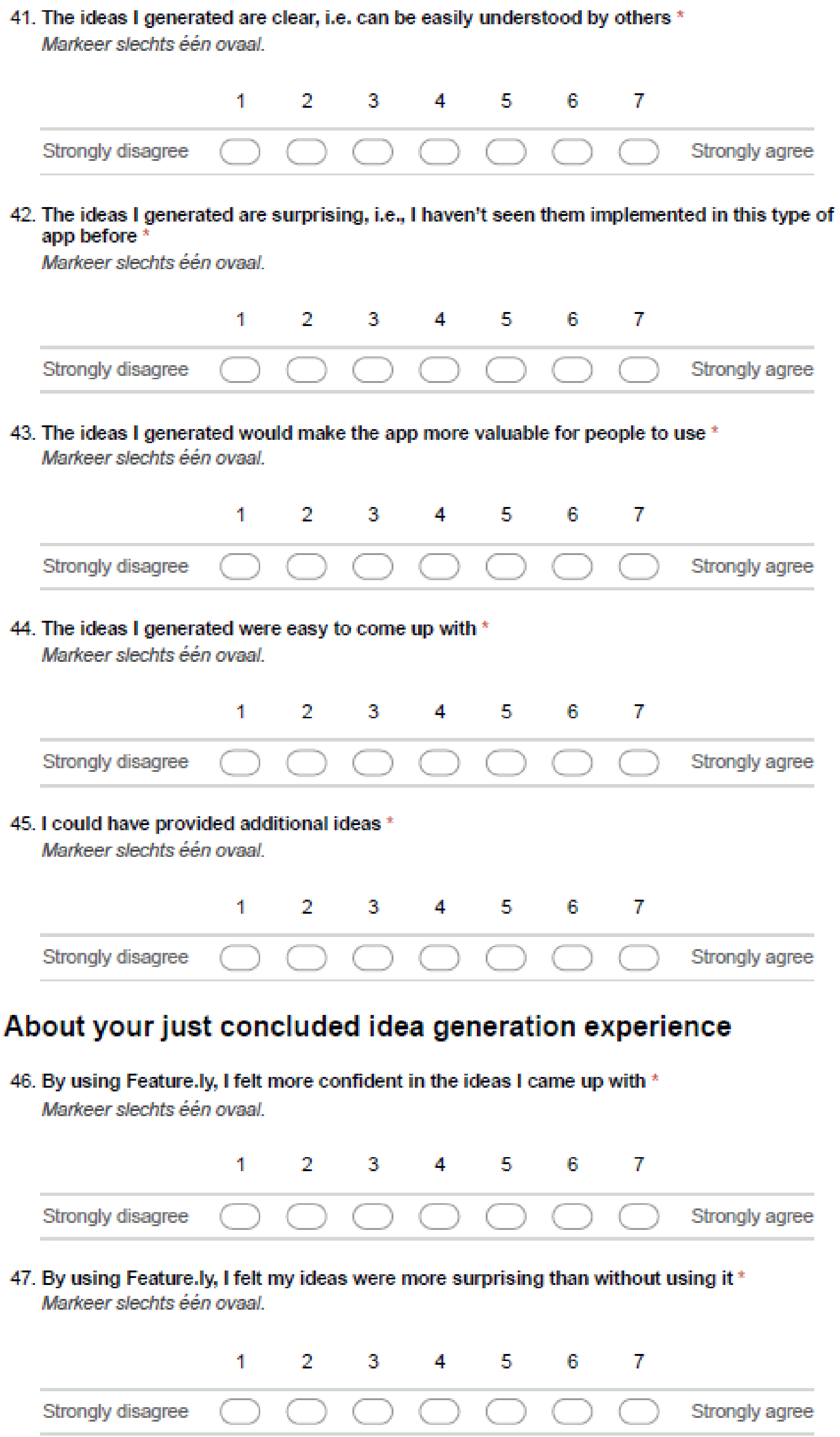
Communify (Continued) Evaluation

39. **I understood the scenario and what type of ideas were asked of me** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

40. **The ideas I generated are addressing the problems stated by the scenario** *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree



41. The ideas I generated are clear, i.e. can be easily understood by others *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

42. The ideas I generated are surprising, i.e., I haven't seen them implemented in this type of app before *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

43. The ideas I generated would make the app more valuable for people to use *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

44. The ideas I generated were easy to come up with *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

45. I could have provided additional ideas *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

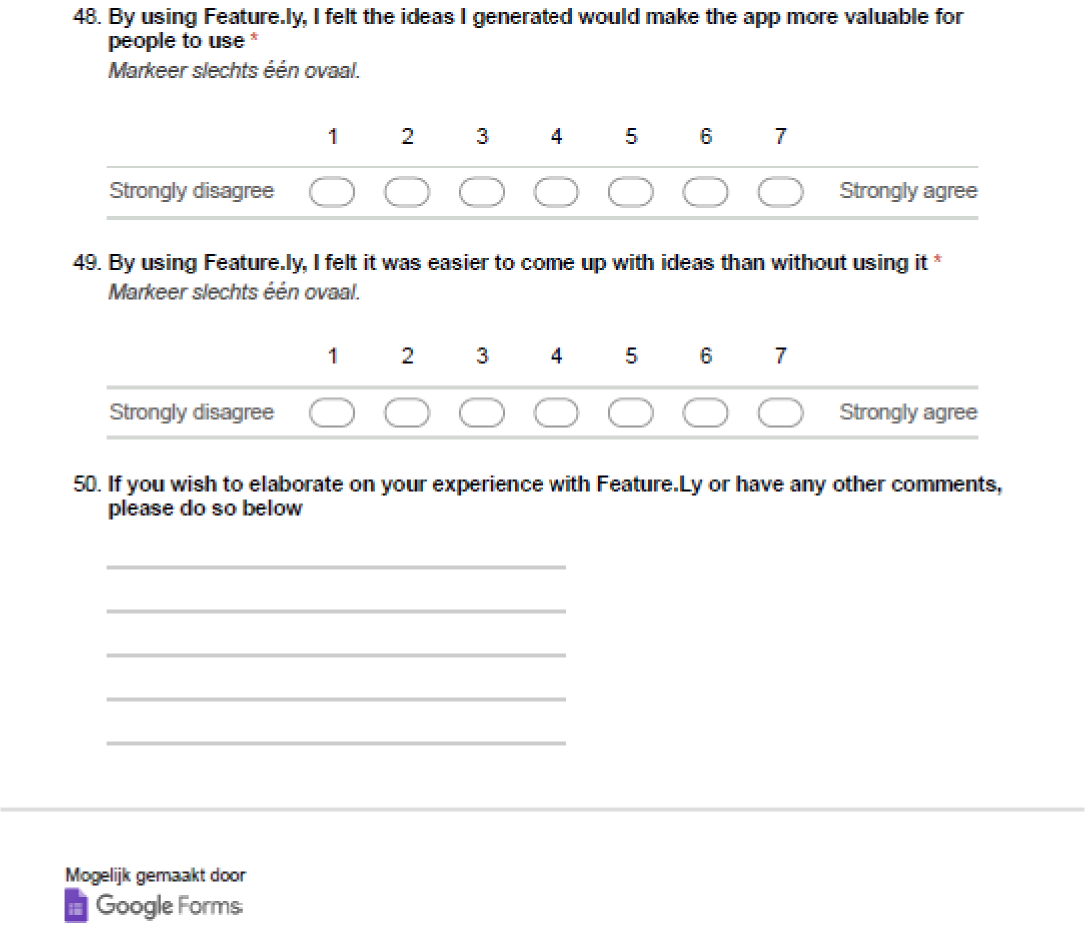
About your just concluded idea generation experience

46. By using Feature.ly, I felt more confident in the ideas I came up with *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

47. By using Feature.ly, I felt my ideas were more surprising than without using it *
Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree



48. By using Feature.ly, I felt the ideas I generated would make the app more valuable for people to use *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

49. By using Feature.ly, I felt it was easier to come up with ideas than without using it *

Markeer slechts één ovaal.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

50. If you wish to elaborate on your experience with Feature.Ly or have any other comments, please do so below

Mogelijk gemaakt door